QualNet®
Building Smarter Networks

QualNet 6.1
TIREM Advanced Propagation Model Library

September 2012

Scalable Network Technologies, Inc.

6100 Center Drive, Suite 1250
Los Angeles, CA 90045

Phone: 310-338-3318
Fax: 310-338-7213

http://www.scalable-networks.com

Scalable Network Technologies, Inc.
6100 Center Drive, Suite 1250
Los Angeles, CA 90045
Phone: 310-338-3318
Fax: 310-338-7213

http://www.scalable-networks.com

Table of Contents

1 Overview of Model Library

1.1 List of Models in the Library

The models described in the TIREM Advanced Propagation Model Library are listed in Table 1-1.

TABLE 1-1. TIREM Advanced Propagation Library Models

Model Name	Model Type	Section Number
TIREM	Propagation	Section 2.1

1.2 Supported Platforms

Although the QualNet TIREM interface is supported on all QualNet platforms (refer to *QualNet Installation Guide* and *QualNet Distributed Reference Guide* for details of QualNet platforms), the TIREM model is not supported on all platforms. Because of this, the use of the TIREM interface is limited to the platforms supported by the TIREM model.

Notes: **1.** Although the QualNet TIREM interface has been tested only on 32-bit Windows platforms, it should work on all QualNet platforms on which the TIREM model is supported.

2. Since the TIREM model is not thread-safe, running scenarios using the QualNet TIREM interface on shared memory multi-processor platforms may produce unrepeatable and /or incorrect results. However, these scenarios can be safely run on distributed architectures.

1.3 Installation and Compilation

To use the QualNet TIREM interface, you must install TIREM 3.15 (or a later version) which is available from the Defense Information Systems Agency, Joint Spectrum Center.

The TIREM interface source code is included in the QualNet distribution. However, in order to use the TIREM interface, QualNet needs to be compiled after TIREM 3.15 (or a later version) is installed. Refer to *QualNet Programmer's Guide* for detailed instructions for compiling QualNet.

This section describes how to activate the TIREM Advanced Propagation Model Library on Windows (see Section 1.3.1) and Linux (see Section 1.3.2) platforms.

1.3.1 Activation on Windows

To activate the TIREM Advanced Propagation Model Library on Windows, perform the following steps:

1. Open a command window and change the directory to QUALNET_HOME/main.
2. Open the file QUALNET_HOME/main/Makefile-addons-windows with a text editor.
   Change the line

   ```
   #include ../libraries/tirem/Makefile-windows
   ```

   to

   ```
   include ../libraries/tirem/Makefile-windows
   ```

3. Copy the makefile for your compiler to Makefile. For example, if you are using VC9 on a 32-bit platform, use the following command to make a copy of the makefile:

   ```
   copy Makefile-windows-vc9 Makefile
   ```

   Refer to *QualNet Programmer's Guide* for the list of makefiles for different compilers.

4. Use the following commands to remove all object files and recompile:

   ```
   nmake clean
   nmake
   ```

1.3.2 Activation on Linux

To activate the TIREM Advanced Propagation Model Library on Linux systems, perform the following steps:

1. Open a command window and change the directory to QUALNET_HOME/main.
2. To automatically select the makefile for your system, type the following command:

   ```
   ./configure.sh
   ```

   The script will prompt you for the model libraries that should be compiled with QualNet. Type "Yes" when the script prompts you for including the TIREM Advanced Propagation Model Library. The script will create a file called Makefile in the folder QUALNET_HOME/main.

   Note: Once you have created the makefile using this script, you do not need to run the script again unless you want to change the list of libraries to be compiled with QualNet.

3. Alternatively, manually select and edit the makefiles as follows:
   a. Open the file QUALNET_HOME/main/Makefile-addons-unix with a text editor.
   Change the line

```
#include ../libraries/tirem/Makefile-unix
```

   to

```
include ../libraries/tirem/Makefile-unix
```

   b. Copy the makefile for your compiler to Makefile. For example, for Red Hat Enterprise Linux 5.3 and other Linux distributions with glibc 2.5 and gcc 4.1 on a 32-bit platform, use the following command to make a copy of the makefile:

```
cp Makefile-linux-glibc-2.5-gcc-4.1 Makefile
```

   Refer to *QualNet Programmer's Guide* for the list of makefiles for different compilers.

4. Use the following commands to remove all object files and recompile:

```
make clean
make
```

1.4 Conventions Used

1.4.1 Format for Command Line Configuration

This section describes the general format for specifying parameters in input files, the precedence rules for parameters, and the conventions used in the description of command line configuration for each model.

1.4.1.1 General Format of Parameter Declaration

The general format for specifying a parameter in an input file is:

```
[<Qualifier>] <Parameter Name> [<Index>] <Parameter Value>
```

where

`<Qualifier>` The qualifier is optional and defines the scope of the parameter declaration. The scope can be one of the following: Global, Node, Subnet, and Interface. Multiple instances of a parameter with different qualifiers can be included in an input file. Precedence rules (see Section 1.4.1.2) determine the parameter value for a node or interface.

Global: The parameter declaration is applicable to the entire scenario (to all nodes and interfaces), subject to precedence rules. The scope of a parameter declaration is global if the qualifier is not included in the declaration.

Example:

```
MAC-PROTOCOL          MACDOT11
```

Node: The parameter declaration is applicable to specified nodes, subject to precedence rules. The qualifier for a node-level declaration is a list of space-separated node IDs or a range of node IDs (specified by using the keyword `thru`) enclosed in square brackets.

Example:

```
[5 thru 10] MAC-PROTOCOL          MACDOT11
```

Subnet: The parameter declaration is applicable to all interfaces in specified subnets, subject to precedence rules. The qualifier for a subnet-level declaration is a space-separated list of subnet addresses enclosed in square brackets. A subnet address can be specified in the IP dot notation or in the QualNet N syntax.

Example:

```
[N8-1.0 N2-1.0] MAC-PROTOCOL      MACDOT11
```

Interface: The parameter declaration is applicable to specified interfaces. The qualifier for an interface-level declaration is a space-separated list of subnet addresses enclosed in square brackets.

Example:

```
[192.168.2.1 192.168.2.4] MAC-PROTOCOL MACDOT11
```

`<Parameter Name>`	Name of the parameter.
`<Index>`	Instance of the parameter to which this parameter declaration is applicable, enclosed in square brackets. This should be in the range 0 to *n* -1, where *n* is the number of instances of the parameter. The instance specification is optional in a parameter declaration. If an instance is not included, then the parameter declaration is applicable to all instances of the parameter, unless otherwise specified.
`<Parameter Value>`	Value of the parameter.

Note: There should not be any spaces between the parameter name and the index.

Examples of parameter declarations in input files are:

```
PHY-MODEL                                          PHY802.11b
[1] PHY-MODEL                                      PHY802.11a
[N8-1.0] PHY-RX-MODEL                              BER-BASED
[8 thru 10] ROUTING-PROTOCOL                       RIP
[192.168.2.1 192.168.2.4] MAC-PROTOCOL             GENERICMAC
NODE-POSITION-FILE                                 ./default.nodes
PROPAGATION-CHANNEL-FREQUENCY[0]                   2.4e9
[1 2] QUEUE-WEIGHT[1]                              0.3
```

Note In the rest of this document, we will not use the qualifier or the index in a parameter's description. Users should use a qualifier and/or index to restrict the scope of a parameter, as appropriate.

1.4.1.2 Precedence Rules

Parameters without Instances

If the parameter declarations do not include instances, then the following rules of precedence apply when determining the parameter values for specific nodes and interfaces:

Interface > Subnet > Node > Global

This can be interpreted as follows:

- The value specified for an interface takes precedence over the value specified for a subnet, if any.
- The value specified for a subnet takes precedence over the value specified for a node, if any.
- The value specified for a node takes precedence over the value specified for the scenario (global value), if any.

Parameters with Instances

If the parameter declarations are a combination of declarations with and without instances, then the following precedence rules apply (unless otherwise stated):

Interface[i] > Subnet[i] > Node[i] > Global[i] > Interface > Subnet > Node > Global

This can be interpreted as follows:

- Values specified for a specific instance (at the interface, subnet, node, or global level) take precedence over values specified without the instance.

- For values specified for the same instance at different levels, the following precedence rules apply:
  - The value specified for an interface takes precedence over the value specified for a subnet, if any, if both declarations are for the same instance.
  - The value specified for a subnet takes precedence over the value specified for a node, if any, if both declarations are for the same instance.
  - The value specified for a node takes precedence over the value specified for the scenario (global value), if any, if both declarations are for the same instance.

1.4.1.3 Parameter Description Format

In the Model Library, most parameters are described using a tabular format described below. The parameter description tables have three columns labeled “Parameter”, “Values”, and “Description”. Table 1-2 shows the format of parameter tables. Table 1-4 shows examples of parameter descriptions in this format.

TABLE 1-2. Parameter Table Format

Parameter	Values	Description
<Parameter Name>	*<Type>*	*<Description>*
<Designation>	[<Range>]	
<Scope>	[<Default Value>]	
[<Instances>]	[<Unit>]	

Parameter Column

The first column contains the following entries:

- ***<Parameter Name>*:** The first entry is the parameter name (this is the exact name of the parameter to be used in the input files).
- ***<Designation>:*** This entry can be *Optional* or *Required*. These terms are explained below.
  - ***Optional***: This indicates that the parameter is optional and may be omitted from the configuration file. (If applicable, the default value for this parameter is included in the second column.)
  - ***Required***: This indicates that the parameter is mandatory and must be included in the configuration file.
- ***<Scope>:*** This entry specifies the possible scope of the parameter, i.e., if the parameter can be specified at the global, node, subnet, or interface levels. Any combination of these levels is possible.If the parameter can be specified at all four levels, the keyword “All” is used to indicate that.

  Examples of scope specification are:

  Scope: All

  Scope: Subnet, Interface

  Scope: Global, Node
- ***<Instances>:*** If the parameter can have multiple instances, this entry indicates the type of index. If the parameter can not have multiple instances, then this entry is omitted.

Examples of instance specification are:

Instances: channel number

Instances: interface index

Instances: queue index

Values Column

The second column contains the following information:

- **<*Type*>:** The first entry is the parameter type and can be one of the following: Integer, Real, String, Time, Filename, IP Address, Coordinates, Node-list, or List. If the type is a List, then all possible values in the list are enumerated below the word "List". (In some cases, the values are listed in a separate table and a reference to that table is included in place of the enumeration.)

  Table 1-3 shows the values a parameter can take for each type.

TABLE 1-3. Parameter Types

Type	Description
Integer	Integer value Examples: `2`, `10`
Real	Real value Examples: `15.0`, `-23.5`, `2.0e9`
String	String value Examples: `TEST`, `SWITCH1`
Time	Time value expressed in QualNet time syntax (refer to *QualNet User's Guide*) Examples: `1.5S`, `200MS`, `10US`

Filename	Name of a file in QualNet filename syntax (refer to *QualNet User's Guide*)
	Examples:
	`../../data/terrain/los-angeles-w`
	(For Windows and UNIX)
	`C:\snt\qualnet\6.1\scenarios\WF\WF.nodes`
	(For Windows)
	`/root/snt/qualnet/6.1/scenarios/WF/WF.nodes`
	(For UNIX)

Path	Path to a directory in QualNet path syntax (refer to *QualNet User's Guide*)
	Examples:
	`../../data/terrain` (For Windows and UNIX)
	`C:\snt\qualnet\6.1\scenarios\default`
	(For Windows)
	`/root/snt/qualnet/6.1/scenarios/default`
	(For UNIX)

IP Address	IPv4 or IPv6 address Examples: `192.168.2.1`, `2000:0:0:0::1`

TABLE 1-3. Parameter Types (Continued)

Type	Description
IPv4 Address	IPv4 address Examples: `192.168.2.1`
IPv6 Address	IPv6 address Examples: `2000:0:0:0::1`
Coordinates	Coordinates in Cartesian or Lat-Lon-Alt system. The altitude is optional. Examples: `(100, 200, 2.5)`, `(-25.3478, 25.28976)`
Node-list	List of node IDs separated by commas and enclosed in "{" and "}". Examples: `{2, 5, 10}`, `{1, 3 thru 6}`
List	One of the enumerated values. Example: See the parameter `MOBILITY` in Table 1-4.

Note: If the parameter type is List, then options for the parameter available in QualNet and the commonly used model libraries are enumerated. Additional options for the parameter may be available if some other model libraries or addons are installed. These additional options are not listed in this document but are described in the corresponding model library or addon documentation.

- ***<Range>*:** This is an optional entry and is used if the range of values that a parameter can take is restricted. The permissible range is listed after the label "*Range:*" The range can be specified by giving the minimum value, the maximum value, or both. If the range of values is not restricted, then this entry is omitted.

  If both the minimum and maximum values are specified, then the following convention is used to indicate whether the minimum and maximum values are included in the range:

`(min, max)`	`min` < parameter value < `max`
`[min, max)`	`min` ≤ parameter value < `max`
`(min, max]`	`min` < parameter value ≤ `max`
`[min, max]`	`min` ≤ parameter value ≤ `max`

  `min` (or `max`) can be a parameter name, in which case it denotes the value of that parameter.

  Examples of range specification are:

  Range: ≥ `0`

  Range: `(0.0, 1.0]`

  Range: `[1, MAX-COUNT]`

  Range: `[1S, 200S]`

Note: If an upper limit is not specified in the range, then the maximum value that the parameter can take is the largest value of the type (integer, real, time) that can be stored in the system.

- ***<Default>*:** This is an optional entry which specifies the default value of an optional or conditional-optional parameter. The default value is listed after the label "*Default:*"
- ***<Unit>*:** This is an optional entry which specifies the unit for the parameter, if applicable. The unit is listed after the label "*Unit:*". Examples of units are: meters, dBm, slots.

Description Column

The third column contains a description of the parameter. The significance of different parameter values is explained here, where applicable. In some cases, references to notes, other tables, sections in the User's Guide, or to other model libraries may be included here.

Table 1-4 shows examples of parameter descriptions using the format described above.

TABLE 1-4. Example Parameter Table

Parameter	Values	Description
MOBILITY Optional *Scope:* Global, Node	List: • NONE • FILE • GROUP-MOBILITY • RANDOM-WAYPOINT Default: NONE	Mobility model used for the node. If MOBILITY is set to NONE, then the nodes remain fixed in one place for the duration of the simulation. See Table 7-11 for a description of mobility models.
BACKOFF-LIMIT Required *Scope:* Subnet, Interface	Integer *Range:* [4,10) *Unit:* slots	Upper limit of backoff interval after collision. A backoff interval is randomly chosen between 1 and this number following a collision.
IP-QUEUE-PRIORITY-QUEUE-SIZE Required *Scope:* All *Instances:* queue index	Integer *Range:* [1, 65535] *Unit:* bytes	Size of the output priority queue.
MAC-DOT11-DIRECTIONAL-ANTENNA-MODE Optional *Scope:* All	List • YES • NO *Default:* NO	Indicates whether the radio is to use a directional antenna for transmission and reception.

1.4.2 Format for GUI Configuration

The GUI configuration section for a model outlines the steps to configure the model using the GUI. The following conventions are used in the GUI configuration sections:

Path to a Parameter Group

As a shorthand, the location of a parameter group in a properties editor is represented as a path consisting of the name of the properties editor, name of the tab within the properties editor, name of the parameter group within the tab (if applicable), name of the parameter sub-group (if applicable), and so on.

Example

The following statement:

Go to **Default Device Properties Editor > Interfaces > Interface # > MAC Layer**

is equivalent to the following sequence of steps:

1. Open the Default Device Properties Editor for the node.
2. Click the **Interfaces** tab.
3. Expand the applicable Interface group.
4. Click the **MAC Layer** parameter group.

The above path is shown in Figure 1-1.

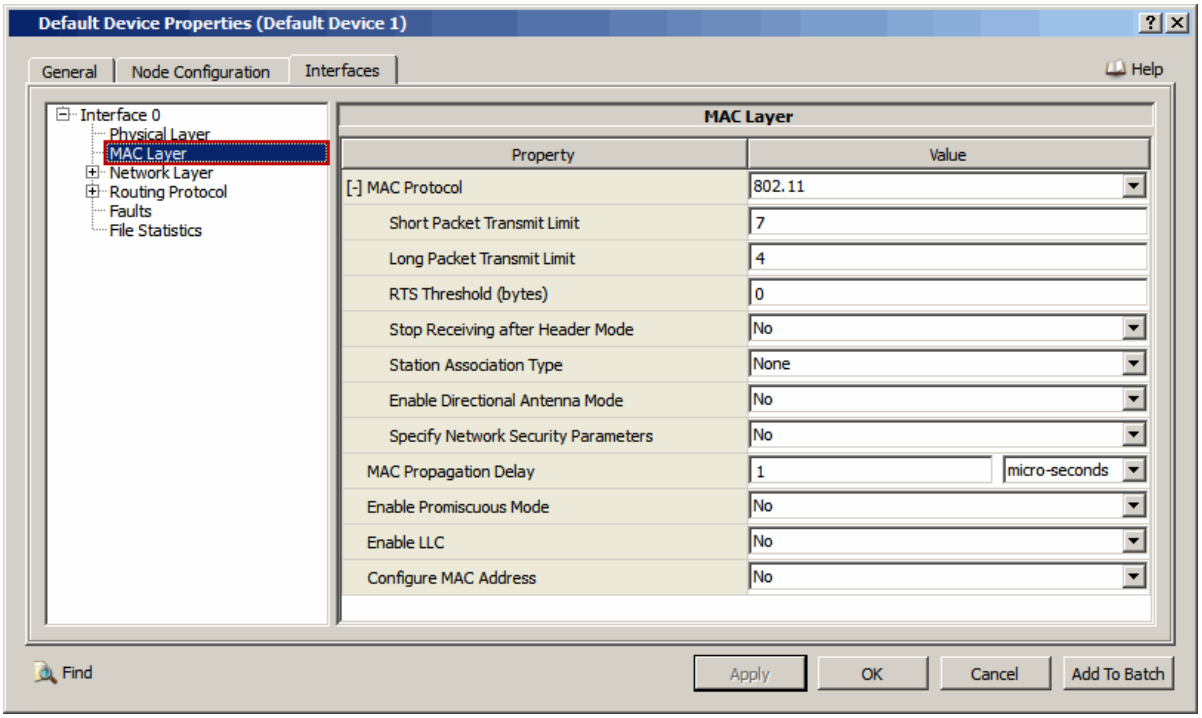

FIGURE 1-1. Path to a Parameter Group

Path to a Specific Parameter

As a shorthand, the location of a specific parameter within a parameter group is represented as a path consisting of all ancestor parameters and their corresponding values starting from the top-level parameter. The value of an ancestor parameter is enclosed in square brackets after the parameter name.

Example

The following statement:

> Set **MAC Protocol** *[= 802.11]* **> Station Association Type** *[= Dynamic]* **> Set Access Point** *[= Yes]* **>** *Enable Power Save Mode* to *Yes*

is equivalent to the following sequence of steps:

1. Set **MAC Protocol** to *802.11*.
2. Set **Station Association Type** to *Dynamic.*
3. Set **Set Access Point** to *Yes*.
4. Set **Enable Power Save Mode** to *Yes*.

The above path is shown in Figure 1-2.

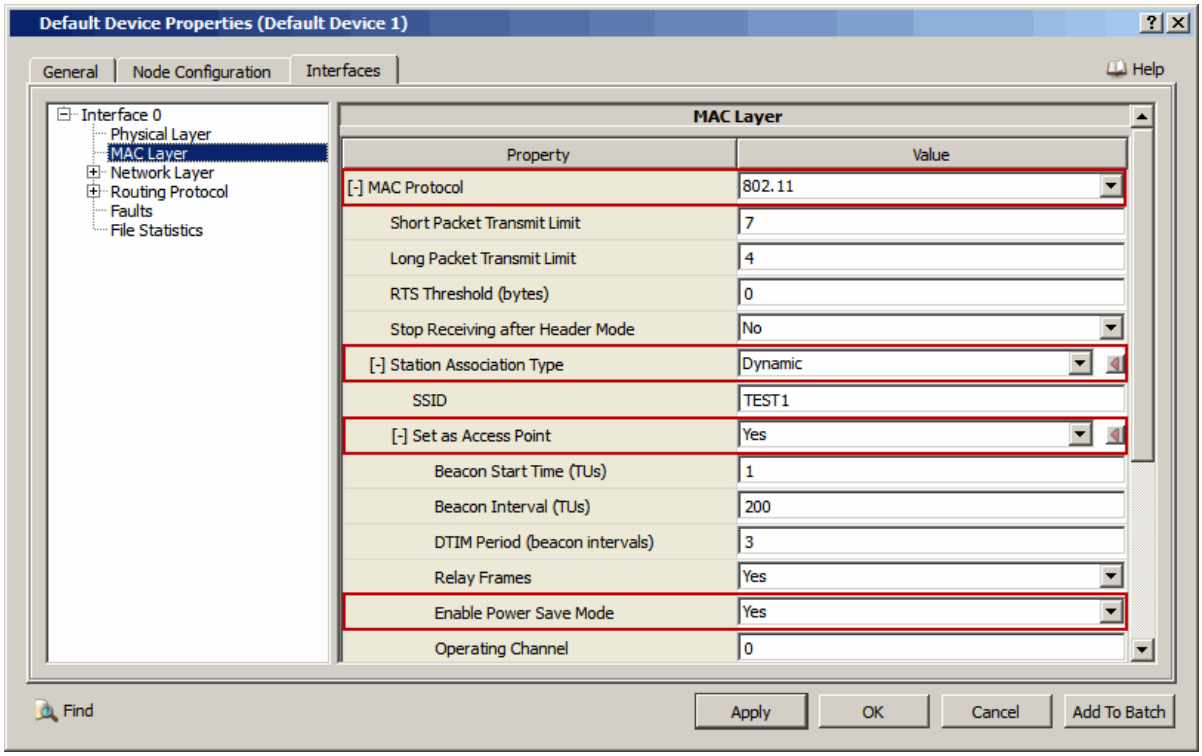

FIGURE 1-2. Path to a Specific Parameter

Parameter Table

GUI configuration of a model is described as a series of a steps. Each step describes how to configure one or more parameters. Since the GUI display name of a parameter may be different from the name in the configuration file, each step also includes a table that shows the mapping between the GUI names and command line names of parameters configured in that step. For a description of a GUI parameter, see the description of the equivalent command line parameter in the command line configuration section.

The format of a parameter mapping table is shown in Table 1-5.

TABLE 1-5. Mapping Table

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
<GUI Display Name>	*<Scope>*	*<Command Line Parameter Name>*

The first column, labeled “GUI Parameter”, lists the name of the parameter as it is displayed in the GUI.

The second column, labeled “Scope of GUI Parameter”, lists the level(s) at which the parameter can be configured. <*Scope*> can be any combination of: Global, Node, Subnet, Wired Subnet, Wireless Subnet, Point-to-point Link, and Interface.

Table 1-6 lists the Properties Editors where parameters with different scopes can be set.

Notes: **1.** Unless otherwise stated, the “Subnet” scope refers to “Wireless Subnet”.

2. The scope column can also refer to Properties Editors for special devices and network components (such as ATM Device Properties Editor) which are not included in Table 1-6.

TABLE 1-6. Properties Editors for Different Scopes

Scope of GUI Parameter	Properties Editor
Global	Scenario Properties Editor
Node	Default Device Properties Editor (General and Node Configuration tabs)
Subnet Wireless Subnet	Wireless Subnet Properties Editor
Wired Subnet	Wired Subnet Properties Editor
Point-to-point Link	Point-to-point Link Properties Editor
Interface	Interface Properties Editor, Default Device Properties Editor (Interfaces tab)

The third column, labeled “Command Line Parameter”, lists the equivalent command line parameter.

Note: For some parameters, the scope may be different in command line and GUI configurations (a parameter may be configurable at fewer levels in the GUI than in the command line).

Table 1-7 is an example of a parameter mapping table.

TABLE 1-7. Example Mapping Table

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Define Area	Node	`OSPFv2-DEFINE-AREA`
OSPFv2 Configuration File	Node	`OSPFv2-CONFIG-FILE`
Specify Autonomous System	Node	N/A

TABLE 1-7. Example Mapping Table (Continued)

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Configure as Autonomous System Boundary Router	Node	AS-BOUNDARY-ROUTER
Inject External Route	Node	N/A
Enable Stagger Start	Node	OSPFv2-STAGGER-START

2 Propagation Models

This chapter describes features, configuration requirements and parameters, statistics, and scenarios for Propagation Models, and consists of the following section:

- Terrain Integrated Rough Earth Model (TIREM)

2.1 Terrain Integrated Rough Earth Model (TIREM)

The QualNet TIREM Advanced Propagation model is based on the available references.

2.1.1 Description

TIREM is a propagation model which predicts the pathloss along the propagation path over irregular terrain at frequencies between 1 MHz and 40 GHz. Based on the geometry of the terrain profile, the appropriate propagation model is used to calculate the pathloss.

QualNet provides an interface only to the TIREM model. Customers must acquire the TIREM library from the Defense Information Systems Agency, Joint Spectrum Center. QualNet supports terrain data for both Cartesian and latitude/longitude coordinate systems. DTED and DEM, which both require latitude/longitude coordinates, are the most commonly used.

2.1.2 Command Line Configuration

To specify TIREM as the propagation pathloss model, include the following parameter in the scenario configuration (.config) file:

```
[<Qualifier>]   PROPAGATION-PATHLOSS-MODEL     TIREM
```

The scope of this parameter declaration can be Global or Subnet. See Section 1.4.1.1 for a description of `<Qualifier>` for each scope.

TIREM Parameters

Table 2-1 lists the configuration parameters for TIREM interface. See Section 1.4.1.3 for a description of the format used for the parameter table.

TABLE 2-1. TIREM Parameters

Parameter	Value	Description
`PROPAGATION-SAMPLING-DISTANCE` *Optional* *Scope:* All	Real *Range:* (see note) *Unit:* meters *Default:* `100`	Specifies the sample distance (in meters) along the propagation path. For a given terrain profile along propagation path, the shorter the sample distance, the more sample points for the terrain profile. This should be set to the resolution of the terrain data, for example, it should be 30 meters for DTED level1. **Note:** As the general rule, the range should be larger than the resolution of the terrain data and less than the half of terrain dimensions for the scenarios.

TABLE 2-1. TIREM Parameters (Continued)

Parameter	Value	Description
PROPAGATION-REFRACTIVITY *Optional* *Scope:* All	Real *Range:* [200.0, 450.0] *Default:* 360	Specifies the refractivity value of the terrain. The recommended values for different terrain types are: Equatorial 360 Continental Subtropical 320 Maritime Tropical 370 Desert 280 Continental Temperate 301 Maritime Temperate, Over Land 320 Maritime Temperate, Over Sea 350
PROPAGATION-CONDUCTIVITY *Optional* *Scope:* All	Integer *Range:* [0.00001, 100.0] *Default:* 0.005	Specifies the conductivity of the earth's surface (in siemens/meter). The recommended values for different terrain types are: Average Ground 0.005 Poor Ground 0.001 Good Ground 0.02 Fresh Water 0.01 Salt Water 5.0
PROPAGATION-PERMITTIVITY *Optional* *Scope:* All	Real *Range:* [1.0 to 100.0] *Default:* 15.0	Specifies the relative permittivity of the earth's surface. The recommended values for different terrain types are: Average Ground 15 Poor Ground 4 Good Ground 25 Fresh Water 81 Salt Water 81
PROPAGATION-HUMIDITY *Optional* *Scope:* All	Real *Range:* [0.0, 110.0] *Default:* 10.0	Specifies the relative humidity of the environments.
PROPAGATION-CLIMATE *Optional* *Scope:* All	Integer *Range:* [1, 7] *Default:* 1	Climate specification. The integer values correspond to the following: 1 Equatorial 2 Continental Subtropical 3 Maritime Tropical 4 Desert 5 Continental Temperate 6 Maritime Temperate, Over Land 7 Maritime Temperate, Over Sea

TABLE 2-1. TIREM Parameters (Continued)

Parameter	Value	Description
ANTENNA-POLARIZATION *Optional* *Scope:* All	List: • HORIZONTAL • VERTICAL *Default:* VERTICAL	Antenna polarization.
TIREM-DLL-FILENAME *Required* *Scope:* All	Filename	Filename (including the path) for the TIREM library.

2.1.3 GUI Configuration

To configure the TIREM model in QualNet GUI, perform the following steps:

1. Go to **Scenario Properties Editor > Channel Properties**.
2. Set **Number of Channels** to the desired value as shown in Figure 2-1.

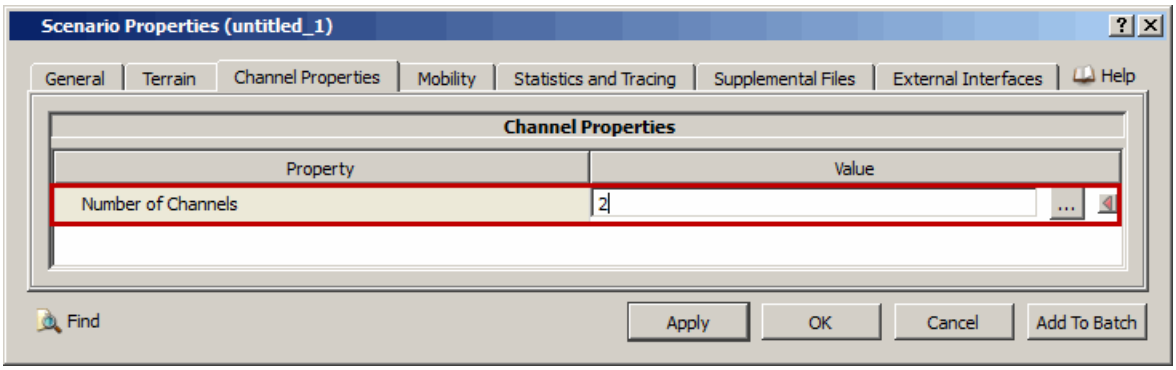

FIGURE 2-1. Setting Number of Channels

3. Click on the **Open Array Editor** [...] button in the **Value** column. This opens the Array Editor.

4. In the left panel of the Array Editor, select the index of the channel to be configured. In the right panel, set **Pathloss Model** to *TIREM* and set the dependent parameters listed in Table 2-2.

FIGURE 2-2. Setting TIREM Parameters

TABLE 2-2. Command Line Equivalent of TIRM Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
TIREM DLL	Global	TIREM-DLL-FILENAME
Sampling Distance	Global	PROPAGATION-SAMPLING-DISTANCE
Refractivity	Global	PROPAGATION-REFRACTIVITY
Conductivity	Global	PROPAGATION-CONDUCTIVITY
Permittivity	Global	PROPAGATION-PERMITTIVITY
Humidity	Global	PROPAGATION-HUMIDITY
Climate	Global	PROPAGATION-CLIMATE
Antenna Polarization	Global	ANTENNA-POLARIZATION

2.1.4 Statistics

There are no statistics generated for the TIREM Advanced Propagation model.