Creating a Wireless Network

Introduction

In this tutorial, you will create a radio network with a mobile jamming node. You will also

- Use a new type of link, the **radio link**, and a new type of node, the **mobile node**
- Use the Antenna Pattern Editor to create a **directional antenna pattern**
- Define the **trajectory** of a mobile node
- Execute parametric simulations

With the Wireless module, OPNET can model both terrestrial and satellite radio systems. In this tutorial, you use Modeler and Wireless modeling to create a radio network and observe variations in the quality of received signal that results from radio noise at the receiving node in a dynamic network topology.

Interference (radio noise) can decrease the signal-to-noise ratio (SNR) in a radio-based network. Different types of antennas, such as directional antennas, can improve the SNR in a network.

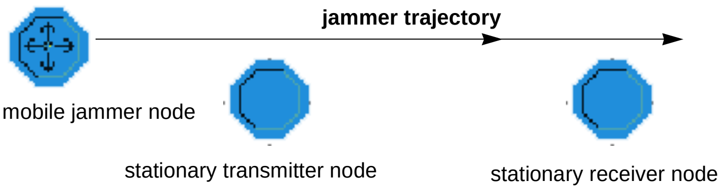
In this lesson, you will design a simple radio network with a mobile jammer node and two stationary communications nodes, then demonstrate the differences in the SNR of the network when the stationary nodes use an isotropic or directional antenna.

Getting Started

The network topology consists of three nodes:

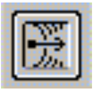
- The **transmitter node** transmits at uniform strength in all directions. It consists of a packet generator module, a radio transmitter module, and an antenna module.
- The **receiver node** measures the quality of the signal emitted by the stationary transmitter node. It consists of an antenna module, a radio receiver module, a sink processor module, and an additional processor module that works with the directional antenna.
- The **mobile jammer node** creates radio noise. The jammer's trajectory takes it in and out of the radio range of the receiver node, increasing and decreasing interference at the receiver.

Wireless Topology

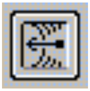


To create the node models with each of the network objects, you will need several new objects in the Node Editor, including an **antenna module** to model directional gain, **radio transmitter** and **receiver** modules, and a **processor module**.

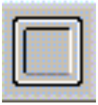
The **antenna module** models the directional gain of a physical antenna by referencing its pattern attribute. The antenna uses two different patterns: the isotropic pattern (which has uniform gain in all directions) and a directional pattern that you will define.



The **radio transmitter module** transmits packets to the antenna at 1024 bits/second, using 100 percent of its channel bandwidth.



For each arriving candidate packet, the **radio receiver module** consults several properties to determine if the packet's average bit error rate (BER) is less than a specified threshold. If the BER is low enough, the packet is sent to the sink and destroyed.



The **processor module** (called a “pointing processor” in this tutorial due to its function) calculates the information that the antenna needs to point at a target: latitude, longitude, and altitude coordinates. The pointing processor makes this calculation by using a Kernel Procedure that converts a node’s position in a subnet (described by the x position and y position attributes) into the global coordinates that the antenna requires.

Radio links exist between radio transmitter-receiver channel pairs and are dynamically established during simulation; these links are not visible in any editor.

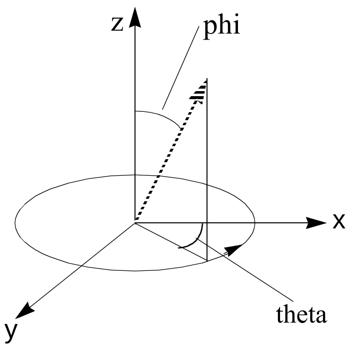
In this lesson, information is transferred from a stationary transmitter object to a stationary receiver object. These objects are connected by a **radio link**. This link depends on many different physical characteristics of the components involved, including frequency band, modulation type, transmitter power, distance, and antenna direction.

The Antenna Pattern Editor

The OPNET **Antenna Pattern Editor** uses the spherical angles **phi** and **theta** to graphically create a 3-dimensional antenna pattern.

OPNET divides an antenna pattern into segments for values of the spherical angles **phi** and **theta**. The constant values of **phi** represent approximate two-dimensional (2D) cone-shaped surfaces which are mapped into cartesian coordinates and described by 2D numeric functions called slices or planes. For each 2D slice, the function's abscissa is **theta**, and its ordinate is the associated gain value. The three-dimensional (3D) antenna pattern function is thus represented as a collection of 2D slices, as shown in the following figure.

Representation of an Antenna Pattern



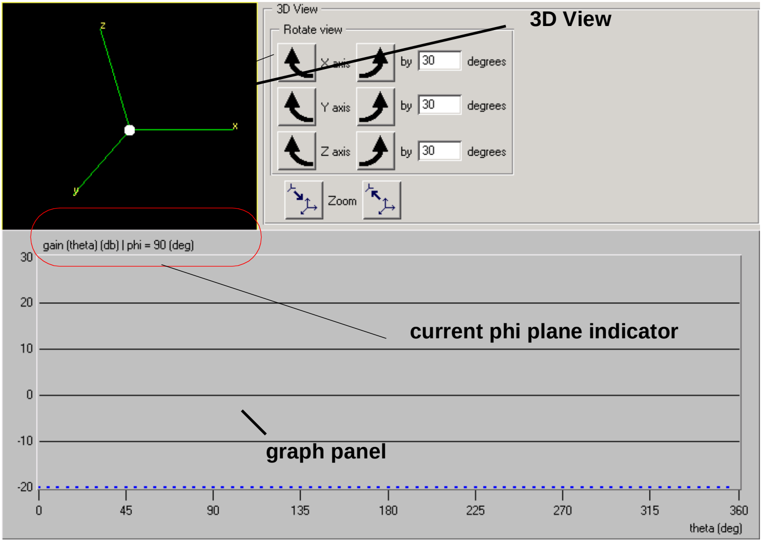
Each slice is shown in a graph panel in which sample points specify gain values for varying degrees of **theta.** You use the phi plane operations menu to select which 2D function slice, or value of **phi**, is displayed for editing.

For this lesson, you will create a new antenna pattern, one with a gain of about 200 dB in one direction and gain of about 0 dB in all other directions (a *very* directional antenna).

1. Choose **File** > **New...**, then select **Antenna Pattern** from the pull-down list and click **OK**.

   ➥ The Antenna Pattern Editor opens as a new window.

Antenna Pattern Editor



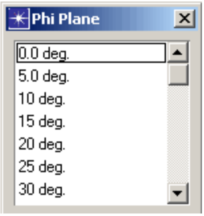
For this tutorial, you can use the default number of **theta** divisions (72), so the largest value of **theta** that has a sample point is 355 degrees. You can specify sample points with gain equal to about 200 dB for values of **theta** from 0 to 355 degrees. Specifying any two sample points in the graph panel automatically sets all sample points in between with linearly-interpolated gain values. Therefore, you need to specify only two sample points in this slice: one at 0 degrees and one at 355 degrees.

To adjust the current slice setting to 5 degrees (360/72), do the following steps:

1 Right-click in the graph panel and select **Set Phi Plane** from the Workspace pop-up menu.

➥ A menu with degree options appears.

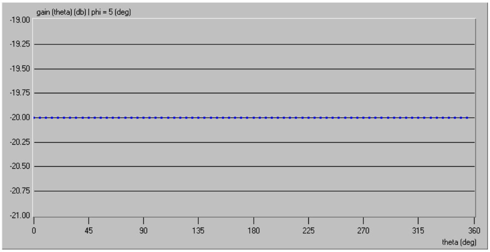
Phi Plane Dialog Box



2 Select **5.0 deg.** from the menu.

➥ The menu closes and the graph panel displays the 2D slice curve for the slice designated by phi = 5 degrees. The function label at the top of the panel displays the current phi setting (5 degrees).

Slice Displayed



Next, set the ordinate bounds:

1 Click on the **Set Ordinate Upper Bound** toolbar button.

2 In the dialog box, enter **201** as the Ordinate Upper Bound and click **OK**.

3 Click on the **Set Ordinate Lower Bound** toolbar button.

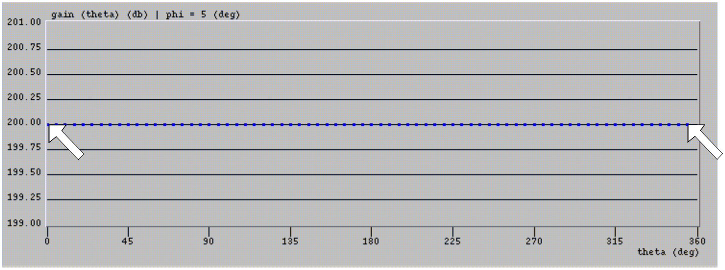
4 In the dialog box, enter **199** as the Ordinate Lower Bound and click **OK**.

➥ The graph panel displays the new ordinate range. This range will make it easier to enter the desired gain accurately.

Now that you have set the graph panel, specify sample points for phi = 5 degrees, as follows:

1 Move the cursor as close to the 200 dB line as possible and left-click on the first sample point (0 degrees) in the graph. Move the cursor to the far right (still on the 200 dB line) and left-click on the second point (355 degrees).

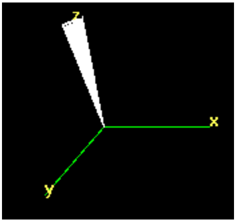
Specifying Sample Points



➥ All sample points in between the two specified points are set automatically with linearly interpolated gain values. A dotted line marks the range of sample points.

When you define points in the graph panel, the 3D projection view displays a cone-shaped shell of gain values for phi = 5 degrees to phi = 10 degrees and for theta = 0 degrees to theta = 360 degrees.

3D Projection View



Now that you have specified the gain values for phi = 5 degrees, you need to change the slice setting to 0 degrees, then set the gain and sample points for this slice. Doing so specifies a gain of about 200 dB for **phi = 0–5 degrees** and for **theta = 0–360 degrees**. This “fills in” the cone-shaped shell specified in the phi = 5 degrees plane.

1. Right-click in the graph panel and select **Decrease Phi Plane** from the Workspace pop-up menu.

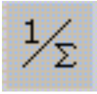
   ➥ The current phi plane setting changes from 5 degrees to 0 degrees.

2. Set the Ordinate Upper Bound to **201** and the Ordinate Lower Bound to **199**.

3. Move the cursor as close to the 200 dB line as possible and left-click on the first sample point (0 degrees) in the graph. Move the cursor to the far right (still on the 200 dB line) and left-click on the second point (355 degrees).

Normalize the function over the entire pattern, as follows:

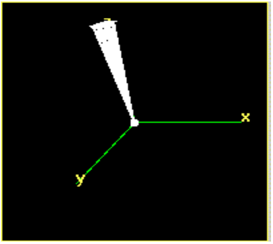
1 Click the **Normalize the Function** toolbar button to normalize the 3D gain function over the entire pattern.



➥ The 3D projection view updates, displaying the result of normalization. The small spherical component pattern shows the sub-unity gain samples required to normalize the pattern so that it integrates to 0 dB.

Normalization shifts the points in the graph upward, so they might disappear from view.

Updated 3D Projection View



2 Choose **File** > **Save**. Name the antenna pattern **<initials>_mrt_cone**.

3 Close the Antenna Pattern Editor.

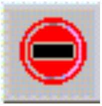
Creating the Pointing Processor

The antenna pointing processor calculates the position of the transmitter module and sets the antenna module's targeting attributes. It receives only a begin-simulation interrupt, so it can be designed as a single unforced state.

1 Choose **File > New...**, then select **Process Model** from the pull-down list and click **OK**.

➥ The Process Model Editor opens as a new window.

2 Using the **Create State** toolbar button, place one state in the workspace.



3 Right-click on the state and select **Set Name** from the Object pop-up menu.

4 Name the state **point.**

The process model determines the identities of the objects of interest, then retrieves and modifies the object's attribute values. Kernel Procedures in the Identification and Topology Packages (prefixed with **op_id** and **op_topo**) do the first task. A Kernel Procedure from the Ima Package (with the prefix **op_ima**) does the second task.

Import the code for the process model:

1 Double-click on the top half of the **point** state to open the enter executives block.

2 Choose **File** > **Import…**. Select the file listed below, and click on the **Open** button to import it (**OK** on UNIX platforms).

<reldir>\models\std\tutorial_req\modeler\mrt_ex

➥ The file is imported.

3 Review the code before you continue.

4 Save the enter executives block.

Next, you need to modify the process attributes:

1 Choose **Interfaces** > **Process Interfaces**.

➥ The Process Interfaces dialog box appears.

2 Change the initial value of the **begsim intrpt** attribute to **enabled**.

3 Change the **Status** of all the attributes to **hidden**.

4 Save your changes by clicking on the **OK** button.

Finally, compile the process model:

1 Left-click on the **Compile Process** toolbar button. When you are prompted to save the model, name it **<initials>_mrt_rx_point** and click **Save**.

If the model does not compile, see the **Troubleshooting** chapter in the **Modeling Concepts** manual.

2 After the process model has compiled, close the compilation dialog box and the Process Editor.

Creating the Node Models

You need three node models to build the radio network model: a **transmitter**, a **receiver**, and a **jammer** node.

The Transmitter Node

The transmitter node model consists of a packet generator module, a radio transmitter module, and an antenna module. The packet generator generates 1024-bit packets that arrive at the mean rate of 1.0 packets/second with a constant interarrival time (these are the default values).

After they are generated, packets move through a packet stream to the radio transmitter module, which transmits the packets on a channel at 1024 bits/second using 100 percent of the channel bandwidth. The packets then pass from the transmitter through another packet stream to the antenna module.

The antenna module uses an isotropic antenna pattern (this is the default value) to apply a transmission gain which is uniform in all spatial directions.

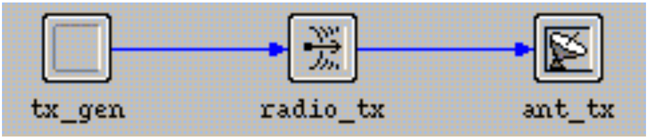
To create the transmitter node model:

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

➥ The Node Model Editor opens as a new window.

2 Create the modules and packet streams as shown, and name the nodes accordingly. Use the processor, radio transmitter, antenna, and create packet stream buttons.

Transmitter Node Model



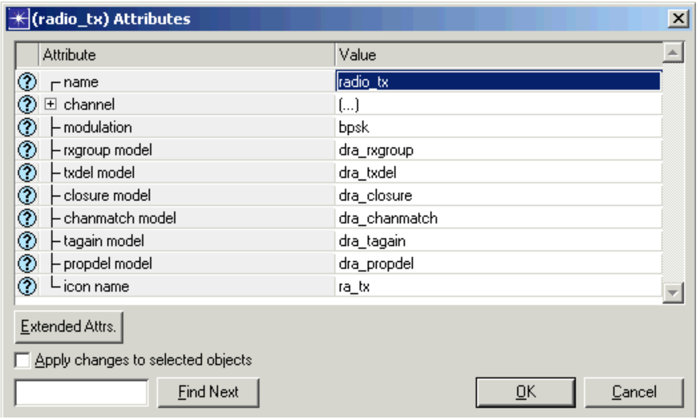
3 Change the **process model** attribute of the tx_gen processor to **simple_source**.

To run parameterized simulations, you must promote the **power** attribute of the utilized channel. When you promote the attribute, it can be changed easily at simulation run time.

1 Right-click on the **radio_tx** node and select **Edit Attributes** from the pop-up menu.

➥ The Transmitter Attributes dialog box appears.

Transmitter Attributes Dialog Box

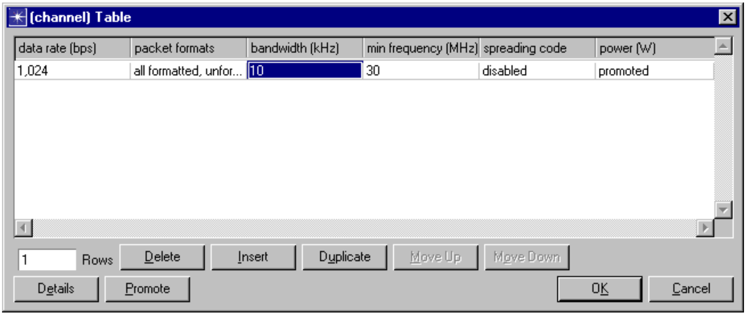


2 Click on the Value field for the **channel** attribute.

➥ A dialog box appears showing the Compound Attribute Table for **channel**.

3 In the Compound Attribute Table for channel, promote the **power** attribute by selecting its value, then clicking on the **Promote** button.

Promoting the Power Attribute



➥ The word **promoted** appears as the value for **power**.

4 Click **OK** to close both dialog boxes.

Next, define the node model interface attributes:

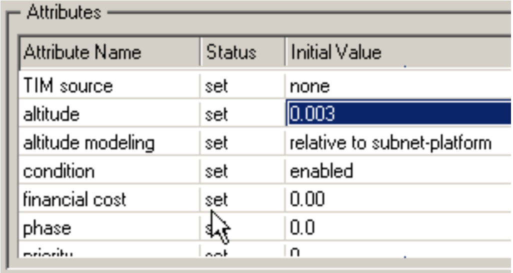
1 Choose **Interfaces** > **Node Interfaces**.

➥ The Node Interfaces dialog box appears.

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

3 In the **Attributes** table, change the **altitude** initial value to **0.003**.

Changing the Altitude Value

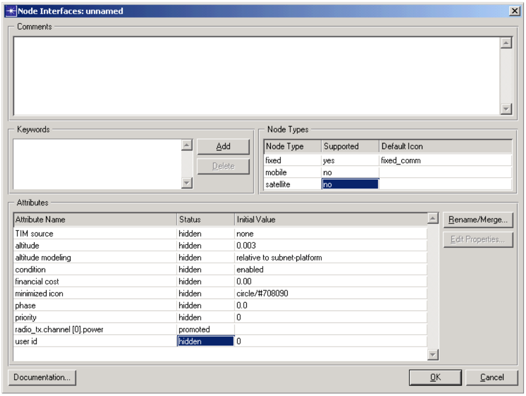


4 Except for the promoted **radio_tx.channel [0].power** attribute, change the **Status** of all attributes to **hidden**.

5 For reference, add a comment to describe the node.

The Node Interfaces dialog box should look like this:

Completed Node Interfaces Dialog Box



6 Save your changes by clicking the **OK** button.

7 Save the node model. Choose **File** > **Save**. Name the model **<initials>_mrt_tx**.

The Jammer Node

The network jammer node introduces radio noise into the network. Like the stationary transmitter node, it consists of a packet generator module, a radio transmitter module and an antenna module. Its behavior is similar to that of the stationary transmitter node, but channel power and signal modulation are different. These differences will make packets transmitted by the jammer node sound like noise to the receiver. The jammer node model is created from a copy of the transmitter node model (<initials>_mrt_tx).

1 Open the **<initials>_mrt_tx** node model if it is not open.

2 Right-click on the **radio_tx** object and select **Edit Attributes** from the pop-up menu. Change the **modulation** attribute to **jammod**.

3 Click **OK** to close the **radio_tx** attribute dialog box.

4 Choose **Interfaces** > **Node Interfaces,** then do the following:

4.1 Change the Supported value to **yes** for mobile type and **no** for fixed type

4.2 Modify the comment to describe the jammer node

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

5 Choose **File** > **Save As...** and save the file as **<initials>_mrt_jam**.

The Receiver Node

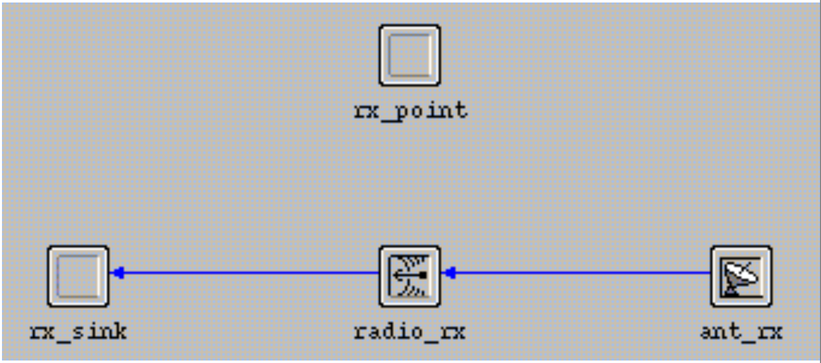
The receiver node consists of an antenna module, a radio receiver module, a sink processor module, and the pointing processor module, which helps to point the directional antenna towards the transmitter.

1 Choose **Edit** > **Clear Model**.

2 Create the modules and packet streams as shown. Set node names accordingly.

Make sure the antenna module has the name **ant_rx**. This name is referenced by the **<initials>_mrt_rx_point** process model.

The Receiver Node



Change the following attributes:

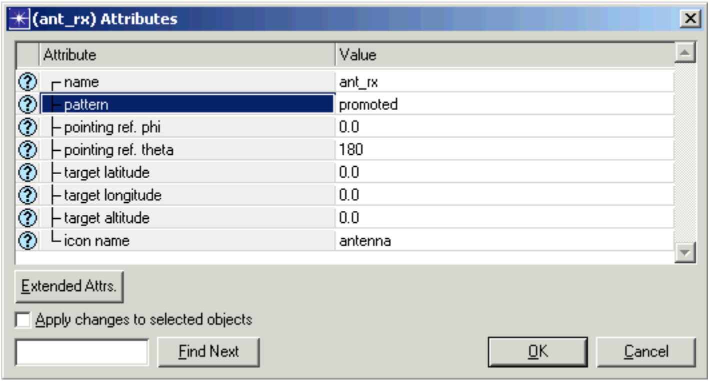
1 Right-click **rx_point** and open its attribute dialog box. Set the value of the **process model** attribute to **<initials>_mrt_rx_point**, then click **OK** to close the dialog box.

2 Right-click **radio_rx** and open its attribute dialog box. Set the value of the **error model** attribute to **dra_error_all_stats**, then click **OK** to close the dialog box.

3 Right-click on **ant_rx** and open its attribute dialog box. Click on **pattern** in the left column to highlight the Attribute name, then right-click and select **Promote Attribute to Higher Level** from the pop-up menu.

➥ The word **promoted** appears in the Value cell of the attribute.

Pattern Attribute is Promoted



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

Next, define the node model interface attributes:

1. Choose **Interfaces** > **Node Interfaces**.
2. In the **Node Types** table, change the **Supported** value to **no** for the **mobile** and **satellite** type.
3. Change the **altitude** attribute to **0.003**.
4. Except for the promoted **ant_rx.pattern** attribute, change the **Status** of all attributes to **hidden**.
5. Save your changes by clicking on the **OK** button.
6. Choose **File** > **Save As...** and save the node model as **<initials>_mrt_rx**. Close the Node Editor.

Creating the Network Model

Now that you have created all the necessary node and process models, you can create the network model.

1 Choose **File** > **New...** Select **Project** from the list of options, then click **OK**.

2 Name the new project **<initials>_mrt_net**, and the scenario **antenna_test**.

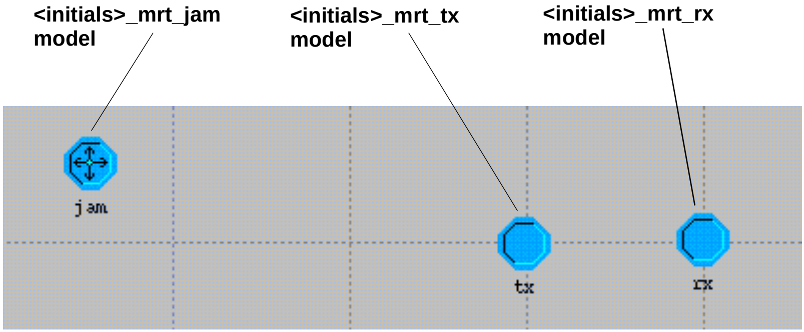
3 In the Startup Wizard, use the following settings:

Dialog Box Name	Value
Initial Topology	Default value: **Create empty scenario**
Choose Network Scale	**Enterprise** (“Use metric units” enabled)
Enterprise Sizing Method	Specify size
Specify Size	**10 km x 10 km**
Select Technologies	None
Review	Check values, then click **Finish**

4 In the object palette, click on **Configure Palette...**, then clear the palette and click on the **Node Models** button to add the **<initials>_mrt_jam**, **<initials>_mrt_rx**, and **<initials>_mrt_tx** node models to the palette. Save the palette as **<initials>_mrt_palette**.

5 Click **OK** to close the Configure Palette dialog box, then build the network shown below.

Network Topology



6 For each node:

6.1 Right-click on the node and choose **Advanced Edit Attribute** to view the advanced attributes dialog box.

6.2 Assign the **name** attribute as shown here.

6.3 Edit the **x position** and **y position** attributes to position each node.

The relative position of nodes plays an important role in the behavior of wireless communications. To get the expected results, make sure you place the nodes exactly as specified:

- **tx** node: **(3, 3)**
- **rx** node: **1 kilometer** to the right of the tx node, **(4, 3)**
- **jam** node: **(0.5, 2.5)**.

7 Close the object palette.

To specify node movement, the network model uses an attribute called **trajectory**. The value of this attribute is the name of an ASCII text file that is created in the Project Editor. The file contains data specifying times and locations that the mobile node will pass through as the simulation progresses.

Now that the network model has been defined, you must specify a trajectory for the mobile jamming node to follow.

1 Choose **Topology** > **Define Trajectory…**.

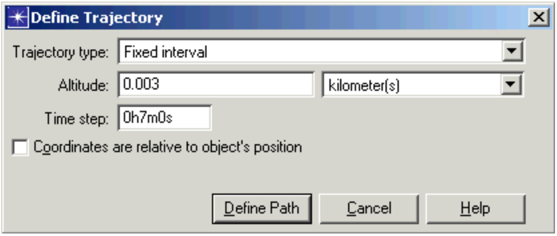
2 In the Define Trajectory dialog box, specify the attributes as shown, then click on the **Define Path** button:

— Trajectory type: **Fixed interval**

— Altitude: **0.003 kilometers**

— Time step: **0h7m0s**

Define Trajectory Dialog Box



When you clicked on the **Define Path** button in the Define Trajectory dialog box, the dialog box disappeared and your cursor changed to a line in the Project Editor.

You can now draw the mobile node's trajectory:

1 Left-click on the left edge of the **jam** node to begin the trajectory.

2 Left-click on the grid at the **(7.5, 2.5)** position (where 7.5 is the horizontal position and 2.5 is the vertical position).

3 Right-click to end the trajectory.

4 Name the trajectory **<initials>_mrt** in the Save Trajectory dialog box and click **OK**.

➥ The trajectory disappears from the screen because it has not yet been referenced by a mobile node.

You need to assign this trajectory to the jammer in the following steps.

Finally, assign the trajectory you just created to the **jam** node.

1 Right-click on the **jam** node and select **Edit Attributes**.

2 Change the **trajectory** attribute to **mrt**.

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

➥ The trajectory is visible as a green arrow in the Project Editor.

4 Right-click on the green line and select **Edit Trajectory**.

➥ The **Edit Trajectory Information** dialog box appears.

5 Edit the X and Y values for each row, as follows. Make sure the **Coordinates are relative to object's position** box is NOT checked.

#	X Position	Y Position
1	0.5	2.5
2	7.5	2.5

6 Click **OK** to exit the dialog box and to overwrite the existing file.

7 Save the project with the default name.

Collecting Statistics and Running Simulations

For this model, you are interested in the effect different antenna patterns have on the receiving node in a network. Instead of changing the antenna pattern attribute (which controls the antenna pattern used) at the node level for each simulation, you can configure the Simulation Sequence Editor to vary this attribute automatically for parametric studies.

You can gather the radio receiver channel statistics for this simulation in the Project Editor. These statistics include the **bit error rate** (**BER**) and **throughput** in packets/sec. The packet throughput statistic indicates the average number of packets the receiver channel successfully received per second. New samples of this statistic are only generated for packets with BER lower than the receiver ECC threshold as specified at the node level in the radio receiver module’s ecc threshold attribute.

Because the radio receiver module used in this tutorial has a value of 0.0 errors/bit for this attribute, only packets that have no bit errors will be accepted.

You can change the **collection mode** for different statistics. These modes specify the way in which statistics are captured (all values, bucket, sample, glitch removal), as well as their collection mode.

To collect the bit error rate and throughput statistics:

1 Right-click on the **rx** node object and select **Choose Individual DES Statistics** from the pop-up menu.

2 Select the following statistics:

— **Module Statistics > radio_rx.channel [0] > radio receiver > bit error rate**

— **Module Statistics > radio_rx.channel [0] > radio receiver > throughput (packets/sec)**

3 Right-click on the **bit error rate** statistic and choose **Change Collection Mode** from the pop-up menu.

4 Select the **Advanced** checkbox in the Capture Mode dialog box.

5 Change the **Capture mode** to **glitch removal**. Click **OK** when done.

To set the collection mode for the throughput statistic:

1 Right-click on the **throughput (packets/sec)** statistic and choose **Change Collection Mode** from the pop-up menu.

2 Select the **Advanced** checkbox in the Capture Mode dialog box

3 Verify that **Capture mode** is set to **bucket**, and that the **Bucket mode** is set to **sum/time**.

4 Change the **Sample Frequency** to **10 seconds** by clicking on the **Every...seconds** radio button and editing the value.

5 Make sure the **Reset** checkbox is not selected.

6 Click **OK** to close the Capture Mode dialog box, then click **OK** again to close the Choose Results dialog box.

Now that you have specified the statistics to collect, you can configure the simulation. You can use the Simulation Sequence Editor to conduct a parametric study—one in which the value of an attribute is varied to determine the effect on network behavior.

1 Choose **DES** > **Configure/Run Discrete Event Simulation (Advanced)**.

➥ The Simulation Sequence Editor opens.

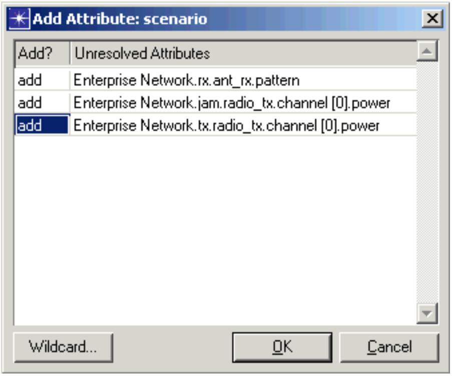
2 Right-click on the simulation set and select **Edit Attributes** from the pop-up menu.

Configure the simulation as follows:

1 Click on the **Inputs** tree node, and then the **Object Attributes** node, then click on the **Add...** button to show the Add Attributes dialog box.

2 Click in the **Add** column for all three of the unresolved attributes, then click **OK** when you have selected them.

Add Attribute Dialog Box



These are the attributes that you promoted in the Node Editor. Because you did not assign values when you promoted the attributes, you must assign them now. Note that the attributes now appear in the Attributes table, but they lack values.

Add the values for the **ant_rx.pattern** attribute (if necessary, drag the column divider to expand the Attribute column and show the full attribute names):

1 Click on the **ant_rx.pattern** attribute so that it is selected.

2 Click on the **Set Multiple Values…** button.

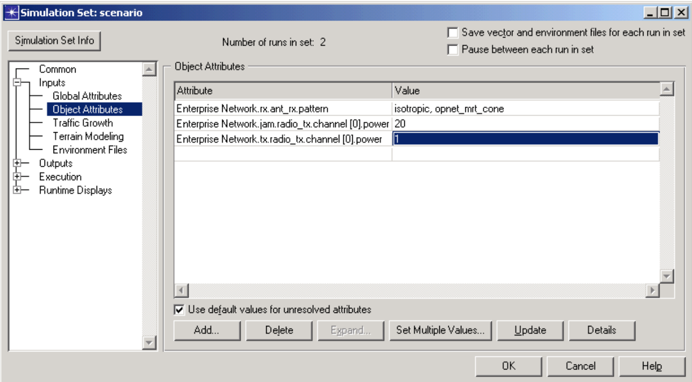
3 In the attribute dialog box, click in the **Value** cell and select **isotropic**. Move down to the next row, click again, and select **<initials>_mrt_cone**. Click **OK**.

Add the values for the **jam.radio_tx.channel [0].power** and **tx.radio_tx.channel [0].power** attributes:

1 Set the **jam.radio_tx.channel [0].power** attribute to **20**. Press **<Return>** when finished.

2 Set the **tx.radio_tx.channel [0].power** attribute to **1**. Press **<Return>** when finished.

Adding Values for the Promoted Attributes



Note that the **Number of runs in set** is now **2**. This is because the **ant_rx.pattern** attribute now has two possible values, so that two separate simulations will run, using a different value for this attribute for each simulation.

When you run multiple simulations, you must specify that you want to save the results for each simulation:

1 Select the **Save vector and environment files for each run in set** checkbox.

You also need to change the Seed and Duration settings for this simulation:

1 Click on the **Common** tree node.

2 Change **Duration** to **420 seconds**.

3 Change **Seed** to **50**.

4 When you are finished making changes in the Simulation Set dialog box, click **OK**.

➥ The simulation set icon changes to indicate that there are multiple runs in the simulation set.

5 Save the simulation set.

6 Verify that the **repositories** preference is empty.

6.1 Choose **Edit** > **Preferences**.

6.2 Type **repositories** and click **Find**.

6.3 Verify that the Value cell shows **()**. Delete any other entry.

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

7 Run the simulation by clicking on the **Execute Simulation Sequence** toolbar button.

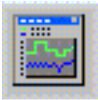
8 When both simulations are complete, close the Simulation Sequence Editor. If you had problems, see *"Troubleshooting Modeler Tutorials"*.

Viewing and Interpreting Results

Now that the simulations have been run, you can check the bit error rate and packet throughput results.

1. Choose **File** > **New...**, then select **Analysis Configuration** and click **OK**.

2. Click on the **Create a Graph of a Statistic** toolbar button.

   ➥ Two sets of results appear:
   <initials>_mrt_net-antenna_test-1
   <initials>_mrt_net-antenna_test-2.

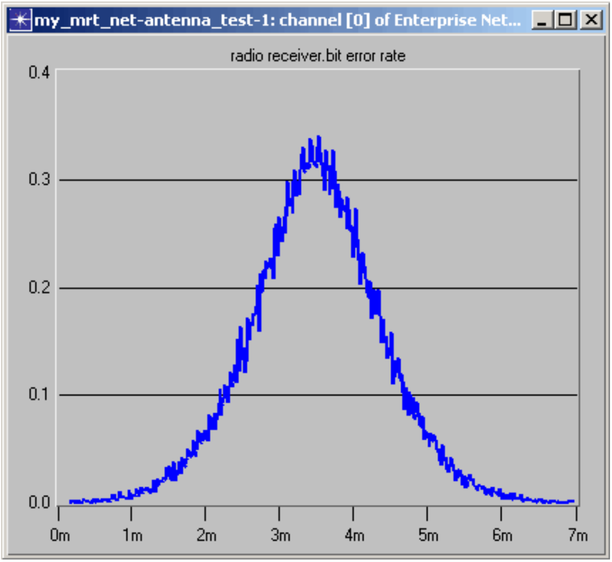
   These results correspond to the two simulations—one for the isotropic antenna pattern and one for the cone antenna pattern.

3. Double-click on the arrow or plus icon next to **<initials>_mrt_net-antenna_test-1** to view the full hierarchy of available statistics for that simulation run.

4. Click in the box next to **bit error rate**, then click the **Show** button. Move the graph off to the side.

   ➥ The graph shows the bit error rate of the isotropic antenna.

Bit Error Rate of the Isotropic Antenna



As expected, the graph for the isotropic antenna pattern shows that the **bit error rate** at the receiver node gradually increases as the distance between the jammer and receiver nodes decreases.

The bit error rate reaches a maximum of about 0.32 errors/bit when the distance between the jammer and the receiver is smallest. The isotropic receiver antenna receives jammer interference during the entire simulation.

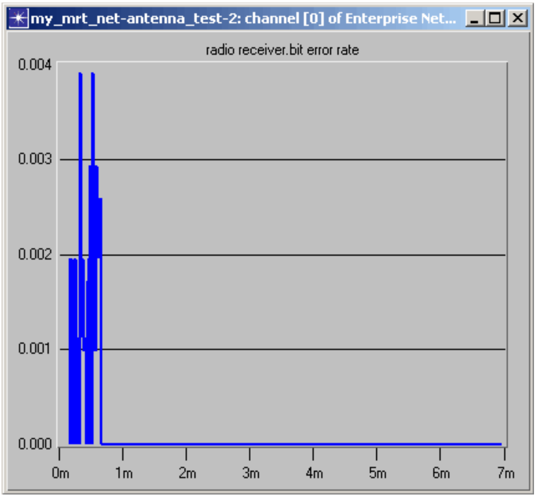
5 Uncheck **bit error rate** for **<initials>_mrt_net-antenna_test-1**.

6 Double-click on the arrow or plus icon next to **<initials>_mrt_net-antenna_test-2**, then click in the box next to **bit error rate** and click the **Show** button.

The results for the directional antenna that follow are highly dependent on the antenna gain. If your results do not match those shown here, it is probably due to small variations in the defined gain.

The graph shows the bit error rate of the directional antenna.

Bit Error Rate of the Directional Antenna



The graph from the directional antenna also reveals that the bit error rate at the receiver node increases initially as the distance between the jammer node and receiver node decreases.

However, after about one minute, the direction vector between the jammer antenna and the receiver antenna was no longer in line with the direction of greatest gain for the receiver antenna. Therefore, the receiver node stopped receiving interference from the jammer node and the bit error rate at the receiver dropped to 0. This drop dramatically increased the number of packets received from the stationary transmitter node (as will be seen in the next graphs).

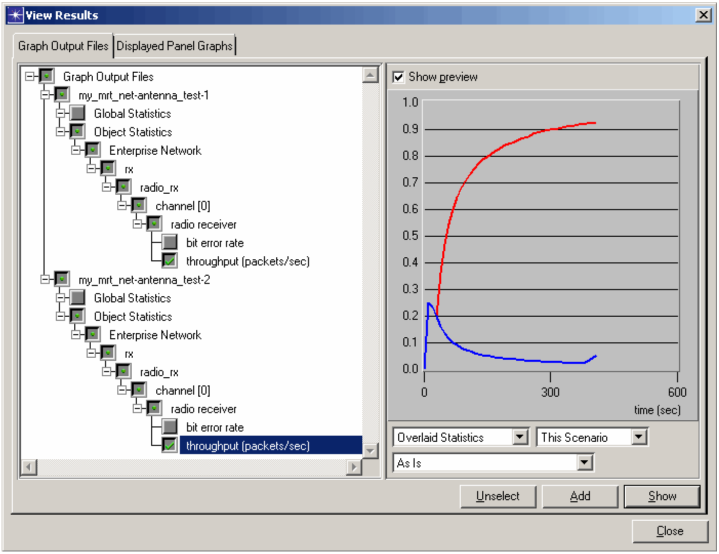
To view the results for packet throughput:

1. Close (delete) both graphs for bit error rate, and unselect the **bit error rate** statistic in the View Results dialog box.

2. Select the **throughput (packets/sec)** statistic for both **<initials>_mrt_net_antenna_test-1** and **<initials>_mrt_net_antenna_test-2**.

3. Select **Overlaid Statistics** to display both graphs on one panel, then click **Show**.

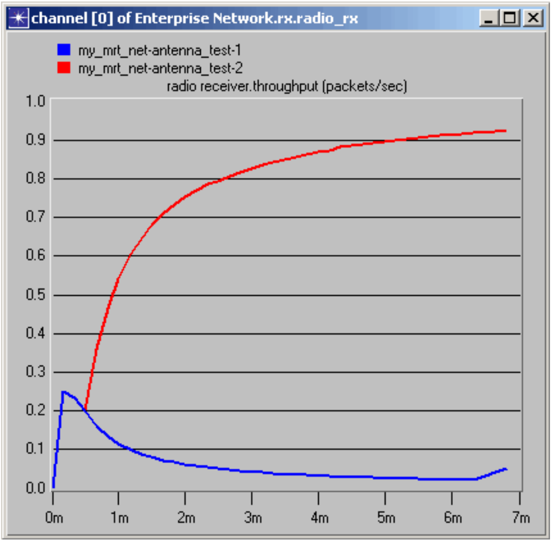
Selecting Statistics Overlaid



➥ The graph of the throughput statistic is shown in the following figure.

The throughput in packets/second on your graph should look similar to the following figure. Your graph may not match exactly because the trajectory path you drew may vary slightly from the path that produced this graph.

Throughput for Both Antennas



Note that for the isotropic antenna pattern, the average number of received packets declined during the simulation (this trend reversed somewhat at the end of the simulation as the distance between the jammer and the receiver increased). For the directional antenna, packet throughput was poor whenever the direction vector connecting the jammer and receiver antennas was in line with the receiver antenna's direction of greatest gain. However, after about thirty seconds—when the jammer was no longer in the direction of greatest gain for the receiver—the number of received packets began to increase.

Congratulations! You have completed this tutorial.

If you installed other modules, return to the main tutorial menu and continue with the tutorials.