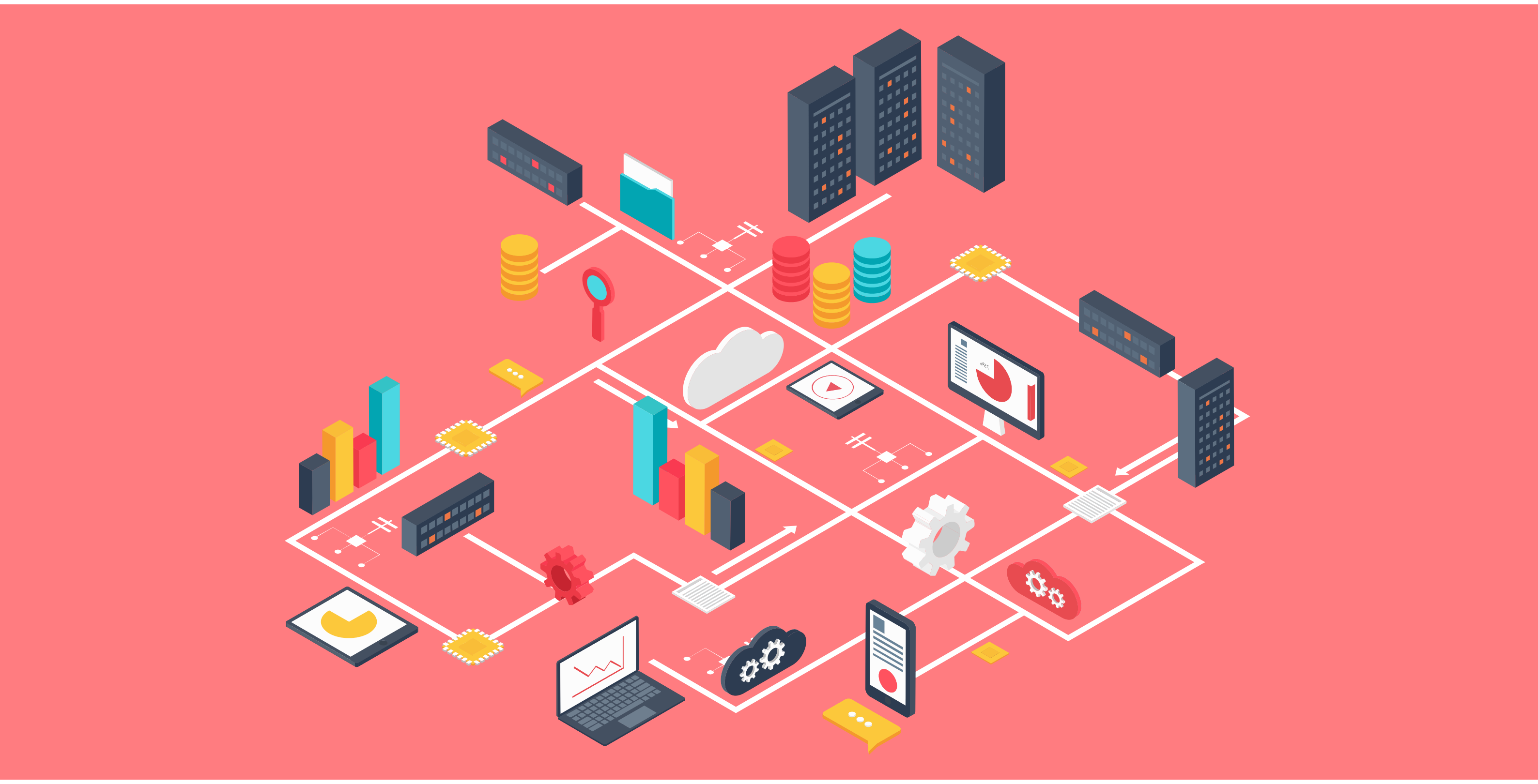
METU EE444 Introduction to Computer Networks

HW2 - Riverbed Modeler

Due: May 20, 2018, 23:55



You are going to submit your homework via **ODTUCLASS** as a **.doc or .docx file, <u>including required screen prints</u>**. Name your document as “HW2_studentid.doc” (ex: “HW2_1234567.doc”).

Please don’t get intimidated by the length of this manual. It is just to make every step clear for you. The procedure won’t take much time. Each part takes about two hours at most, if you pay attention to every step. If your results seem awkward or you can’t see any results at all, just start all over again. It will probably take less time.

Using images or texts from any kind of resource is prohibited. **Believe me, it is not very difficult to find it out.** Cheating will result in zero grade, whereas disciplinary actions may also be taken. Late submissions are not allowed. Our purpose is to introduce you to a tool where you can simulate the network behavior yourself and understand the main concepts. Try to enjoy it.

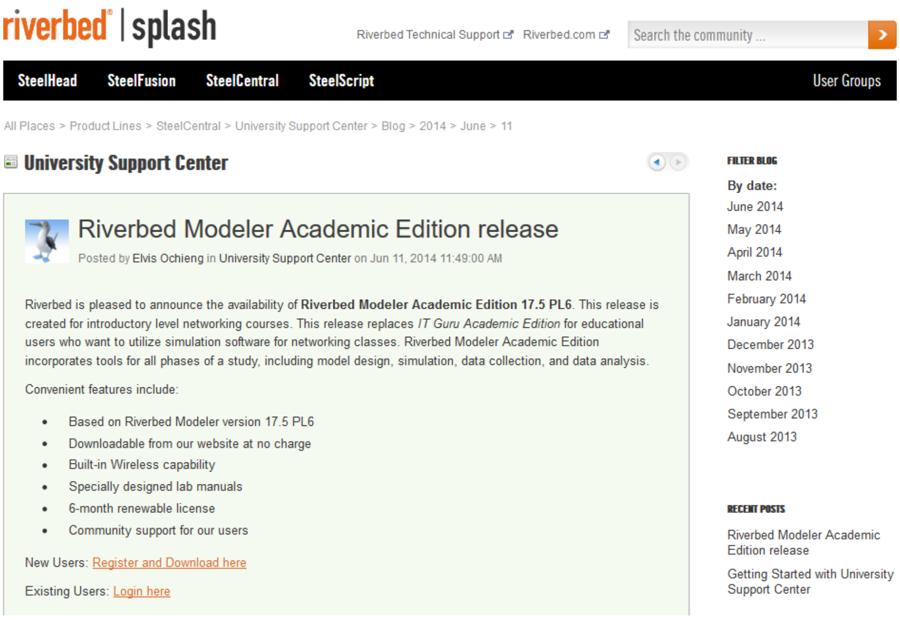
If you have any questions regarding this HW, please send an e-mail to ccakmak@metu.edu.tr

Brief Introduction

Riverbed Modeler comprises of a suite of protocols and technologies with a sophisticated development environment. By modeling all network types and technologies (including VoIP, TCP, OSPFv3, MPLS, IPv6, and more), Riverbed Modeler analyzes networks to compare the impact of different technology designs on end-to-end behavior. Modeler lets you test and demonstrate technology designs before production; increase network R&D productivity; develop proprietary wireless protocols and technologies; and evaluate enhancements to standards-based protocols. In this homework, you will learn the basics of the Riverbed Modeler.

To download the software, please visit the following site to register:

https://splash.riverbed.com/community/product-lines/steelcentral/university-support-center/blog/2014/06/11/riverbed-modeler-academic-edition-release



IMPORTANT!

You may have authorization problems while using this tool. **The solution is to run it as administrator.** If you have previously activated it as a normal user, then you have to activate again, as an administrator. If you can't run as administrator, you may have to wait for a few minutes when you open a new project, or when you open the palette. So don't panic if this happens.

To get the results correctly:

While using the software, change Regional and Language Options of your computer, such that decimal symbol is dot (.). For this purpose you may set your region as English (United States) in your operating system.

Part 1 - TCP: Transmission Control Protocol

A Reliable, Connection-Oriented, Byte-Stream Service

Objective

This part is designed to demonstrate the congestion control algorithms implemented by the Transmission Control Protocol (TCP). It 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 how fast 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 that 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 part you will set up a network that utilizes TCP as its end-to-end transmission protocol and analyze the size of the congestion window with different mechanisms.

Procedure

Create a New Project

1. Start **Riverbed Modeler Academic Edition**
   - ➞ Choose **New** from the **File** menu.
2. Select **Project** and click **OK**
   - ➞ Name the project **<your studentid>_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 Europe from the Map List and add to the list on the right.
   - ➞ Click **Next** twice
   - ➞ Click **Finish. This will create your project and the Object Palette will open, which may take a while.**

Create and Configure the Network

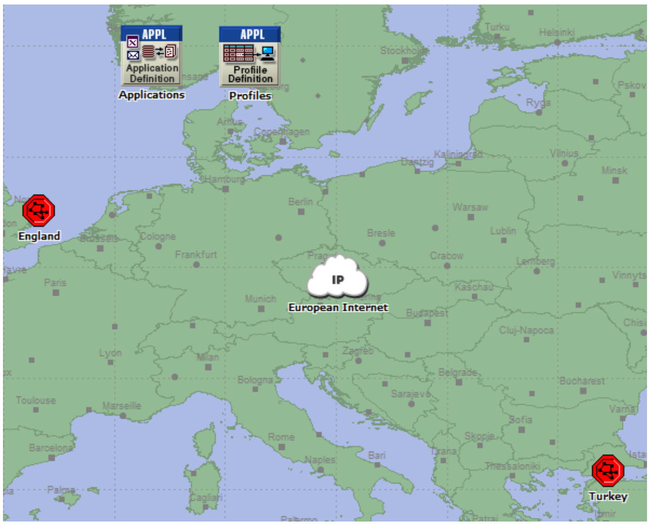
Initialize the Network:

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.
➞ 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.

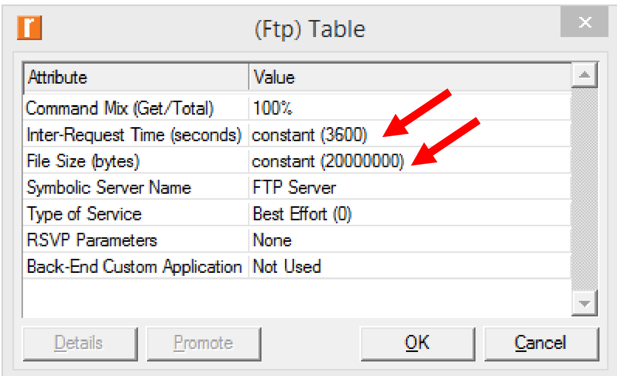
3. Close the palette.
4. Rename the objects you added as shown and then save your project:



*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.*

Configure the Applications

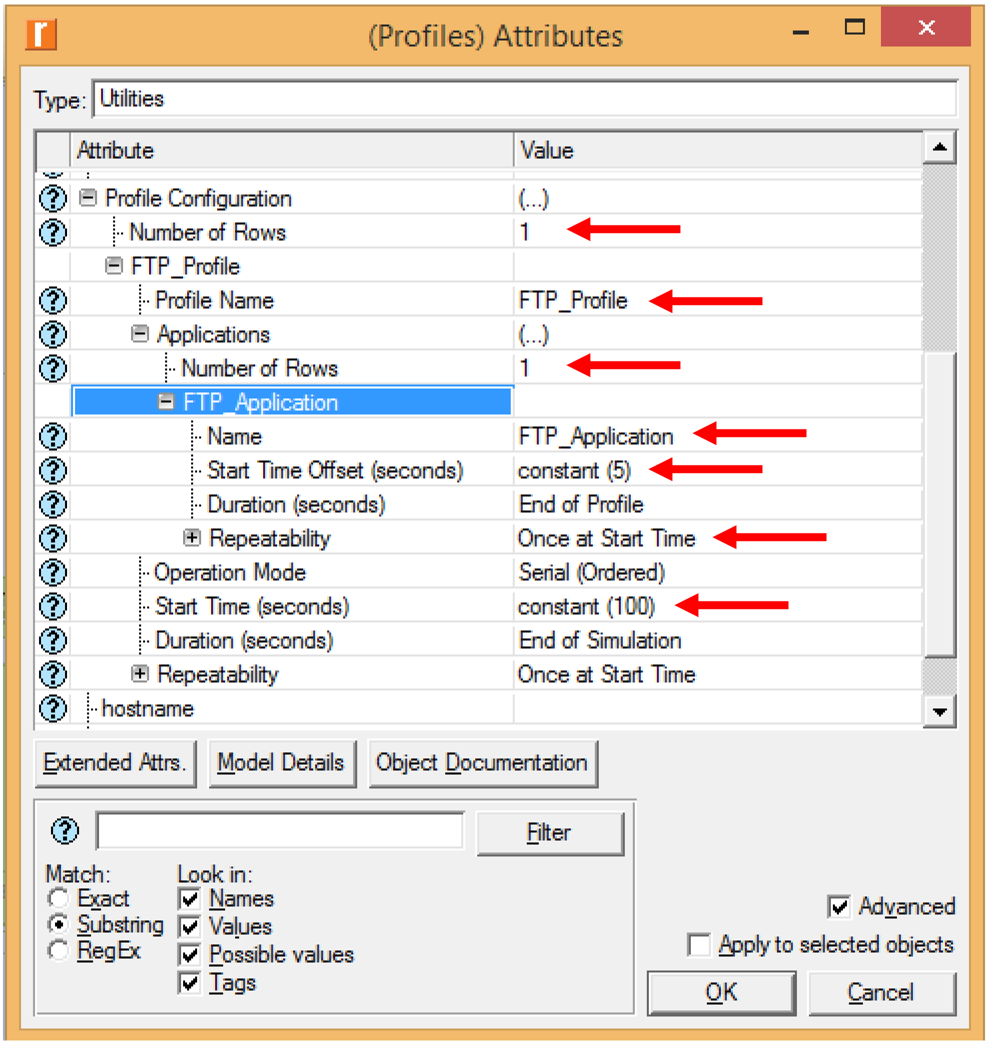
1. Right-click on the Applications node
➞ **Edit Attributes**
➞ Expand the **Application Definitions** attribute and set **Number of rows** to 1
➞ Expand the new row
➞ Name the row **FTP_Application**.
➞ Expand the **Description** hierarchy
➞ Edit the FTP row as shown



2. Click **OK** twice and then save your project.

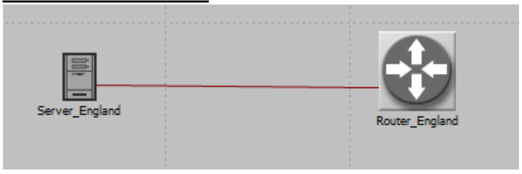
Configure the Profiles

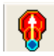
1. Right-click on the **Profiles** node
   ➞ **Edit Attributes**
   ➞ Expand the **Profile Configuration** attribute and set **Number of Rows** to 1.
   ➞ Name and set the attributes as shown below.
   ➞ Click **OK** and save your project**.**



Configure the England Subnet

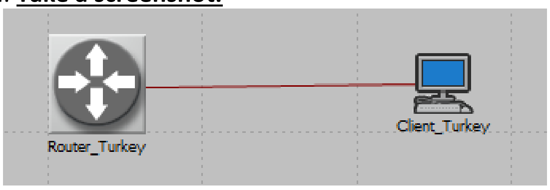
1. Double-click on the **England** subnet node. You get an empty workspace, indicating that the subnet contains no objects.
2. Open the object palette and make sure that the **internet_toolbox** item is selected from the pull-down menu.
3. Add the following items to the subnet workspace: one **ethernet_server**, one **ethernet4_slip8_gtwy** router, and connect them with a bidirectional **100 BaseT** link (Under **Link Models**).
   ➞ Close the palette
   ➞ Rename the objects as shown. **<u>Take a screenshot.</u>**



4. Right-click on the **Server_England** node ➞ Edit Attributes:
   a. Under **Applications**, edit **Application: Supported Services** as follows:
      ➞ Set **Rows** to 1
      ➞ Set **Name** to **FTP_Application**
      ➞ Click **OK.**
   b. Edit the value of the **Server Address** attribute and write down **Server_England**.
   c. Under **TCP**, expand the **TCP Parameters** hierarchy:
      ➞ Set **Flavor** to **Tahoe** and set **Duplicate ACK Threshold** to 100. (We want to see the behavior of the network with/without the retransmission. However, Tahoe flavor has built-in fast retransmission feature. Thus, in the first two scenarios we denote a high value to the number of duplicate ACKs required to do a fast retransmission)
5. Click **OK** and then save your project.
   Now, you have completed the configuration of the **England** subnet. To go back to the top level of the project, click the **Go to Parent Subnet**  button.

Configure the Turkey Subnet

1. Double-click on the **Turkey** 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. **<u>Take a screenshot.</u>**

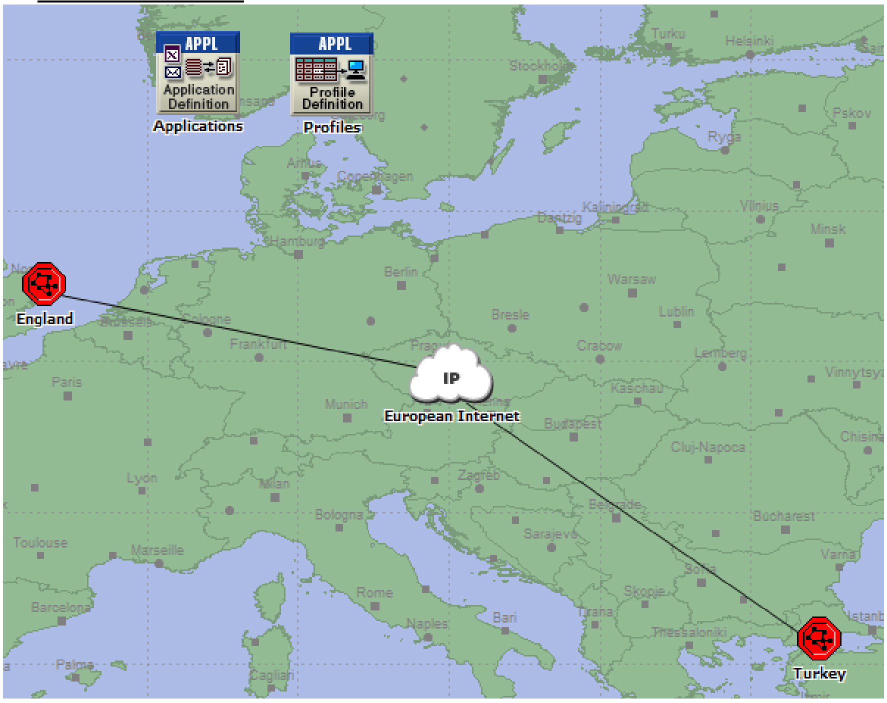


4. Right-click on the **Client_Turkey** node ➞ Edit Attributes:
   a. Under **Applications,** expand the **Application: Supported Profiles** hierarchy
      ➞ Set **rows** to 1
      ➞ Set **Profile Name** to **FTP_Profile**.
   b. Edit the **Application: Destination Preferences** attribute as follows:
      ➞ Set **rows** to 1
      ➞ Under **All Applications,** set **Symbolic Name** to **FTP Server**
      ➞ Edit **Actual Name**
      ➞ Set **rows** to 1
      ➞ In the new row, assign **Server_England** to the **Name** column. (You should be seeing exactly **Server_England** in the drop-down list).
   c. Write **Client_Turkey** to the **Client Address** attributes.

5. Click **OK** and then save your project.
6. You have now completed the configuration of the **Turkey** subnet. To go back to the project space, click the **Go to Parent Subnet** button.

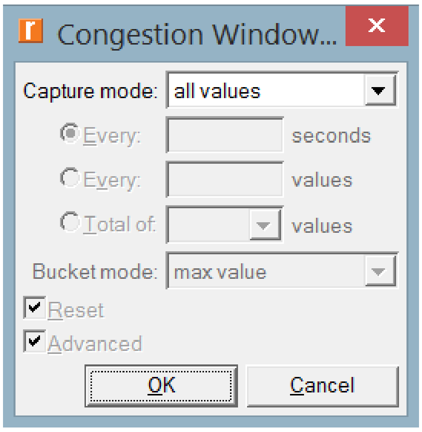
Connect the Subnets to the IP Cloud

1. Open the **Object Palette**
2. Using two **PPP_DS3** bidirectional links connect the **Turkey** subnet to the **IP Cloud** and the **England** subnet to the **IP Cloud**.
3. A pop-up dialog box will appear asking you what to connect the subnet to the **IP Cloud** with. Make sure to select the "routers."
4. Close the palette. **<u>Take a screenshot.</u>**



Choose the Statistics

1. Right-click on **Server_England** in the **England** subnet and select **Choose Individual DES Statistics** from the pop-up menu.
2. In the Choose Results dialog box, check the following statistics:
   Node Statistics ➞ TCP Connection ➞ Congestion Window Size (bytes) and **Sent Segment Sequence Number**.
3. 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**.



4. 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**.
5. Click OK twice and then save your project.
6. Click the **Go to Parent Subnet** 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 be **10.0 minutes**.
3. Click **Run** and then save your project.

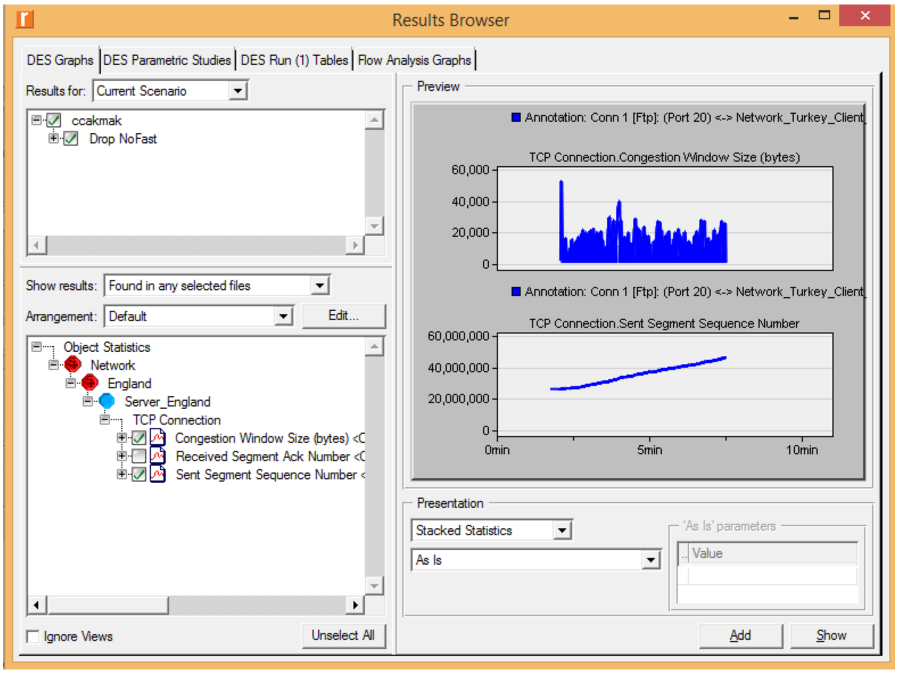
Duplicate the Scenario

In the network we just created we assumed a perfect network with no discarded packets. Also, we disabled the fast retransmission 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**
2. In the new scenario, right-click on the **IP Cloud**
   ➞ **Edit Attributes**
   ➞ Under **Performance Metrics,** assign **1%** to the **Packet Discard Ratio** attribute.
3. Click **OK,** then **run** a simulation and save your project.
4. While you are still in the **Drop_NoFast** scenario, select **Duplicate Scenario** from the **Scenarios** menu and give it the name **Drop_Fast**.
5. In the **Drop_Fast** scenario, right-click on **Server_ England**, which is inside the **England** subnet
   ➞ **Edit Attributes**
   ➞ Under **TCP, e**xpand the **TCP Parameters** hierarchy
   ➞ Set **Flavor** to **Reno** and set **Duplicate ACK Threshold** to 3. (We use the fast retransmission feature in this scenario)
6. Click **OK,** then **run** a simulation and save your project.
   __**Remark:**__ To switch between scenarios 1, 2, 3… you can use the shortcut Ctrl+1, Ctrl+2, Ctrl+3… respectively.

View the Results

1. Switch to the **Drop_NoFast** scenario (the second one) and choose **View Results** from the **Results** in **DES** tab.
2. Fully expand the **Object Statistics** hierarchy and select the following two results: **Congestion Window Size (bytes)** and **Sent Segment Sequence Number**. The results should resemble the ones below. __**Take a screenshot.**__



3. Click **Show**. Zoom in and observe both the slow start and congestion avoidance phases. To zoom in on the details in the graph, click and drag your mouse to draw a rectangle. Notice that the **Segment Sequence Number** is almost flat with every drop in the congestion window. Answer Q1.
4. Close the **View Results** dialog box and select **Compare Results** from the **Results** in **DES** tab.
5. Fully expand the **Object Statistics** hierarchy as shown and select the following result**: Sent Segment Sequence Number**.
6. Click **Show**. Answer Q2.

__**Remark:**__ If you changed anything in scenarios and not sure whether the simulation data is up-to-date or not, you can always check them from **Manage Scenarios** under **Scenarios** tab.

__**Remark:**__ In your report, first put the screenshots related to the procedure. While answering the questions, take screenshots of the related "graphs" (Where you click **Show** and make it full screen) and include them in your answers.

Questions - Part 1:

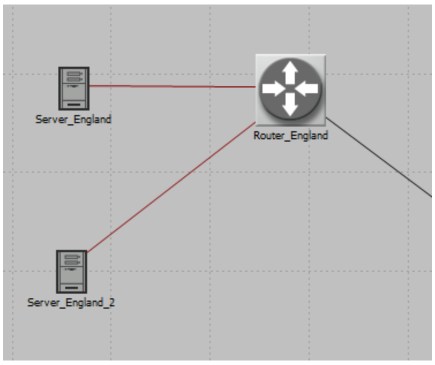
Remark: Write brief, clear, to-the-point answers to the questions!

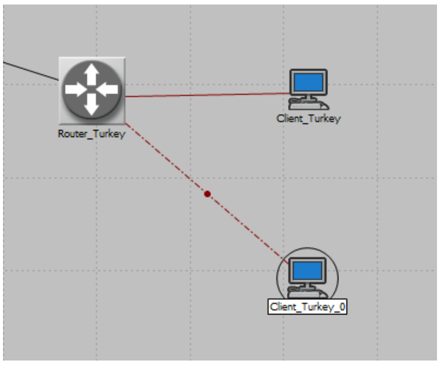
1. Why does the **Segment Sequence Number** remain unchanged (indicated by a horizontal line in the graphs) with every drop in the congestion window?

2. Analyze the graph that compares the **Segment Sequence** numbers of the three scenarios. Why does the **Drop_NoFast** scenario have the slowest growth in sequence numbers?

3. In the **Drop_NoFast** scenario, obtain and zoom in the overlaid graph that compares **Sent Segment Sequence Number** with **Received Segment ACK Number** for **Server_England**. Does the SSSN always increase? Why?
Hint: - Make sure to assign **all values** to the capture mode of the **Received Segment ACK Number** statistics.

4. Switch the scenario to **Drop_Fast**. Select **Duplicate Scenario** from the **Scenarios** menu and give it the name **Q4_Drop_Fast_Multiple** ➞ Click **OK.** Now, switch to the England subnet. Copy the server in England subnet (Ctrl+C) and paste it to its near (Ctrl+V). After connecting the new client to the router with 100 BaseT link, change its name and **address** to "Server_England_2". Now, switch to Turkey subnet. Copy the client in Turkey subnet (Ctrl+C) and paste it to its near (Ctrl+V). After connecting the new client to the router with 100BaseT link , change its name and **address** to "Client_Turkey_2". Change the **Destination Preferences** of the new client and choose destination address as Server_England_2. The subnets should look like the following.





Now, we created a second FTP connection between the newly created client and server. In this part, you will observe the behavior of multiple TCP connections. Now, run the simulation, compare the **Sent Segment Sequence Number** of two connections and comment on the **fairness** of TCP.

Part 2 - Ethernet

A Direct Link Network with Media Access Control

Objective

This part is designed to demonstrate the operation of the Ethernet network. The simulation in this part will help you examine the performance of the Ethernet network under different scenarios.

Overview

The Ethernet is a working example of the more general Carrier Sense, Multiple Access with Collision Detect (CSMA/CD) local area network technology. The Ethernet is a multiple-access network, meaning that a set of nodes sends and receives frames over a shared link. The "carrier sense" in CSMA/CD means that all the nodes can distinguish between an idle and a busy link. The "collision detect" means that a node listens as it transmits and can therefore detect when a frame it is transmitting has interfered (collided) with a frame transmitted by another node. The Ethernet is said to be a 1-persistent protocol because an adaptor with a frame to send transmits with probability 1 whenever a busy line goes idle.

In this part you will set up an Ethernet with thirty nodes connected via a coaxial link in a bus topology. The coaxial link is operating at a data rate of 10 Mbps. You will study how the throughput of the network is affected by the network load as well as the size of the packets.

Procedure

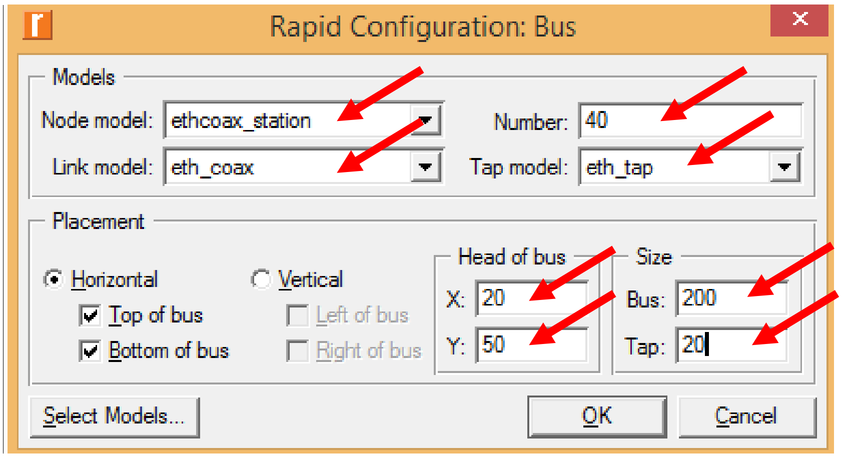
Create a New Project

1. Start **Riverbed Modeler Academic Edition**
   ➞ Choose **New** from the **File** menu.
2. Select **Project** and click **OK**
   ➞ Name the project **<your studentid>_ Ethernet**, and the scenario **Coax_2.** Make sure that the **Use Startup Wizard** is checked.
   ➞ Click **OK**.
3. In the *Startup Wizard: Initial Topology* dialog box, make sure that **Create Empty Scenario** is selected
   ➞ Click **Next**
   ➞ Choose **Office** from the *Network Scale* list
   ➞ Click **Next**
   ➞ Assign **300** to **X Span** and keep **Y Span** as **100**
   ➞ Click **Next** twice
   ➞ Click **Finish.**
4. **Close** the *Object Palette* dialog box.

Create the Network

To create our coaxial Ethernet network:

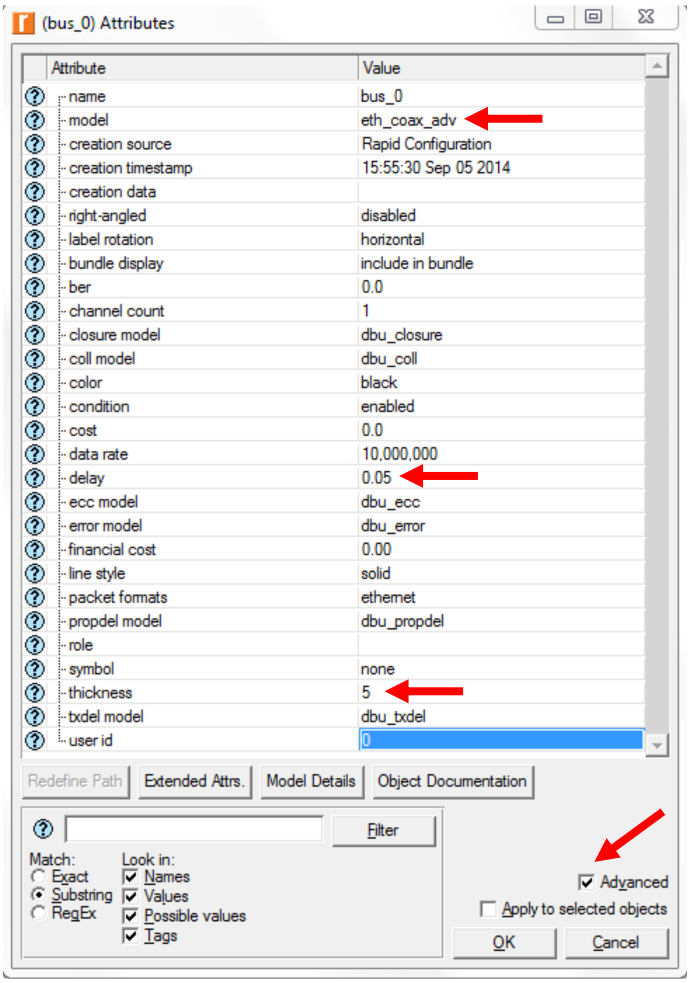
1. To create the network configuration, select **Topology**
   ➞ **Rapid Configuration.** From the drop-down menu choose **Bus** and click **Next.**
2. Click the **Select Models** button in the *Rapid Configuration* dialog box. From the *Model List* drop-down menu choose **eth_coax** and click **OK.**
3. In the *Rapid Configuration* dialog box, set the following eight values and click **OK.**



The ***eth_tap*** *is an Ethernet bus tap that connects a node with the bus.*

The ***eth_coax*** *is an Ethernet bus that can connect nodes with bus receivers and transmitters via taps. If* ***ethcoax_station*** *does not exist in Node Model category, you can choose it from* ***Select Models*** *button.*

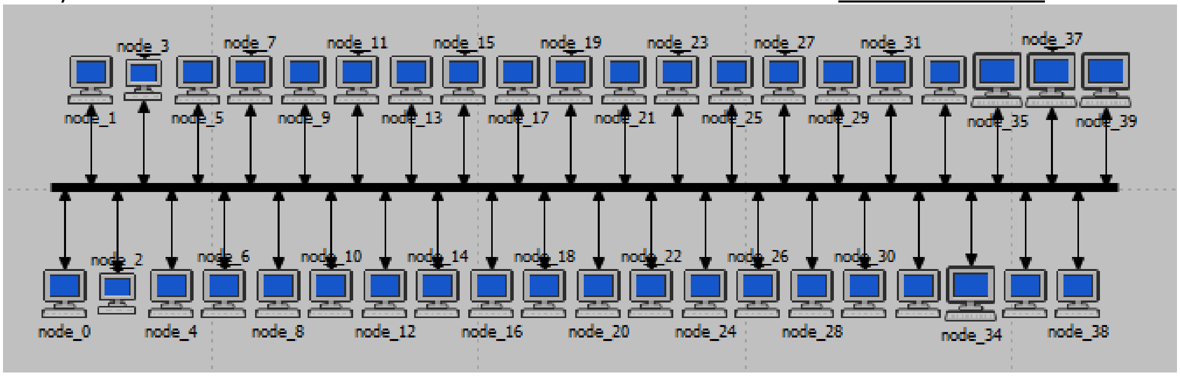
4. To configure the coaxial bus, right-click on the horizontal link
   ➞ Select **Edit Attributes (Advanced)** from the menu:
   a. Click on the value of the **model** attribute
      ➞ Select **Edit** from the dropdown menu
      ➞ Choose the **eth_coax_adv** model.
   b. Assign the value **0.05** to the **delay** attribute (propagation delay in sec/m).
   c. Assign **5** to the **thickness** attribute.
   d. Click **OK.**



A higher delay *is used here as an alternative to generating higher traffic which would require much longer simulation.*

Thickness *specifies the thickness of the line used to "draw" the bus link.*

5. Now you have created the network. It should look like the illustration below. **Take a screenshot.**

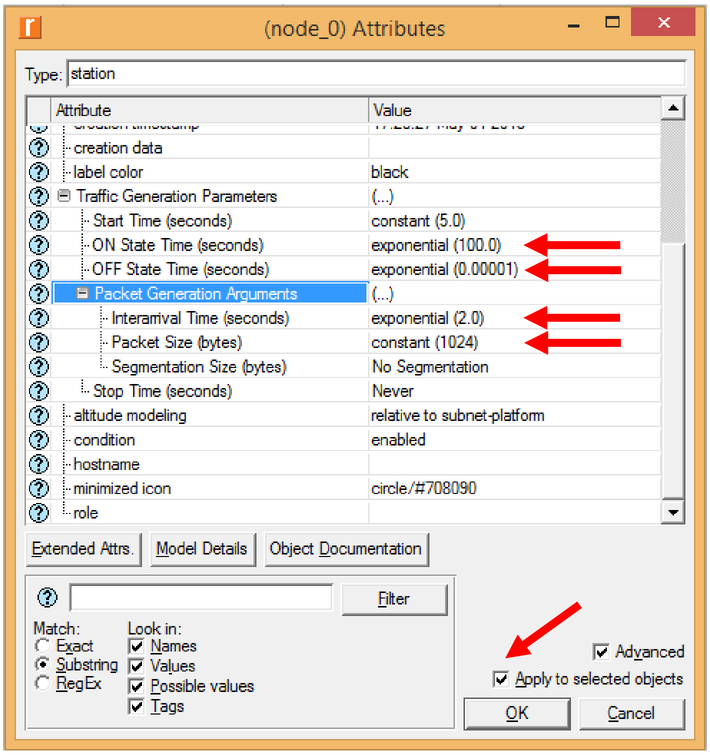


6. Make sure to save your project.

Configure the Network Nodes

To configure the traffic generated by the nodes:

1. Right-click on any of the 40 nodes
   ➞ **Select Similar Nodes.** Now all nodes in the network are selected.
2. Right-click on any of the 40 nodes
   ➞ **Edit Attributes.**
3. Check the **Apply to Selected Objects** check box. This is important to avoid reconfiguring each node individually.
4. Expand the **Traffic Generation Parameters** hierarchy:
   Change the value of the **ON State Time** to **exponential(100)**
   ➞ Change the value of the **OFF State Time** to **exponential(0.00001).** *(Note:* Packets are generated only in the "ON" state.)
5. Expand the **Packet Generation Arguments** hierarchy:
   ➞Change the value of the **Packet Size** attribute to **constant(1024).**
   ➞Change the value of the **Interarrival Time** attribute to **exponential(2)**.

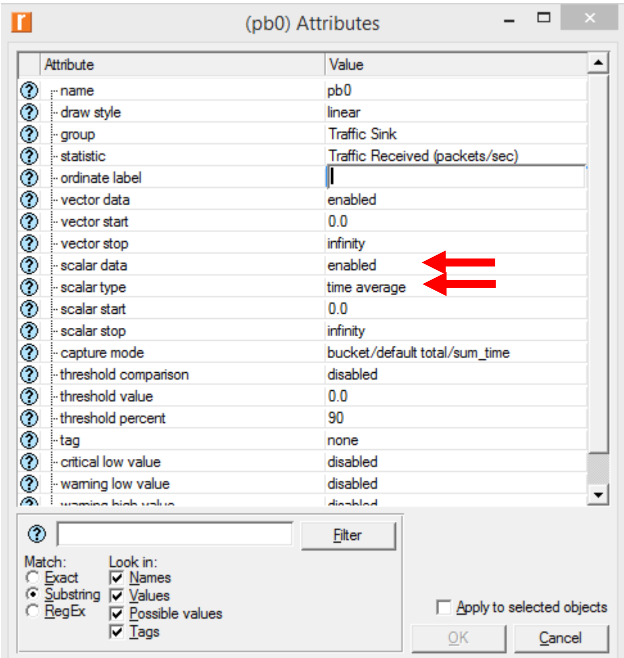


6. Click **OK** to return back to the *Project Editor.*
7. Make sure to save your project.

Choose the Statistics

To choose the statistics to be collected during the simulation:

1. Right-click anywhere in the project workspace (but not on one of the nodes or links) and select **Choose Individual DES Statistics** from the pop-up menu ➞ Expand the **Global Statistics** hierarchy.
   a. Expand the **Traffic Sink** hierarchy ➞ Click the check box next to **Traffic Received (packets/sec)**
   b. Expand the **Traffic Source** hierarchy ➞ Click the check box next to **Traffic Sent (packets/sec).**
   c. Click **OK.**
2. Now to collect the average of the above statistics as a scalar value by the end of each simulation run:
   a. Select **Choose Statistics (Advanced)** from the **DES** tab.
   b. The **Traffic Sent** and **Traffic Received** probes should appear under the **Global Statistic Probes.** (A **probe** represents a request by the user to collect a particular piece of data about a simulation.)
   c. Right-click on **Traffic Received** probe
      ➞ **Edit Attributes.** Set the **scalar data** attribute to **enabled**
      ➞ Set the **scalar type** attribute to **time average**
      ➞ Compare to the following figure and click **OK.**

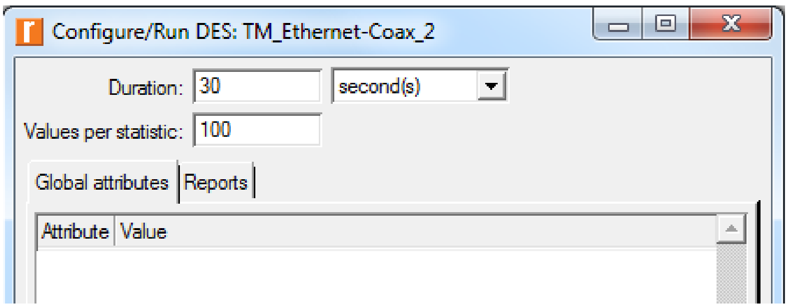


   d. Repeat the previous step with the **Traffic Sent** probe.
   e. Select save from the **File** menu in the *Probe Model* window and then close that window.
   f. Now you are back to the *Project Editor.* Make sure to save your project.

Run the Simulation

To run the simulation:

1. Click on the **Configure/Run Simulation** button.
   ➞ Make sure that **30 second(s)** is assigned to the **Duration**
   ➞ Click **Run.** Depending on the speed of your processor, this may take several minutes to complete.

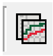


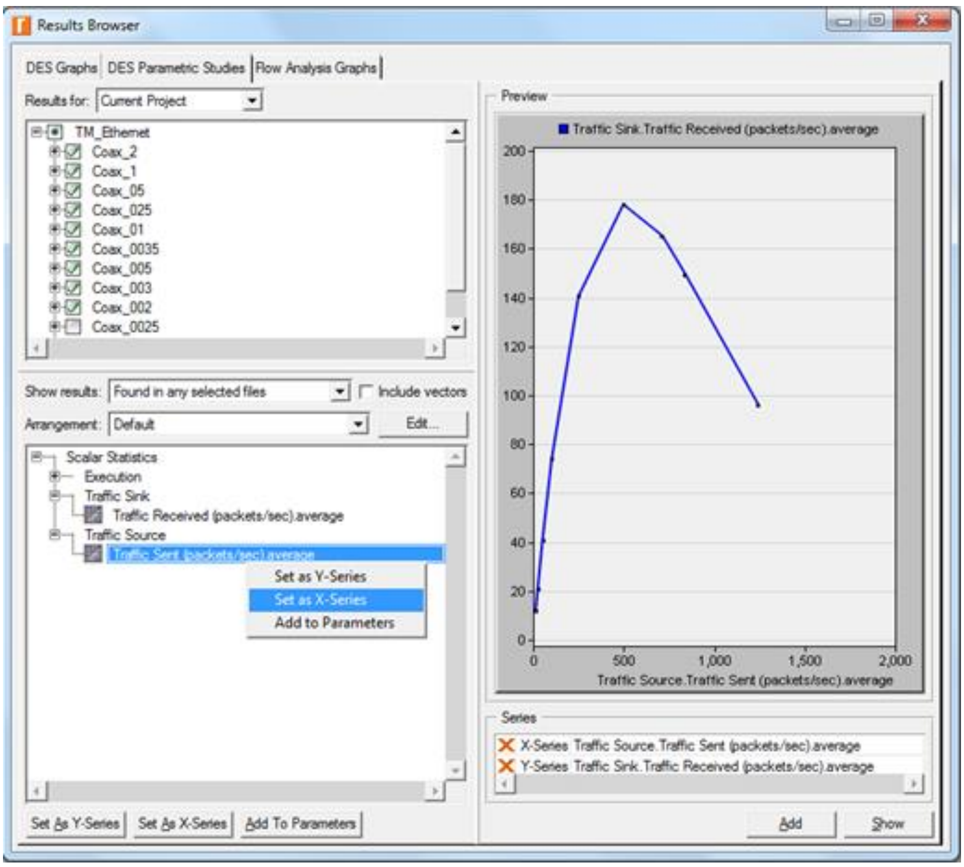
2. After the simulation run is complete, click **Close.**
3. Save your project.
4. Select **Duplicate Scenario** from the *Scenarios* menu.
5. Name the new scenario **Coax_1**.
6. Right-click on any of the 40 nodes ⇒ **Select Similar Nodes**. Now all nodes in the network are selected.
7. Right-click on any of the 40 nodes ⇒ **Edit Attributes**.
8. Check the **Apply to Selected Objects** check box.
9. Expand the **Traffic Generation Parameters** hierarchy ⇒ Expand the **Packet Generation Arguments** hierarchy ⇒ Change the value of the **Interarrival Time** attribute to **exponential(1)**.
10. Go to back to step 1 in this page and repeat the instructions 8 times for the **Interarrival Times** of **exponential(0.5)**, **exponential(0.25)**, **exponential(0.1)**, **exponential(0.05)**, **exponential(0.035)**, **exponential(0.03)**, **exponential(0.025)** and **exponential(0.02)** using the scenarios names: **Coax_05**, **Coax_025**, **Coax_01**, **Coax_005**, **Coax_0035**, **Coax_003**, **Coax_0025** and **Coax_002**. Notice that each time the simulator is completing a run for different traffic generation interarrival time (representing the load into the network) and that each successive run takes longer to complete because the traffic intensity is increasing.

    REMEMBER TO CHECK **APPLY TO SELECTED OBJECTS** EACH TİME!!!

View the Results

To view and analyze the results:

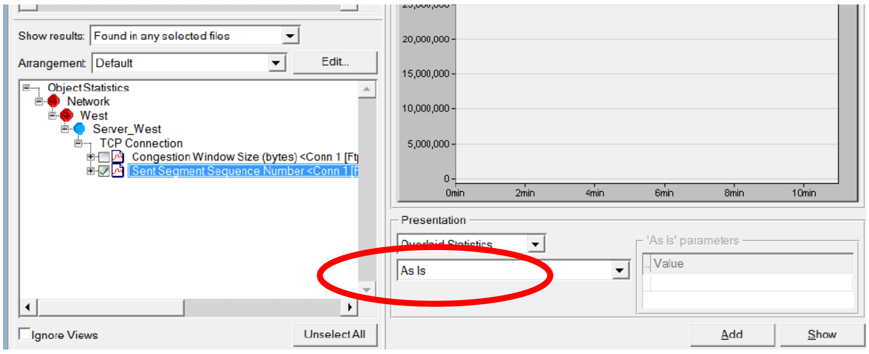
1. Click **View Results** button  ; now the **Results Browser** is open.
2. Select **DES Parametric Studies** tab.
3. From the **Results for** drop-down list, select **Current Project**
4. Check all scenarios.
5. Uncheck **Include Vectors** option.
6. Expand the **Scalar Statistics**
7. Expand the **Traffic Sink** and **Traffic Source**.
8. Right click on **Traffic Received** and select **Set as Y-Series**
9. Right click on **Traffic Sent** and select **Set as X-Series**
10. The resulting window should look like the one below. **<u>Take a screenshot.</u>**



Questions – Part 2

1. Show the resulting graph. Why does the throughput drop when the load is either very low or very high?

Remark: For the following questions, change the **Presentation** mode of the graphs from **As Is** to **average** to obtain average values of the collision counts and the amount of received traffic.



2. Create three duplicates from the simulation scenarios implemented in this lab named **Coax_01**, **Coax_005**, and **Coax_0025**. Make sure that the **Interarrival Time** attribute of the *Packet Generation Arguments* for all nodes in the new scenarios are as follows:

    a. **Coax_Q2_01** scenario: exponential(0.1)
    b. **Coax_Q2_005** scenario: exponential(0.05)
    c. **Coax_Q2_0025** scenario: exponential(0.025)

    For all three scenarios, choose the following statistic for node 0: **Node Statistics** →**Ethcoax** →**Collision Count**. Make sure that the following global statistic is chosen: **Global Statistics**→**Traffic Sink**→**Traffic Received (packet/sec)**. (Refer to the *Choose the Statistics* section in the lab.)

    Run the simulation for all three scenarios. Get two graphs: one to compare node 0's collision counts in these three scenarios and another graph to compare the received traffic from the three scenarios. Which one has the best throughput? Why? (*Note:* To compare results you need to select **Compare Results** from **Results** in the **DES** menu after the simulation runs is done.)

3. Create a duplicate of the **Coax_0025** scenario. Name the new scenario **Coax_Q3.** In the new scenario, remove the odd-numbered nodes, a total of 20 nodes (node 1, node 3, …, and node 39). Run the simulation for the new scenario. Create a graph that compares node 0's collision counts in scenarios **Coax_0025** and **Coax_Q3.** How does the number of collisions change?

4. In the simulation a packet size of 1024 bytes is used *(Note:* Each Ethernet packet can contain up to 1500 bytes of data). Create a duplicate of the **Coax_0025** scenario. Name the new scenario **Coax_Q4.** In the new scenario use a packet size of 512 bytes (for all nodes). Rerun the simulation for **Coax_0025** and **Coax_Q4** scenarios. Create a graph that compares the throughput as packets/sec in **Coax_0025** and **Coax_Q4** scenarios. Which one performs better?