M/M/1 Queue

Introduction

An M/M/1 queue consists of a first-in-first-out (FIFO) buffer with packets arriving randomly according to a **Poisson** process, and a processor, called a server, that retrieves packets from the buffer at a specified service rate. In this tutorial, you will explore the Node Editor and how it can be used to create an M/M/1 queue. You will also

- Use the Probe Editor to gather different types of statistics
- Learn how to apply filters to data gathered during simulation
- Mathematically analyze the statistical data from the simulation

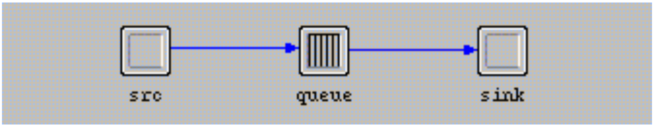
In this tutorial, you will use the Node, Project, Probe, and Analysis editors to construct an M/M/1 queue, collect statistics about the model, run a simulation, and analyze the results.

The performance of an M/M/1 queuing system depends on the following parameters:

- Packet arrival rate
- Packet size
- Service capacity

If the combined effect of the average packet arrival rate and the average packet size exceeds the service capacity, the queue size will grow indefinitely. In this tutorial, you will build an M/M/1 queue model and make sure the queue reaches steady state for a specific arrival rate, packet size, and service capability.

M/M/1 Queue Model



Getting Started

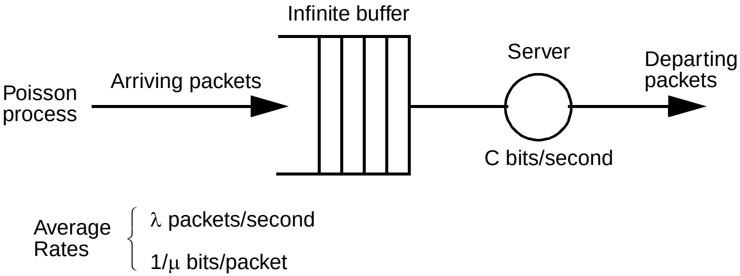
When modeling any network, you must have a clear understanding of the network to be modeled and the questions to be answered. In this section, you will explore

- The M/M/1 queue system
- The types of node modules used to model this system
- The distributions that can be used for this model

The M/M/1 queue is generally depicted by a Poisson process governing the arrival of packets into an infinite buffer. When a packet reaches the head of the buffer, it is processed by a server and sent to its destination.

The M/M/1 queue system is shown in the following figure:

M/M/1 Queue



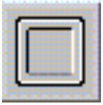
λ, $1/\mu$, and C are representative of mean packet arrival rate, mean packet size (service requirement), and service capacity, respectively.

Thus, the queue model requires a means of generating, queuing, and serving packets, all of which can be done with existing node modules provided in the Node Editor.

To create the M/M/1 queue, you will need several objects in the Node Editor, including one **queue** and two **processors**. The **source module** generates packets and the **sink module** is used to dispose of the packets generated by the source. The **queue** represents the infinite buffer and the server.

The **processor** module is the general-purpose building block of the node model. Setting the **process model** attribute of the module determines the behavior of a particular processor module.

Processor Module



The **source** module (which the processor module represents) generates packets and specifies the creation rate of generated packets in terms of **packets/second**. This rate is specified by an **exponential** distribution of **packet interarrivals** (that is, the time between packet arrivals at the queue).

The **queue** module can be used to represent the infinite buffer and also the server (the reasons for using the queue to represent the server will be discussed later).

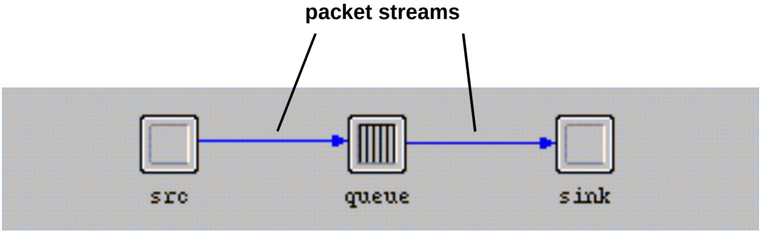
Queue Module

Although it is not a part of the M/M/1 queue system described earlier, a **sink** module will be used in the node model to dispose of serviced packets. Destroying packets that are no longer needed frees up memory to be reused during simulation.

Packet streams are used to connect each of the modules in the Node Editor.

To transfer packets between the generator module, the queue module, and the sink module, you use **packet streams**. These are the paths through which packets move from one module to another.

Packet Streams Move Packets Between Modules



Now that all the elements necessary to build this node model have been discussed, you can begin model construction. Note that the network model (in the Project Editor) will contain one instance of the M/M/1 queue node model. No other node or network objects are required to complete the modeling goals, and a network model consisting of a single “stand-alone” node is valid.

To set up a new project and scenario for this lesson:

1. Close any existing projects.

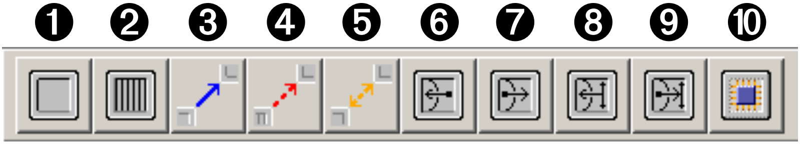
2. Open a new project and a new scenario. Name the new project **<initials>_mm1net** and the scenario **mm1**.

3. In the Startup Wizard, click **Quit**. You will set up the scenario after the node model has been created.

4. Select **File > New...**, then select **Node Model** from the pull-down list and click **OK**.

   ➥ The Node Editor opens as a new window.

Node Editor Toolbar Buttons



Node Editor Toolbar Button Names

❶	Create Processor	❻	Create Point-to-point Receiver
❷	Create Queue	❼	Create Point-to-point Transmitter
❸	Create Packet Stream	❽	Create Bus Receiver
❹	Create Statistic Wire	❾	Create Bus Transmitter
❺	Create Logical Tx/Rx Association	❿	Create External System Module

You will use many of these buttons in the tutorial. If you have the Wireless module, you will see additional buttons.

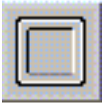
The Node Editor

The Node Editor is used to build node models, which consist of **modules** connected by **packet streams** and **statistic wires**.

The first step to creating the M/M/1 queue is to define the source module that randomly generates packets. For this, you can use the **processor module**.

1 Click on the **Create Processor** toolbar button.

Create Processor Toolbar Button



2 Click in the workspace where you want the module to be placed.

➥ A processor appears in the Node Editor workspace.

3 Right-click to end the operation.

Modules in the Node Editor have attributes just like network objects in the Project Editor. To specify the processor's generation rate, generation distribution, average packet size, and packet size distribution, you must set these attributes in the processor module.

1. Right-click on the processor module and select **Edit Attributes**.

2. Change the **name** attribute to **src** (for source) by left-clicking in the Value column, typing **src**, then pressing **Return**.

3. Change the **process model** attribute to **simple_source**. You may have to scroll to see this model.

   ➥ Generator attributes appear in the attribute list.

4. Left-click in the Value column of the **Packet Interarrival Time** attribute to open the **Packet Interarrival Time Specification** dialog box.

5. Select **exponential** from the **Distribution Name** pop-up menu. This sets the interarrival times of generated packets to be exponentially distributed, as in a **Poisson** process.

6 Set the **Mean Outcome** to **1.0** and click **OK**. This sets the mean interarrival time of a packet to 1 second.

7 Change the **Packet Size** attribute so that **Distribution Name** is **exponential** and **Mean Outcome** is **9000**.

8 Click **OK**. This sets the size of the generated packets to be exponentially distributed with a mean size of 9000 bits per packet.

9 Click **OK** to close the Attributes dialog box.

The **Packet Interarrival Time** attribute specifies the packet interarrival time and distribution. The **Packet Size** attribute specifies the packet size and distribution.

PDF stands for **probability density function**. A PDF determines the distribution of stochastic numbers. The range of possible outcomes from an **exponential PDF**, such as was set for the **Packet Interarrival Time** and **Packet Size** attributes, extends between zero and infinity, with more of the outcomes clustered near or below the specified mean value.

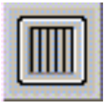
The use of exponential PDFs in the M/M/1 model is consistent with the Poisson process originally specified for the service time and packet arrival probabilities.

The packet generator portion of the M/M/1 model is complete, and during simulation will generate packets according to the exponential PDF values assigned.

The next step is to create a queue module that emulates both the infinite buffer and the server of the M/M/1 queue, as follows.

1 Click on the **Create Queue** toolbar button, and place a queue module to the right of the generator module in the workspace.

Create Queue Toolbar Button



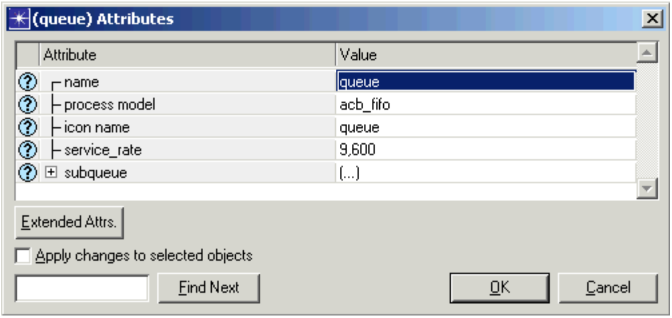
2 Right-click to end the operation.

3 Right-click on the queue module and select **Edit Attributes** to bring up its attributes.

4 Change the **name** attribute to **queue**.

5 Change the **process model** attribute to **acb_fifo** (the reasoning behind this is discussed after this procedure).

Changing the Module Attributes



Attribute	Value
name	queue
process model	acb_fifo
icon name	queue
service_rate	9,600
⊞ subqueue	(...)

□ Apply changes to selected objects

6 Make sure the **service_rate** attribute is set to **9600**.

7 Click **OK** to close the dialog box.

The **process model** attribute of the queue module sets the queue’s underlying process model, which defines the behavior of the queue module.

The underlying process model that controls our queue module’s behavior, **acb_fifo**, emulates both the infinite buffer and the server in the M/M/1 queue. If you wish to view this process model, double-click on the queue module. The Process Editor, with the queue’s underlying process, will open. However, the process used to define **acb_fifo** will not be explored in this lesson. The Basic Processes Lesson introduces the Process Editor.

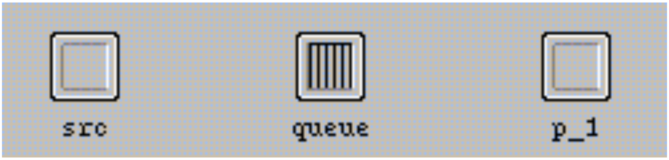
The name of the underlying process model, **acb_fifo**, reflects its major characteristics: “**a**” indicates that it is active (that is, it acts as its own server), “**c**” indicates that it can concentrate multiple incoming packet streams into its single internal queuing resource, “**b**” indicates that the service time is a function of the number of bits in the packet, and “**fifo**” indicates the service ordering discipline.

Another point to note about assigning process models to node modules is that the process model’s attributes appear in the module’s attribute list. Thus, the **service_rate** attribute’s value that you set is also set in the underlying **acb_fifo** process model.

For proper memory management, packets should be destroyed when they are no longer needed. For this task, you can use a **sink** module.

1 Click on the **Create Processor Module** toolbar button and place a processor module to the right of the queue module, as shown in the following figure.

Placing the Sink Module

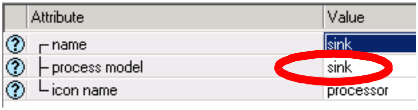


2 Right-click on the processor to open its Attributes dialog box.

3 Change the **name** attribute to **sink**.

4 Note that the default value of the **process model** attribute is **sink**.

“sink” is the Default Value



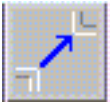
	Attribute	Value
?	┌ name	sink
?	├ process model	sink
?	└ icon name	processor

5 Click OK to close the dialog box.

Now that all the modules to emulate the M/M/1 queue have been placed in the workspace and configured correctly, they must be connected by packet streams so that the packets can be transferred from module to module.

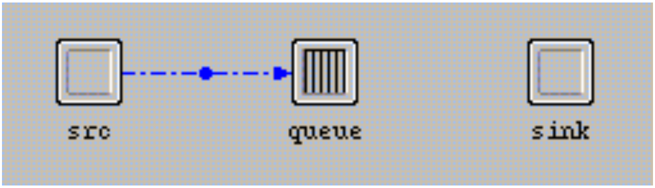
1 Click on the **Create Packet Stream** toolbar button.

Create Packet Stream Toolbar Button



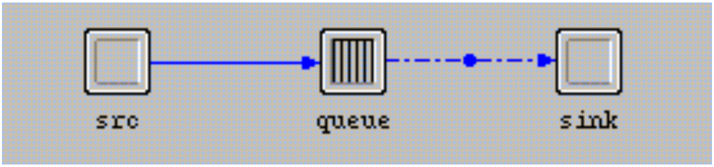
2 Connect the **src** module to the **queue** module by clicking on the **src** icon, then clicking on the **queue** icon.

Connecting src to queue



3 Connect the **queue** module with the **sink** module by clicking on the **queue** icon, then clicking on the **sink** icon. Remember to end the **Create Packet Stream** operation by right-clicking.

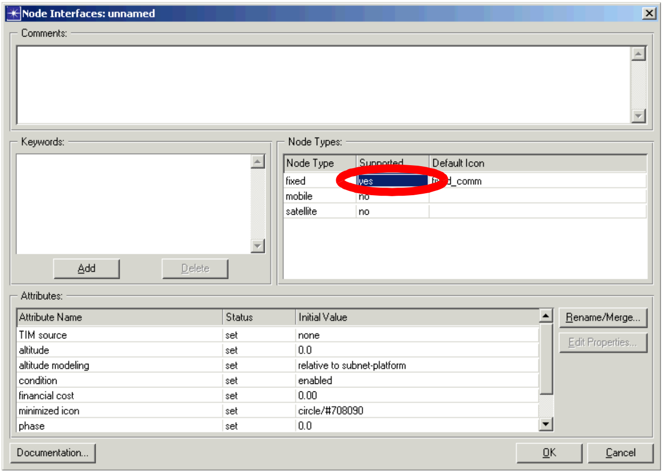
Connecting queue to sink



4 Verify that the node type is **fixed** (not mobile or satellite).

4.1 Select **Interfaces > Node Interfaces**.

Supporting the Fixed Node Type



Node Type	Supported	Default Icon
fixed	yes	fixed_comm
mobile	no	
satellite	no	

Attribute Name	Status	Initial Value
TIM source	set	none
altitude	set	0.0
altitude modeling	set	relative to subnet-platform
condition	set	enabled
financial cost	set	0.00
minimized icon	set	circle/#708090
phase	set	0.0

4.2 In the **Node Types** table, change the **Supported** value to **no** for the mobile and satellite types.

4.3 Click **OK** to close the **Node Interfaces** dialog box.

You have finished creating the node model and are ready to create the network model. First, you should save your work:

1. Select **File > Save**. Name the node **<initials>_mm1**, then click **Save**.

2. Close the Node Editor.

Creating the Network

The network model for the M/M/1 queue can be built as a single node object in the Project Editor.

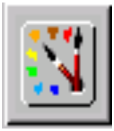
Now that the underlying node module has been built, you can create the network model in the Project Editor. Usually to create a network model, you would create a subnet (or use the Startup Wizard to set up the scenario) and place nodes within the subnet. However, because the M/M/1 model requires only a single non-communicating node, the node location is irrelevant. Thus you can place it in the top (global) subnet. This topic focuses on

- Creating a custom object palette
- Specifying the network model

The first step in building this network model is to create a new model list. You can then specify this list, customizing the palette to display only the network objects you need. To construct your own model list:

1 Click on the **Open Object Palette** toolbar button.

Open Object Palette Toolbar Button

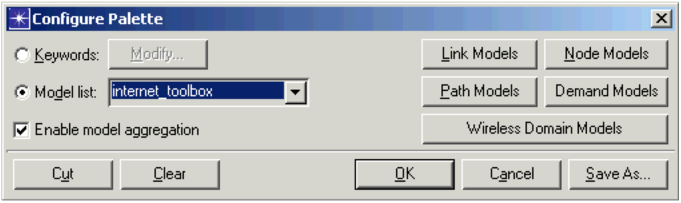


2 Click on the **Configure Palette...** button in the object palette.

➥ This brings up a number of options that allow you to customize your palette.

3 Click on the **Model list** radio button if it is not already selected.

Selecting "Model list"

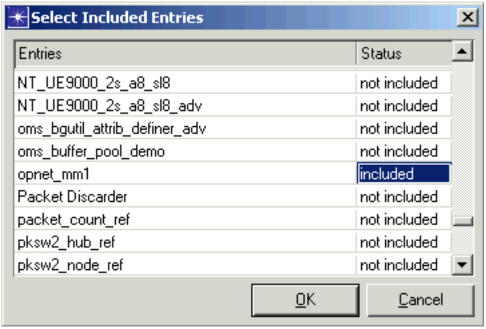


4 Click on the **Clear** button in the Configure Palette dialog box. This removes all models from the palette except for the subnet icon, which is always present.

5 Click on the **Node Models** button to bring up the list of available node models.

6 Scroll in the table until you find the **<initials>_mm1** node model. Click in the **Status** column next to **<initials>_mm1** to change it from **not included** to **included**.

Including the Custom Node Model



Entries	Status
NT_UE9000_2s_a8_sl8	not included
NT_UE9000_2s_a8_sl8_adv	not included
oms_bgutil_attrib_definer_adv	not included
oms_buffer_pool_demo	not included
opnet_mm1	included
Packet Discarder	not included
packet_count_ref	not included
pksw2_hub_ref	not included
pksw2_node_ref	not included

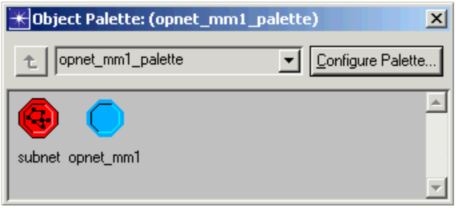
7 Click **OK** to close the table.

➥ The palette now contains an icon for the newly-included node model and, by default, a subnet icon.

8 Click **OK** in the Configure Palette dialog box to save the palette configuration. Enter the name **<initials>_mm1_palette** at the prompt and click **Save**.

➥ The Configure Palette dialog box closes. The palette now contains the node model you created in the Node Model Editor.

Custom Model included in Object Palette



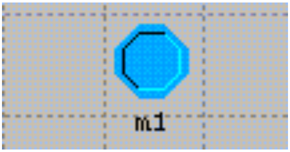
Now that you have configured a custom palette, you can create the network model.

1 Click and drag the **<initials>_mm1** node model from the object palette to the workspace. Remember to right-click to end the operation.

2 Right-click on the node object and select **Set Name** from the Object pop-up menu.

3 Enter **m1** as the name, then click **OK**.

Naming the Node



Now that you have finished creating the node, you need to set the statistics to collect during simulation. In this lesson, you will use the Probe Editor (although you could also use the Choose Results option on the object pop-up menu) to set the statistics. But first, save your network model:

Select **File > Save**. Click **Save** to accept the existing name **<initials>_mm1net**.

Collecting Results with the Probe Editor

The Probe Editor provides another option for collecting statistics from a simulation. The Probe Editor is a more powerful collection tool than the commands in the Project Editor, and can be used to customize many different types of statistics.

In this section, you will:

- Become familiar with the Probe Editor and its uses.
- Learn about different types of statistics, including some that are not available in the Project Editor.
- Configure different probes for the collection of statistics.

For this lesson, you will use the Probe Editor to collect **queue delay** and **queue size** statistics.

For the M/M/1 queue, there are several statistics to be collected, including:

- The average time a packet is delayed in the infinite buffer (**queue delay**).
- The average number of packets that are in the queue (**queue size**, in packets).

These two statistics are necessary to answer the two primary questions we have for this network model:

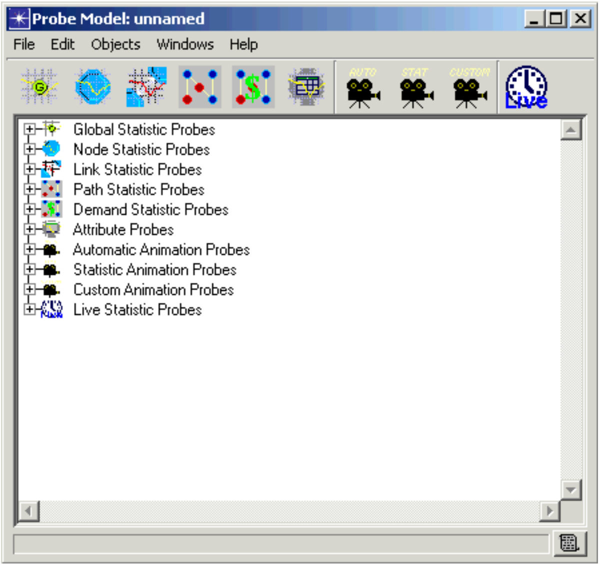
- Does the average time a packet waits in the queue exceed an acceptable limit? For this tutorial, we will say that the acceptable limit is 20 seconds.
- Does the queue size increase monotonically, or does it reach steady state? If the queue size does not reach steady state, that may be an indication that the system (especially the server, in this case) is overloaded.

In this lesson, you will use the Probe Editor to collect the **queue size** and **queue delay** statistics:

1 Select **File > New...**, then select **Probe Model** and click **OK**.

➥ The Probe Editor opens.

Probe Editor (Initial State)



When using the Probe Editor to collect statistics, you must specify the network model from which you wish to collect statistics:

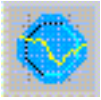
1 Select **Objects > Set Network Model...**

2 Choose the **<initials>_mm1net-mm1** network model from the list of available models.

Now that the network model is set correctly, you need to set the probes. To set the queue delay probe:

1 Click on the **Create Node Statistic Probe** button and a probe appears below **Node Statistic Probes** in the workspace.

Create Node Statistic Probe Toolbar Button



You can type in the attributes for the probe, or you can use the automatic selection method. To use the automatic selection method:

1 Right-click on the probe and select **Choose Probed Object** from the pop-up menu.

➥ The **Choose Probed Object** dialog box appears.

2 Expand the **m1** node to show its components.

3 Click on the **queue** module in the node diagram to select it.

4 Click **OK** to dismiss the dialog box.

➥ **top.m1.queue** appears in the Object column.

5 Right click on the probe and select
Edit Attributes from the pop-up menu.

6 Set the **name** attribute to **Queue Delay**.

7 Set the **submodule** attribute to **subqueue [0]**.

8 **Left-click in the Value column of the statistic row.**

➥ The Available Statistics dialog box appears.

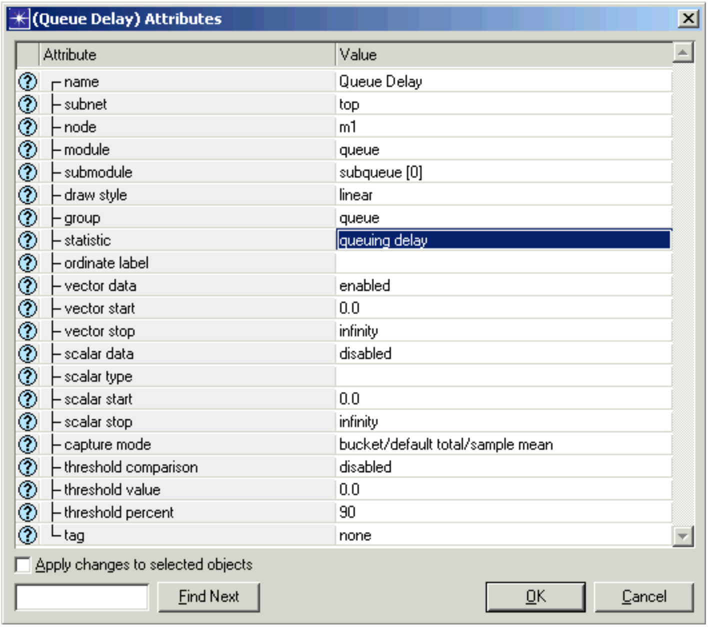
This dialog box shows the statistic and the group it belongs to (group.statistic), the statistics dimension, if any, and a description.

9 Select **queue.queuing delay** from the list and click **OK**.

➥ In the Edit Attributes dialog box, the **group** attribute changes to **queue** and the **statistic** attribute changes to **queuing delay.**

10 Click **OK** to close the probe's Attribute dialog box.

Attributes Change When Statistic is Selected



(Queue Delay) Attributes

Attribute	Value
name	Queue Delay
subnet	top
node	m1
module	queue
submodule	subqueue [0]
draw style	linear
group	queue
statistic	queuing delay
ordinate label	
vector data	enabled
vector start	0.0
vector stop	infinity
scalar data	disabled
scalar type	
scalar start	0.0
scalar stop	infinity
capture mode	bucket/default total/sample mean
threshold comparison	disabled
threshold value	0.0
threshold percent	90
tag	none

□ Apply changes to selected objects

Next, you need to specify the remaining probe for queue size in packets.

1 Click on the **Create Node Statistic Probe** toolbar button.

2 Right-click on the new probe and select **Edit Attributes.** Change the attributes of the probe to the values shown in the following table.

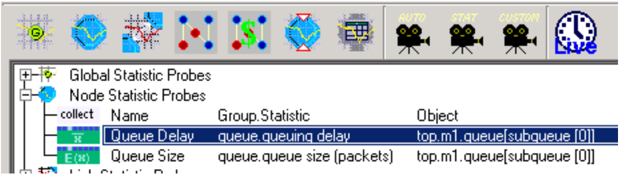
Attribute Values for Node Statistic Probe

Attribute	Value
name	Queue Size
subnet	top
node	m1
module	queue
submodule	subqueue [0]
group	queue
statistic	queue size (packets)

3 Click **OK** to close the probe's Attribute dialog box.

➥ The changes appear in the Probe Editor.

Two Node Probes in Probe Editor



Global Statistic Probes

Node Statistic Probes

collect	Name	Group.Statistic	Object
x̄	Queue Delay	queue.queuing delay	top.m1.queue[subqueue [0]]
E(x)	Queue Size	queue.queue size (packets)	top.m1.queue[subqueue [0]]

Now that you have set up the probes correctly to collect the desired statistics, save the probe file:

1 Select **File > Save**. Name the file **<initials>_mm1probe**.

2 Close the Probe Editor.

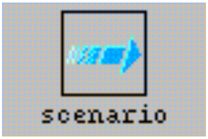
Now that you have created probes to collect statistics during simulation, you need to run the simulation. In the previous lesson, you ran the simulation from the Project Editor. For this lesson, you will use the Simulation Sequence Editor.

1 In the Project Editor, choose **DES > Configure/Run Discrete Event Simulation (Advanced)**.

➥ The Simulation Sequence Editor opens.

2 Right-click on the simulation sequence icon in the workspace and select **Edit Attributes**.

Simulation Sequence Icon



3 Set the simulation **Duration** to **7** hours.

4 Set the **Seed** to **431**.

5 Select the **Advanced** tab, then the **Files** tab and set the **probe file** to **<initials>_mm1probe.**

6 Click **Run** to begin the simulation. (You can go on to the next section while the simulation runs.)

Viewing Results

The Analysis Tool provides several features for viewing results that are not available in the Project Editor.

Depending on the speed of your machine, the simulation should take less than a minute to complete. When the Simulation Sequence dialog box shows that the simulation has completed, close the dialog box and the Simulation Sequence editor window. If you had problems, see *"Troubleshooting Tutorials"*.

In the first lesson, you viewed the results from your simulation directly in the Project Editor. You can also use the Analysis Tool to view results:

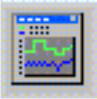
1. From the Project Editor, select **File > New**.
2. From the pull-down menu in the dialog box, change the file type to **Analysis Configuration**, then click **OK.**

   ➥ The Analysis Tool opens in a new window.

The first statistic of interest is the mean queuing delay, which shows the delay experienced by packets in the queue during the simulation. To view the mean queuing delay:

1 Left-click on the **Create a Graph of a Statistic** toolbar button.

Create a Graph of a Statistic Toolbar Button



➥ The View Results dialog box opens.

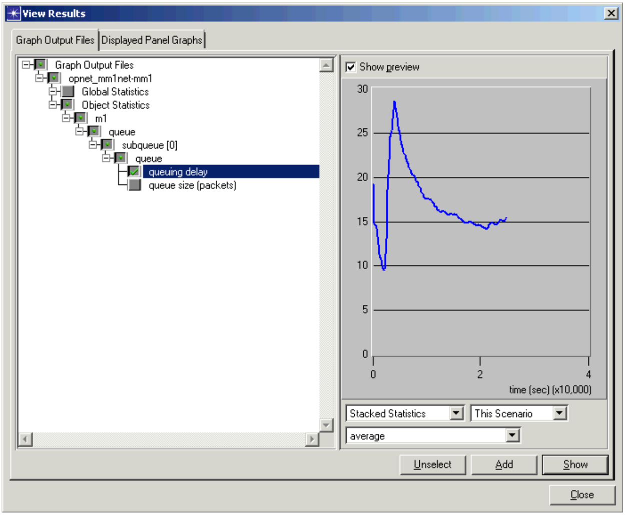
2 Expand the following hierarchy: **Graph Output Files > <initials>_mm1net-mm1 > Object Statistics > m1 > queue > subqueue [0] > queue**.

3 Select the **queuing delay** statistic.

➥ A check-mark appears in the box next to the statistic name.

4 From the pull-down list of filters, select **average**.

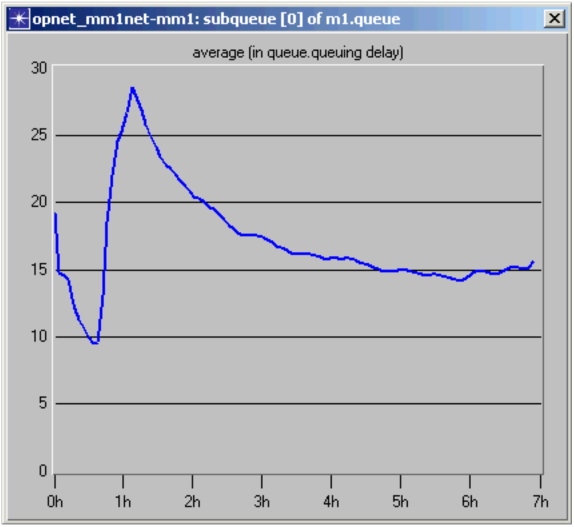
Selecting the “average” Filter



5 Click on the **Show** button (the graph is shown in the following figure).

The graph of the average queuing delay appears in the following figure:

Average Queuing Delay



The large change early in the simulation reflects the sensitivity of averages to the relatively small number of samples collected. Towards the end of the simulation, the average stabilizes.

It is important to validate your simulation results for accuracy. Note that the mean queuing delay for this simulation is around 15 seconds. There are several calculations that can be used to determine if this is accurate:

Queuing Delay Calculations

mean arrival rate:

$$\lambda = \frac{1.0}{\textit{mean interarrival time}} = \frac{1\mathrm{p}}{1.0\mathrm{s}} = 1(\mathrm{p}/\mathrm{s})$$

mean service requirement: $\frac{1}{\mu} = 9000(\mathrm{b}/\mathrm{p})$

service capacity: $C = 9600(\mathrm{b}/\mathrm{s})$

mean service rate:

$$\mu C = \left(\frac{1}{9000(\mathrm{b}/\mathrm{p})}\right)(9600(\mathrm{b}/\mathrm{s})) \cong 1.067(\mathrm{p}/\mathrm{s})$$

mean delay:

$$\overline{W} = \frac{1}{\mu C - \lambda} = \frac{1\mathrm{p}}{1.067(\mathrm{p}/\mathrm{s}) - 1(\mathrm{p}/\mathrm{s})} = \frac{1\mathrm{p}}{0.067(\mathrm{p}/\mathrm{s})} = \boxed{15\mathrm{s}}$$

Another statistic of interest is the time-averaged queue size. This statistic can be viewed by applying the time average filter to the queue size statistic collected during simulation:

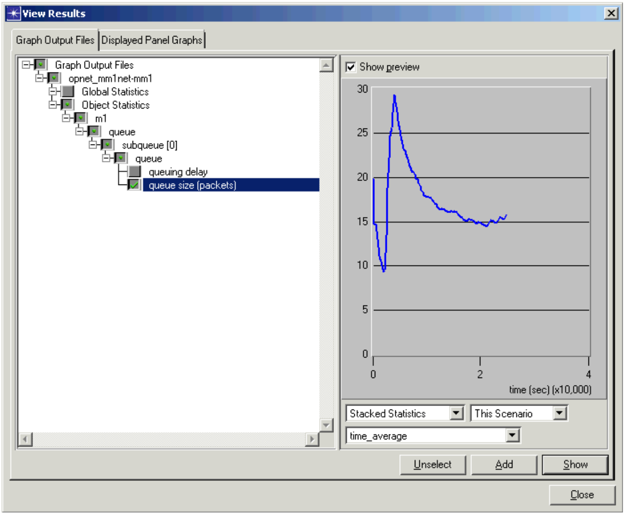
1. Move the mean queuing delay graph out of the way, but don’t close it.

2. In the View Results dialog box, click on the check box to remove the check next to the **queuing delay** statistic.

3. Place a check next to the **queue size (packets)** statistic.

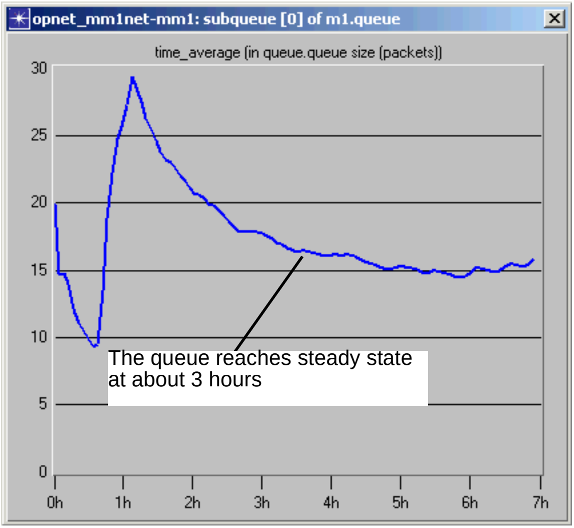
4. From the pull-down list of filters, select **time_average**.

Selecting the queue size (packets) Statistic



5 Click on the **Show** button. The graph of the time-averaged queue size appears in the following figure:

Time-averaged Queue Size



The queue reaches steady state at about 3 hours

The value of the time-averaged queue size can be compared to the expected value obtained from the formula: $\rho / (1-\rho)$, where $\rho = \lambda / (\mu C)$. This yields 15.0, which is in accord with the results produced from the simulation.

You may have noticed that the graphs of the mean queuing delay and the time-averaged queue size are very similar. It turns out that for this model, because of the large number of well-dispersed queue insertions and removals that occur during the simulation, the difference between the final values of the two statistics is negligible.

The final graph for this lesson is of the queue size versus the time-averaged queue size, plotted on the same graph:

1 Close the existing View Results dialog box.

2 Right-click on the graph of the time-averaged queue size, then choose **Add Statistic** from the pop-up menu.

➥ A new View Results dialog box appears.

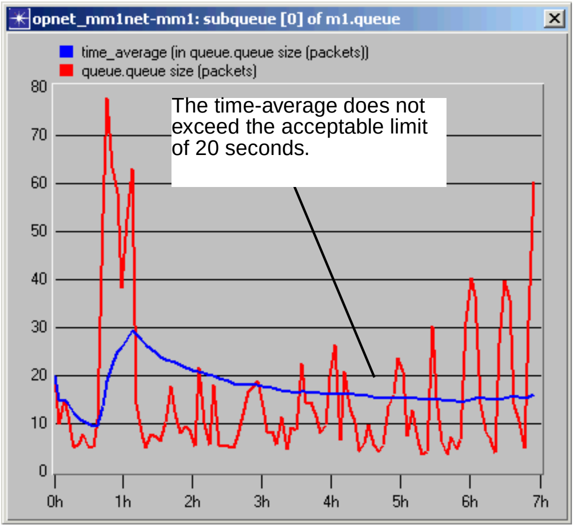
3 Place a check next to **Graph Output Files > <initials>_mm1net-mm1 > Object Statistics > m1 > queue > subqueue [0] > queue > queue size (packets)** in the View Results dialog box.

4 Left-click on the **Add** box.

➥ The graph of the **queue size (packets)** is added to the existing time-averaged queue size graph (as shown in the following figure).

The graph now displays two traces. The **queue size** shows the instantaneous number of packets in the queue during the course of the simulation. The time-averaged queue size trace depicts the average number of packets in the queue over time.

Queue size, Time-averaged and instantaneous



The time-average does not exceed the acceptable limit of 20 seconds.

From this graph, we can say that the time-average does not exceed the acceptable limit of 20 seconds, and that the queue is not monotonically increasing, as it reaches steady state at around 3 hours. Therefore, this is a stable system.

You have now completed the M/M/1 queue lesson of the tutorial. You should have a good understanding of how node models fit in the modeling scheme, and how they can be used to create complex network models. You should also be familiar with the Node Editor, the Simulation Sequence Editor, and the Analysis Tool.

The next lesson, Basic Processes, focuses on the use of process models in network modeling. If you wish to continue with the tutorial, save any work that has not yet been saved, then close all open editor windows in OPNET. Return to the main tutorial menu and choose **Basic Processes** from the list of available lessons.