

Lab 7 TCP Throughput

Overview

TCP uses a sliding window mechanism to provide flow control. The destination advertises how much space it has available in its buffers, and the source restricts its transmissions so that the receiver is not overloaded. A small flow control window, however, can impact the throughput of the TCP connection. In the extreme, a small flow control window can cause TCP to act like a stop-and-wait protocol. A very large flow control window would allow the TCP source to transmit continuously.

Objective

To examine the throughput of a TCP connection as the flow control window size is varied.

Build the Simulation Model

Start up OPNET IT Guru Academic Edition.

Select the **File** tab => **New...**

Choose **Project** and click on **OK**.

Change the **Project Name** to **xx_TCP_Window_Size** (where **xx** are your initials) and click on **OK**.

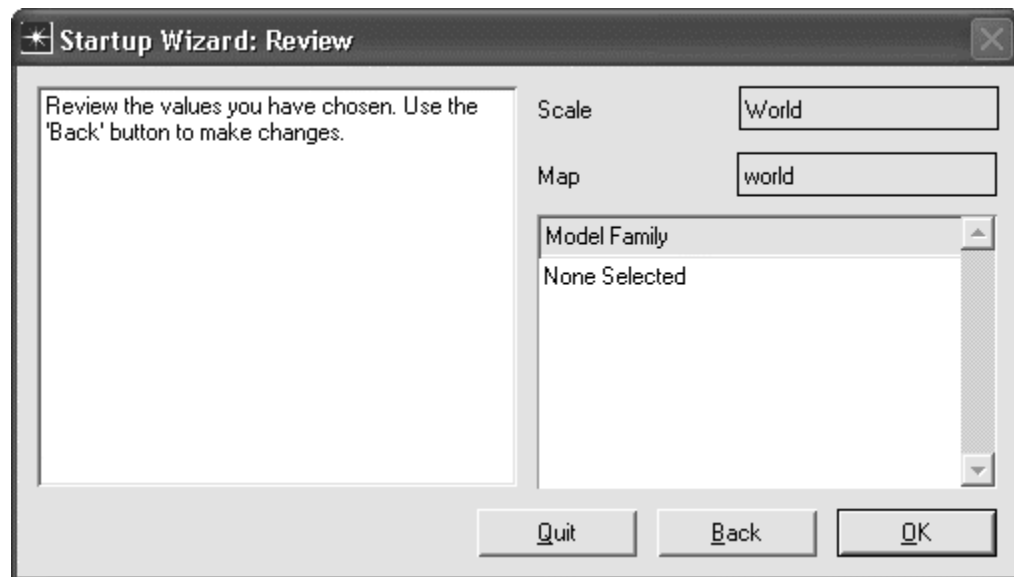
In the **Initial Topology** window, select **Create Empty Scenario** and click on **Next**.

In the **Choose Network Scale** window, select **World** and click on **Next**.

In the **Choose Map** window, choose **world** and click on **Next**.

In the **Select Technologies** window, click on **Next**.

In the **Review** window, click on **OK**.

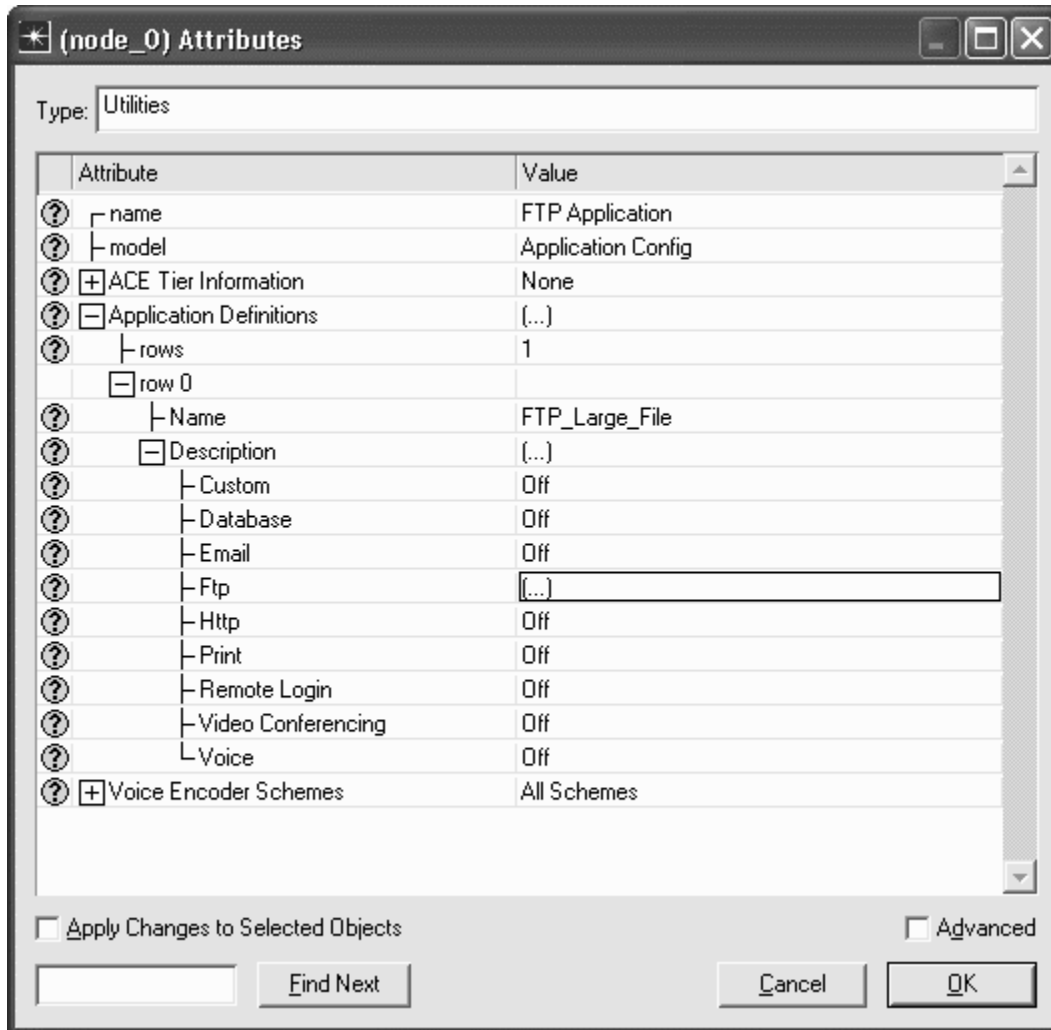


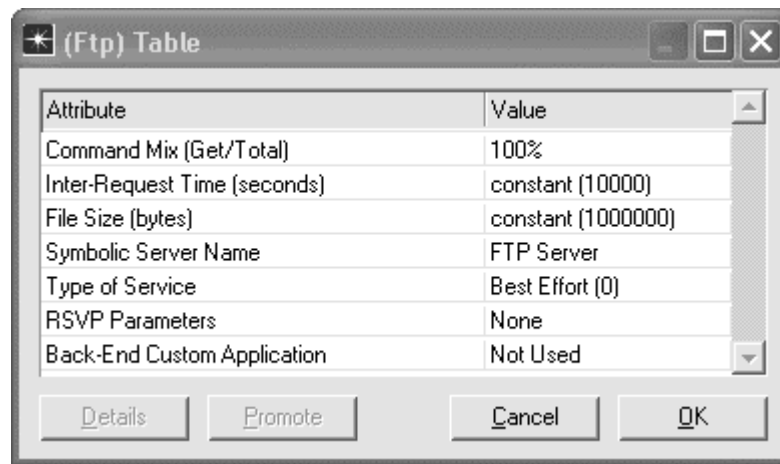
First, it would be helpful to familiarize yourself with the TCP functionality of OPNET.

Select the **Protocols** tab => **TCP** => **Model Usage Guide**. Read through the guide and click on the close window icon to close the PDF viewer when you are done.

Next, we need to configure an application which uses TCP, and associate the application with a profile that will drive our network devices. The File Transfer Protocol (FTP) is built on top of TCP and will serve our purposes.

Select an **Application Config** object from the Object Palette and place it in the project workspace. Right click on the object and choose **Edit Attributes**. Set the **name** attribute to **FTP Application**. Expand the **Application Definitions** attribute and set the **rows** attribute to **1**. Expand the **row 0** attribute and set the **Name** attribute to **FTP_Large_File**. Expand the **Description** attribute and edit the value of the **Ftp** attribute. Set the **Inter-Request Time (secs)** attribute to **constant(10000)**. Set the **File Size (bytes)** attribute to **constant(1000000)**. In both cases, you will need to set the **Special Value** field to **Not Used** in order to modify the attribute values. The application you have now defined will transfer one 1 MB file at a time so that we can easily analyze the TCP behavior. Click on **OK** to close the window.

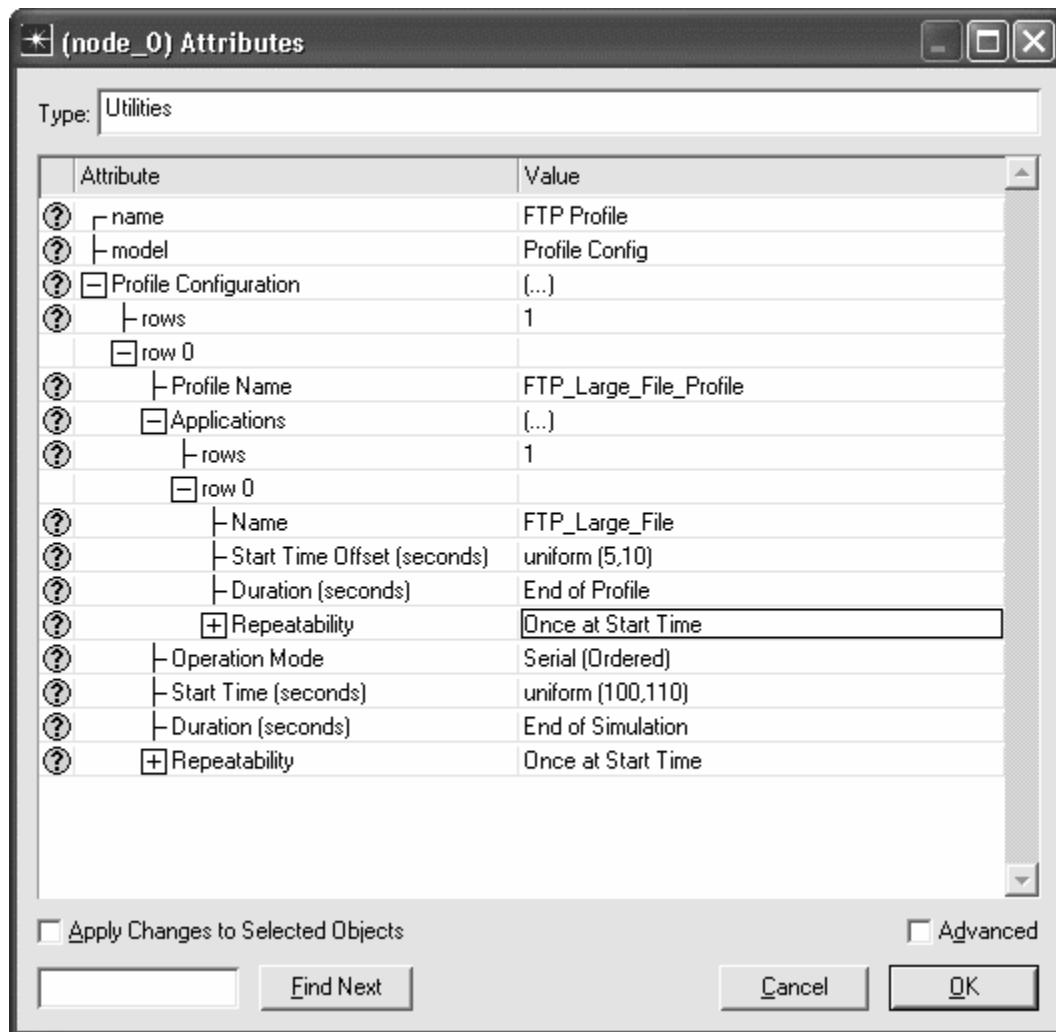




Select a **Profile Config** object from the Object Palette and place it in the project workspace. Right click on the object and choose **Edit Attributes**. Set the **name** attribute to **FTP Profile**.

Expand the **Profile Configuration** attribute and set the **rows** attribute to **1**. Expand the **row 0** attribute and set the **Profile Name** to **FTP_Large_File_Profile**. Expand the **Applications** attribute and set the **rows** attribute to **1**. Expand the **row 0** attribute and set the **Name** to **FTP_Large_File**. Set the **Repeatability** attribute to **Once at Start Time**.

Click on **OK** to close the window.



Select an **ip32_cloud** object from the Object Palette and place it in the project workspace. Right click on the cloud and choose **View Node Description**. The cloud represents a WAN consisting of IP-capable routers that supports up to 32 serial links

Right click on the cloud and select **Set Name**. Set the name to **ip32_cloud**. Click on **OK** to close the window.

Right click on the cloud and select **Edit Attributes**. Set the **Packet Latency (secs)** to **constant(0.25)**. This means that the round-trip delay will be at least 0.5 second. You will need to set the **Special Value** field to **Not Used** in order to modify the latency. Note that the **Packet Discard Ratio** is set to **0.0%**. This means that no losses will occur.

Click on **OK** to close the window.

Select a **ppp_wkstn** device from the Object Palette and place it in the project workspace.

Right click on the station and choose **View Node Description**. Note that the station supports the TCP protocol. We will now set up the client to use the FTP application that we defined.

Right click on the station and choose **Edit Attributes**. Modify the **name** attribute of the device to **FTP Client**.

Edit the **Application: Supported Profiles** attribute. Insert a row and set the **Profile Name** to **FTP_Large_File_Profile**.

Expand the **TCP Parameters** attribute and note that the **Receive Buffer (bytes)** field is set to **8760**. This means that the TCP source may not send more than 8760 bytes of data without receiving an acknowledgment.

Click on **OK** to close the window.

Select a **ppp_server** device from the Object Palette and place in the project workspace.

Right click on the server and choose **View Node Description**. Note that the server supports the TCP protocol. We will now set up the server to use the FTP application that we defined.

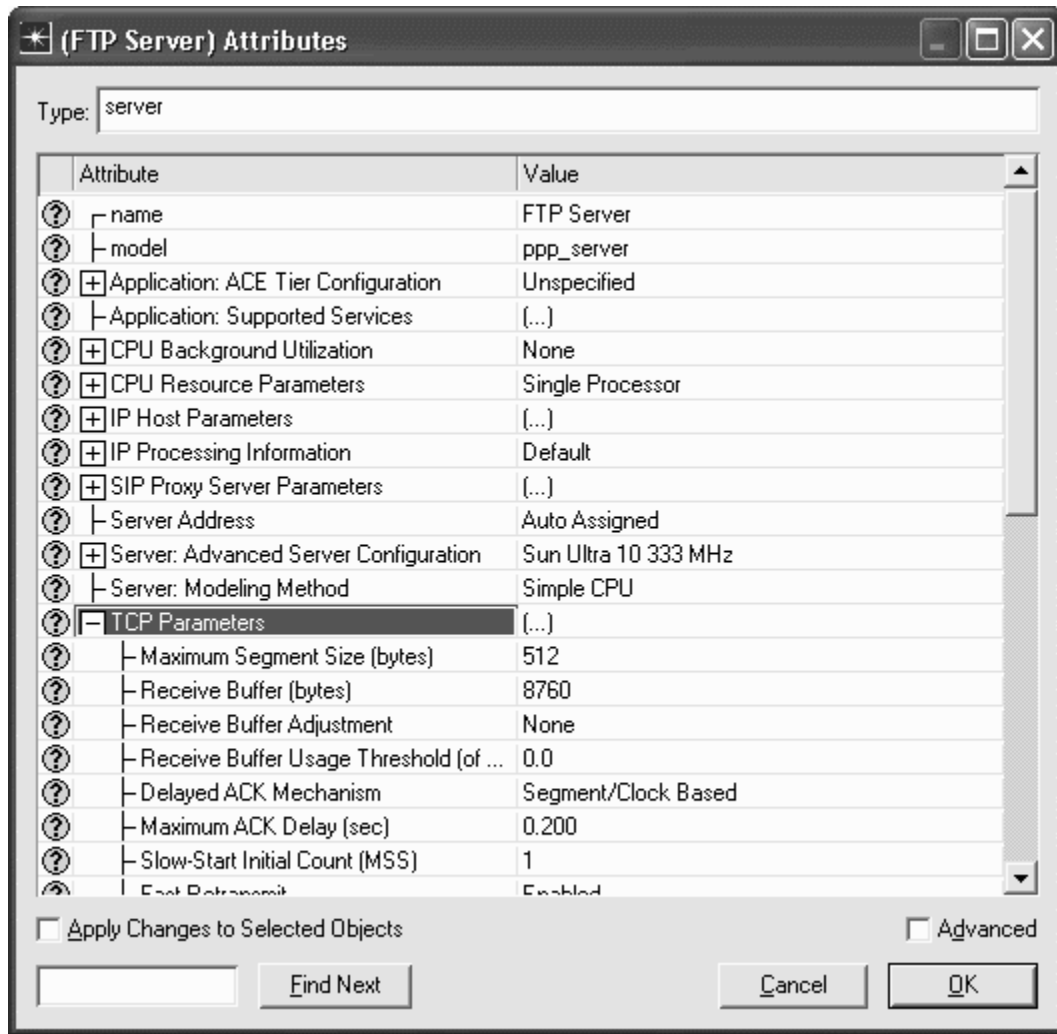
Right click on the device and choose **Edit Attributes**. Modify the **name** attribute of the server to **FTP Server**.

Edit the **Application: Supported Services** attribute. Set the **rows** field to **1**. In the first row, set the **Name** to **FTP_Large_File**. Click on **OK** to close the table window.

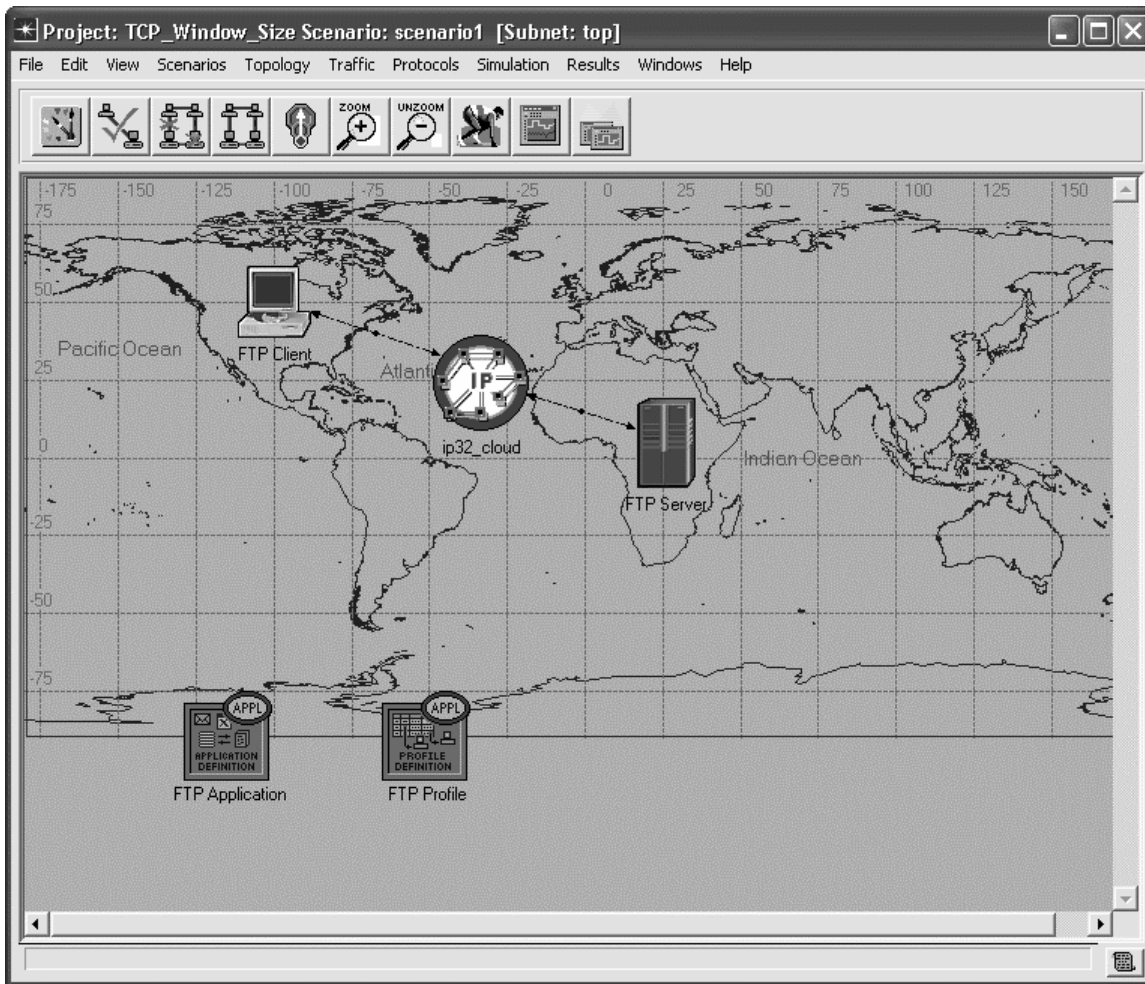
Expand the **TCP Parameters** attribute and set the **Maximum Segment Size (bytes)** attribute to **512**. This will ensure that each TCP packet sent is 512 bytes long.

Note that the receive buffer size on the server will not affect the TCP throughput as data flow is only in one direction (from server to client).

Click on **OK** to close the window.



Select a **PPP_DS1** link from the Object Palette and use it to connect the FTP client to the ip32_cloud and the FTP server to the cloud. The PPP protocol is commonly used for long-distance links. Also, remember that DS1 speed is 1.5 Mbps.



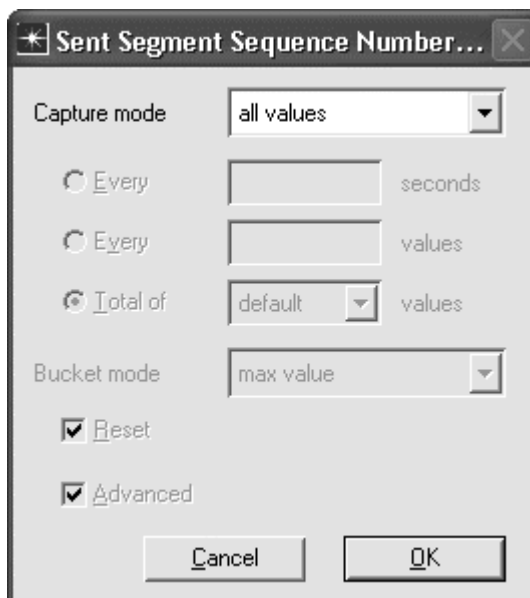
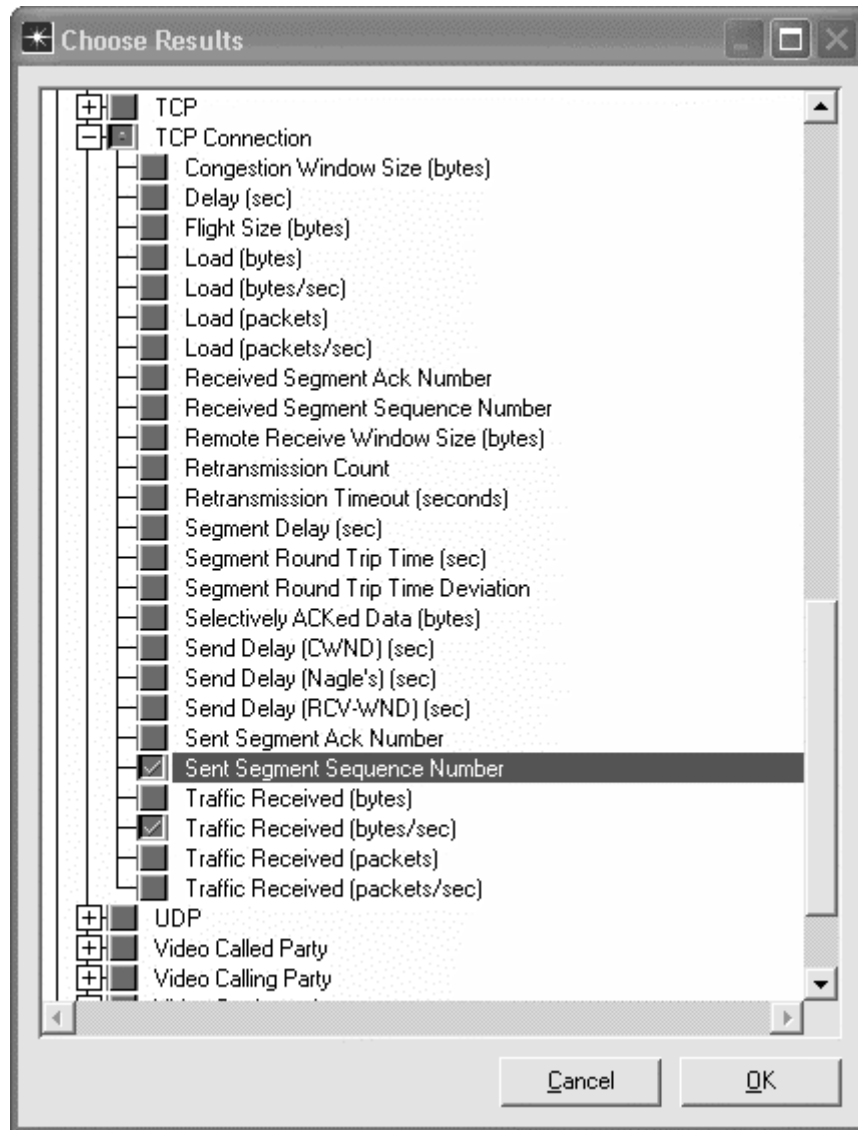
Configure and Run the Simulation

Select the **Simulation** tab => **Choose Individual Statistics...**

Expand the **Global Statistics** item and the **FTP** item, and select the **Download Response Time (sec)** statistic.

Expand the **Node Statistics** item and the **TCP Connection** item, and select the **Sent Segment Sequence Number** and **Traffic Received (bytes/sec)** statistics. Right-click on the **Sent Segment Sequence Number** statistic, and choose **Change Collection Mode**. Click on **Advanced**, and set the **Capture mode** to **all values**. This will ensure that IT Guru will store the sequence number of every packet sent to give us a more detailed graph.

Click on **OK** to close window.

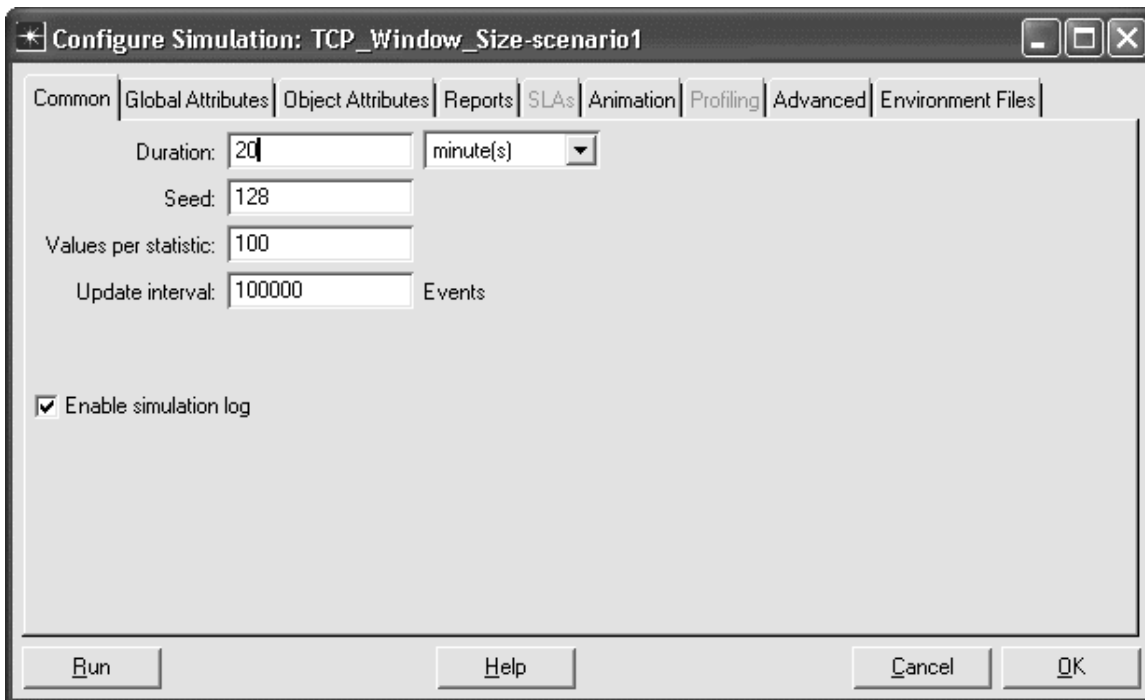


Select **Simulation => Configure Discrete Event Simulation...**

Under the **Common** tab, set the **Duration** to **20**, and the unit to **minute(s)**.

Click on **Run** to run the simulation.

When the simulation has completed, click on **Close** to close the window.

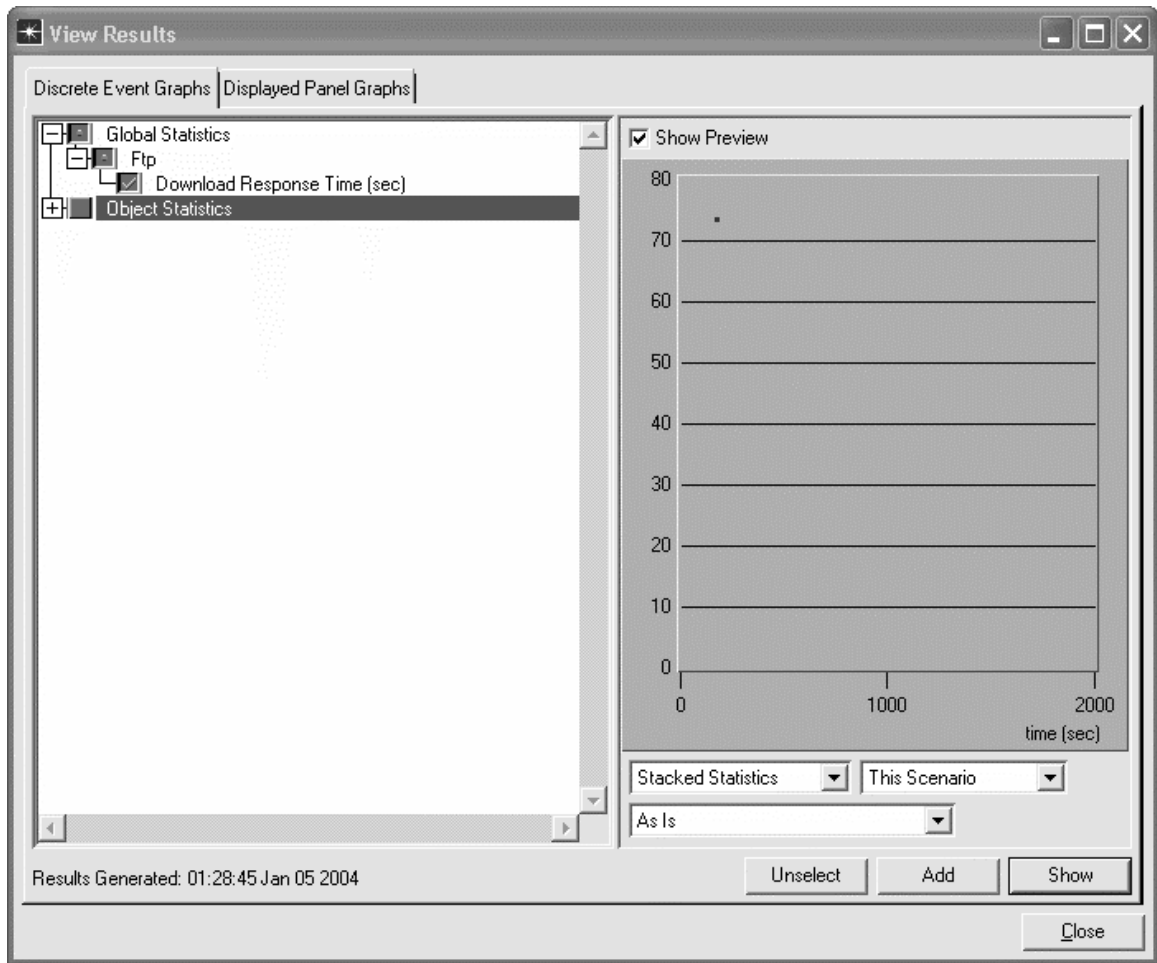


Inspect and Analyze Results

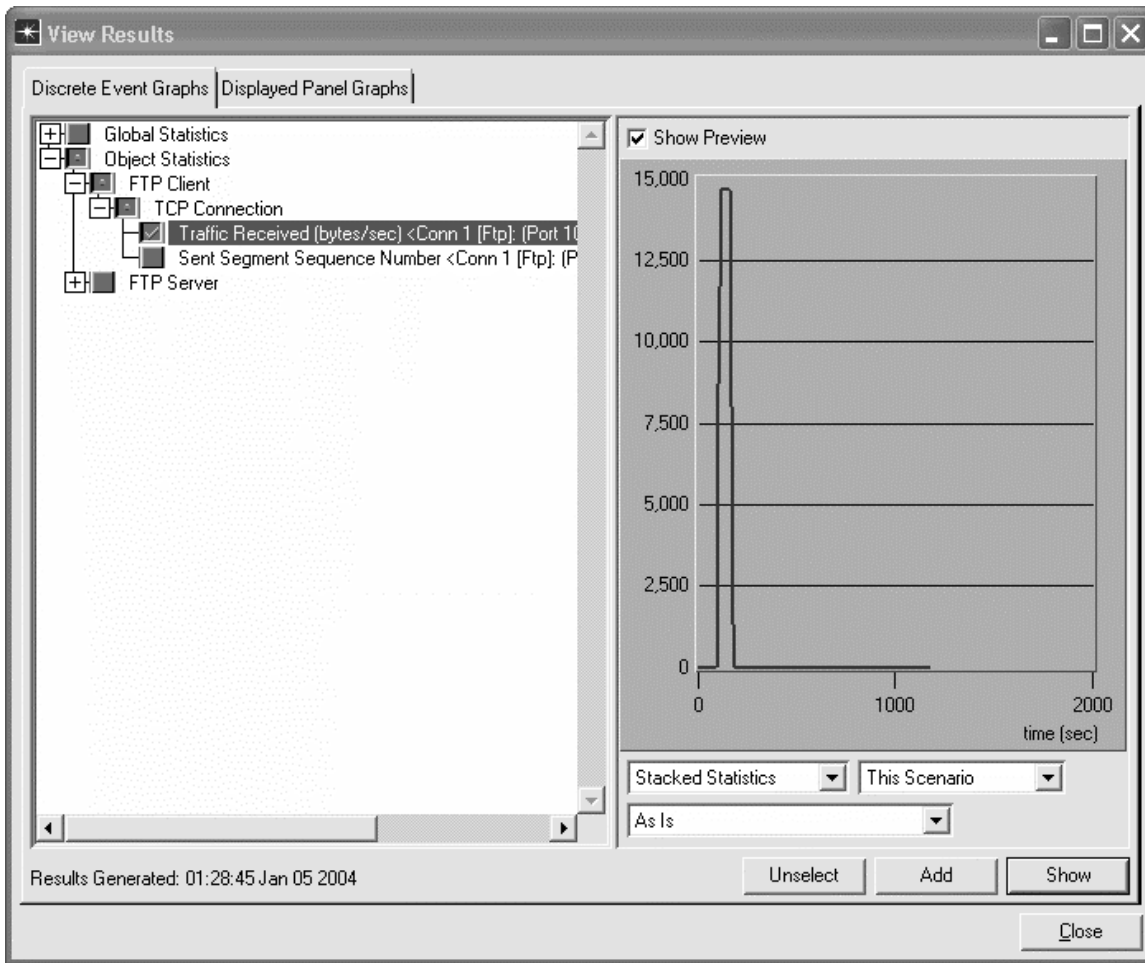
Select the **Results** tab => **View Results...**

Select and expand the **Global Statistics** item and the **Ftp** item, and select the **Download Response Time (sec)** statistic. This statistic shows how long the download took. There will be one point for each file downloaded. Click on the statistic again to disable the preview.

We can calculate the TCP throughput from the download response time. We set the file size to be 1 Mbyte when we configured the FTP application $[(1 \text{ Mbyte} * 8 \text{ bits/byte}) / 73 \text{ seconds} = 109.6 \text{ Kbps}]$.



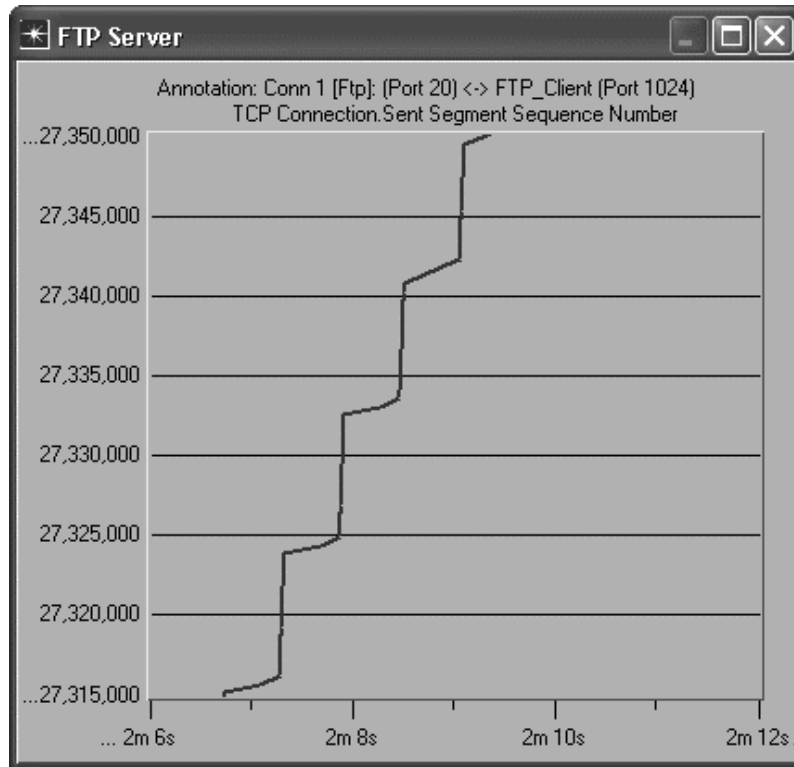
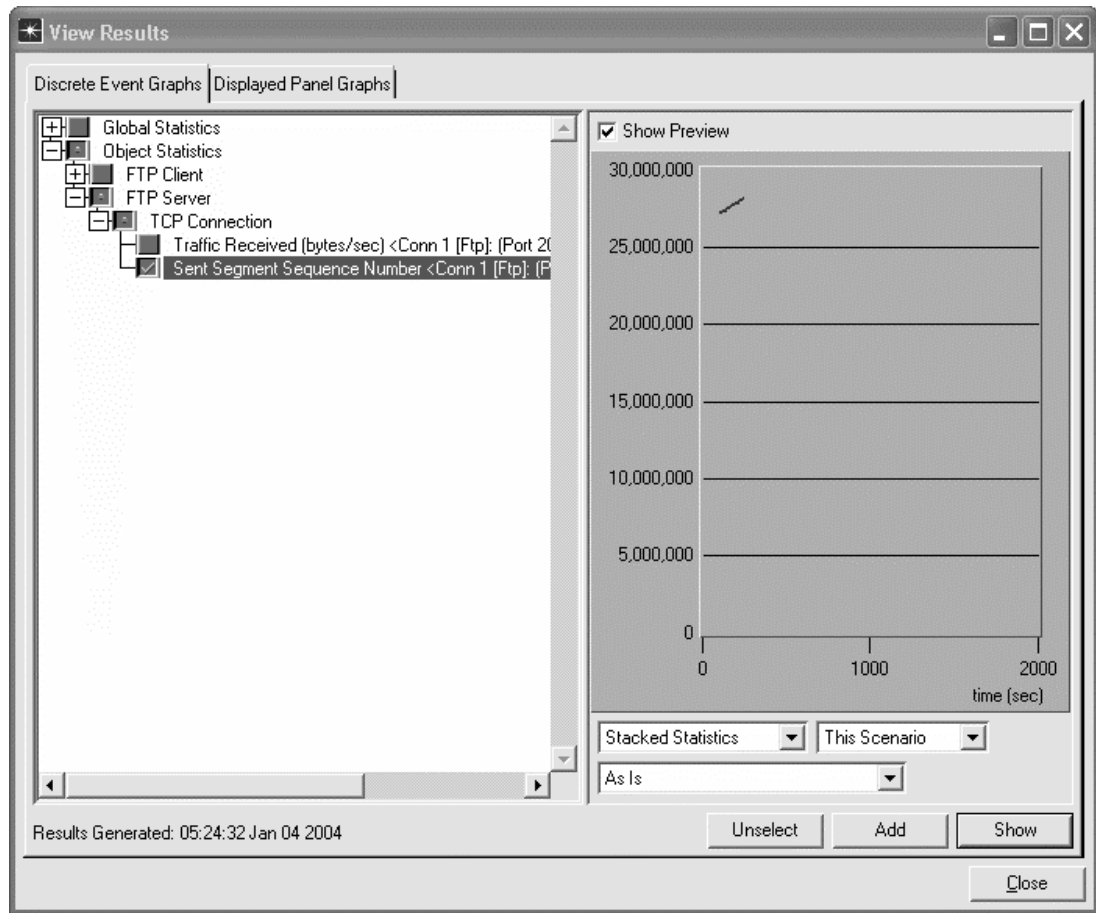
Select and expand the **Object Statistics** item, the **FTP Client** item, and the **TCP Connection** item. Select the **Traffic Received (bytes/sec)** statistic under the FTP Server. You can see the measured throughput over the length of the FTP connection. Note that measured throughput of 14,500 bytes/sec is roughly equal to the throughput we just calculated ($14,500 \text{ bytes/sec} * 8 \text{ bits/byte} = 116 \text{ Kbps}$). This connection, however, still consumes much less than the capacity of the DS1 lines which connect the server and client to the cloud. Click on the statistic again to disable the preview.



Expand the **Object Statistics**, **FTP Server** and **TCP Connection** items, and select the **Sent Segment Sequence Number** statistic. Click on the **Show** button and expand a small portion of the graph so that you can see more detail. From this graph, you can see the rate at which sequence numbers are used at the source. This rate is directly related to the throughput of the connection. You can also see that the server alternates between periods when it is actively sending (the sequence numbers increase) and periods when it is waiting for acknowledgements from the client (the sequence numbers stay the same). The throughput of this connection is low because the window size is not large enough. The TCP source sends a full window of packets (8 KB), but then must wait for acknowledgements before it can send again.

Note that you may right click on the graph and select **Export Graph Data to Spreadsheet**. This will allow you to examine the actual time value/ sequence number pairs in more detail.

Click on the close window icon and choose to **Delete** the panel to close the statistics window. Click on **Close** to close the results window.



Save your model and close all windows.

Questions

1. Duplicate the scenario and rerun the simulation using client TCP window sizes of 16 KB, 24 KB, 32 KB, 48 KB, and 64 KB. Note the average file download time for each window size. Plot the values using a spreadsheet. How does the window size affect the download time?
2. For the same download times you found in Question 1, calculate the corresponding TCP throughputs.
3. Explain the advantages of implementing a small receive window size. Explain the advantages of implementing a large window size.
4. What would be the effect of increasing the capacity of the lines connecting the server and client to the IP cloud to DS-3 speeds (45 Mbps)? Duplicate the scenario and rerun the simulation to verify your hypothesis and provide statistics to support your answer.
5. What would be the effect of decreasing the capacity of the lines connecting the server and client to the IP cloud to DS-0 speeds (64 Kbps)? Duplicate the scenario and rerun the simulation to verify your hypothesis and provide statistics to support your answer.