International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 13 (2014), pp. 1349-1354

http://www. irphouse.com

Qualnet Simulator

Sorout Dinesh[1] and Goyal Sonal[2]

[1]*Department of CSE, Shri Ram College of Engineering & Management, Palwal, INDIA*
[2]*Department of CSE, Shri Ram College of Engineering & Management Palwal, INDIA*

Abstract

QualNet is a state-of-the-art simulator for large, heterogeneous networks and the distributed applications that execute on such networks. The following QualNet features provide a unique capability for accurate, efficient simulation of large-scale, heterogeneous networks: Robust set of wired and wireless network protocol and device models, useful for simulating diverse types of networks. Optimized for speed and scalability on one processor, QualNet executes equivalent scenarios 5-10x times faster than commercial alternatives Designed from the ground-up as a parallel simulator, QualNet executes your simulation multiples faster as you add processors. A robust graphical user interface covers all aspects of the simulation, from scenario creation and topology setup, integration of custom protocols, through real-time execution of network models from within the GUI, animation, to post-simulation statistical analysis.

Keywords: Qualnet, Simulation, Command, Configuration, Channel

Introduction:-

The QualNet Developer IDE is a GUI program for developing network scenarios that comes with QualNet 5.0. It can be used to visually design network scenarios and then run simulations of these Networks. Although networks can be designed and simulated in a command-line fashion as well, we will focus on the Developer IDE package. The first step is to open the Developer IDE program. Depending on the installation procedure, it can be opened via the start menu or by opening the directory where QualNet was installed.

Command-Line Execution and File-Based Specification

For those users who prefer to execute simulations from the command-line, this chapter describes the syntax and files for configuring a simulation outside of the Graphical User Interface (GUI).

QualNet requires a single parameter, which is the name of the experiment.config file which sets up the experiment, or network scenario, that you wish to run. An included sample of this file is called the **default.config** file. The **default.config** file sometimes references support configuration files (**default.app**, for example, configures the ftp, telnet, CBR, or other application-level models in a QualNet simulation). We call these files, generally, **QualNet Configuration Files**, **config files**, or **default.config files** The naming convention we have employed uses the prefix ("default" in this case) to refer to the name of the experiment, and the suffix (.config, .app, .ber) to indicate the type of file. **default.config** doesn't have to be named **default.config**. It could be named my-scenario-Feb10.config, for example. The same freedom applies to **default.app**, and so on.

Directory Structure of QualNet

The QualNet source tree appears under the directory you chose during QualNet installation. The main subdirectories and a brief description of their contents and purpose are listed below:

/application code for the application layer protocols and traffic generators
/bin executable and configuration or input/output files
/data supplementary data files for experiments and utility programs
/doc the documentation
/gui the Visual Environment Toolset
/include common include files
/mac the code for the MAC layer protocols
/main the basic framework design
/mobility the code for the mobility models
/network the code for the network layer protocols and routing protocols
/phy the code for the physical layer models
/transport the code for the transport layer protocols (TCP/UDP, RSVP, etc)
/verification Sample files and outputs to verify protocol correctness

Command-Line Compilation and Execution

If you are running QualNet for the first time, or if you have created or modified protocol models in QualNet, and placed their source code in the relevant directories, you will need to compile QualNet from the main/ subdirectory to include them. Change to the main/ subdirectory using "cd", and then:

Windows NT®/2000®: run "make" from the command-line
Solaris™/Linux: run "make" from the command-line.

To run the simulation you should change directory to the bin/ subdirectory. The executable produced by the previous step is called "qualnet" (Solaris™/Linux) or "qualnet.exe" (Windows®). This executables takes one required command line parameter, which is the filename of the relevant Scenario Input File. Scenario Input Files describe the network being modeled, the length of time to simulate this network, and all other scenario specific parameters, and are described in Section 2.3. This executable also takes a second, optional, command line parameter as a single word name for the experiment that this simulation execution represents. This parameter overrides the "**EXPERIMENT-NAME"** configuration parameter if it also appears in the Scenario Input File, and serves the same purpose. Type "qualnet **default.config**" from a command prompt within the **bin/** subdirectory to run the program with the sample scenario file. A file called "default.stat" will be produced at the end of the simulation, and contains all the statistics generated by the current experiment. You can use a text editor to make modifications to the **default.config** file to vary the parameters for running the simulation.

Scenario Input File Syntax

This section explains the syntax for specifying network scenarios in the Scenario Input File. It also covers the network-specific and node-specific parameterization syntax that allows users to select different options for different device models, in order to specify the heterogeneous elements such as hosts, routers, and radios that make up the network you are modeling.

Configuration File Converter

The set of recognized and/or required parameters, as described in the remainder of this section and the following chapter, can change with newer releases of QualNet. In order to allow users to easily update their configuration files to operate with the updated releases of QualNet, we have provided a tool for converting old configuration files into the most recent format. This section describes the usage of this tool. The tool is executed from the command line as follows.

The converter takes as parameters the name of the configuration file to be converted, and a name for the resulting file. The two names can be the same. The converter also has two optional parameters. The "-c" flag tells the converter to print all the descriptive comments included in the default configuration file to the new file. This is the default behaviour. The "-nc" flag disables the comments, printing only the active variables. The converted file orders its variables according to the standard format, and will print all the qualified versions of a variable consecutively. For example, assume the user's file contains the following.

LINK N2-1.0 {1, 2}
[N2-1.0] LINK-BANDWIDTH 1000000
LINK N2-2.0 {2, 3}
[N2-2.0] LINK-BANDWIDTH 1000000
LINK N2-3.0 {3, 4}
[N2-3.0] LINK-BANDWIDTH 1000000

The converter will print these as follows:

LINK N2-1.0 {1, 2}
LINK N2-2.0 {2, 3}
LINK N2-3.0 {3, 4}

[N2-1.0] LINK-BANDWIDTH 1000000
[N2-2.0] LINK-BANDWIDTH 1000000
[N2-3.0] LINK-BANDWIDTH 1000000

Note that this conversion utility only updates the variables in the main configuration file. It does not update the format of any auxiliary files, such as the application configuration file, or BGP.

Simulation Statistics

Statistics File

If you click on the **Statistics File** button at the bottom of the window, you will see large amount of per node simulation data. Among the collected statistics we see, - The packets generated by the source nodes and received by the destination nodes are listed as unicast packets. In a connection not all the generated packets may be successfully delivered to the destination. A packet will be discarded by the source as a result of reaching the number of retransmissions' limit. Transmissions may not be successful because of destination node not sending a CTS packet or as a result of the bit errors in the received packet. We note that the bit errors will depend on the distance between source and destination nodes, fading and shadowing. The strength of the received signal which will decrease as the distance between the Source and destination nodes increases. As a result of packet discarding, the throughput of the Source and destination nodes in a connection may not be same.

If the distance between two nodes is higher than the node's transmission range, then, these Nodes are not aware of each other. They cannot communicate with each other and they will not know each other's IP addresses. Therefore, if two such nodes will exchange data traffic with each Other, the packets will be generated by the Application layer of the source node but they will not be passed to the transport and then to the network layers because the IP address of the destination Node is unknown.

In addition to unicast traffic packets, as may be seen there are other packets being transmitted in the network, RTS, CTS, ACK and Broadcasting packets for routing. For each successfully received unicast packet there will be a corresponding ACK path. The number of RTS and CTS packets will correspond to the total number of transmissions of a packet until it is successfully transmitted.

Graph Analyzer Window

If you click on the **Overview** button at the bottom of the window, then we can navigate through the Physical, MAC, Network, Transport, and Application layers shown on left hand window. We can access to each layer data by clicking on the button for that layer at the bottom of left hand window.

This results in opening of a menu for that layer in the left hand window. Then clicking on any of the menu items displays a submenu. Clicking on any submenu item displays a bar chart for the data collected for that item during the simulation.

Comments

In the input file anything following”#" is treated as a comment. The sample **default.config** file contains descriptive information about the parameters that can be selected, using this style of commenting. It also contains all of the available options for each parameter, all of which are commented, except for the parameter currently selected. In order to select a different option than the one already selected, just add a “#” symbol in front of the current selection, and remove it from the selection you wish to use instead.

Channel Properties

Each channel in QualNet is uniquely identified by its frequency. Each channel is assumed to be orthogonal to the others (no inter-channel interference). Users can instantiate channels by initializing a propagation model for each, starting with the mandatory “PROPAGATION-CHANNEL-FREQUENCY” parameter Users can simulate spectrum overlapping by using multiple channels.

Propagation Channel Frequency

The PROPAGATION-CHANNEL-FREQUENCY parameter instantiates and uniquely identifies each simulated channel. The unit is in hertz. A user can distinguish between instances of this parameter by using the Instance ID specification between “[]”, as in the example below. These example instantiates two channels, at the 2.4 GHz and 2.5 GHz frequencies.

Propagation Limit

The following parameter defines the lower under of signal power for consideration by the simulator as a global default, or specified per channel, sing the Instance ID specification, as in the example below. A signal with powers below “PROPAGATION-LIMIT” (in dBm) is not delivered. This value impacts the fidelity of the model, with lower values equating to higher precision and longer simulation execution times. Higher values can improve simulation time, but this value must be smaller than the “RADIO-RX-SENSITIVITY” + “RADIOANTENNA-GAIN” for all nodes in the model or the simulator may not deliver Signals that are within a particular receiver’s ability to accept.

This parameter is purely a simulation parameter, and does not determine the maximum distance between two radios that still allows successful transmission. And 8.2.8 for the parameters that affects radio reception. The following example shows a propagation limit of –111.0 dBm on channel 0, and –121.0 dBm on

Channel 1.

```
#   PROPAGATION-LIMIT [0]     -111.0
    PROPAGATION-LIMIT [1]     -121.0
#
```

Path Loss Model Selection

Path loss models compute signal attenuation according to tunable parameters and node coordinates. This parameter can be specified as a global default, or specified per channel, using the Instance ID specification as in the example below. These parameters vary across the given models, which are listed below and described in more detail in their own sections:

```
#
PROPAGATION-PATHLOSS [0]   TWO-RAY
PROPAGATION-PATHLOSS [1]   FREE-SPACE
#
```

Shadowing Model

The Shadowing Model is a log-normal distribution with standard deviation σ [dB], and can be combined with the Free Space Model, along with the Fading Model. The PROPAGATION-SHADOWING-SIGMA parameter specifies the sigma (σ) to use with this model. This can be either a global default, or specified per channel, using the Instance ID specification, as in the example below.

```
#
PROPAGATION-SHADOWING-SIGMA [0] 0.0
PROPAGATION-SHADOWING-SIGMA [1] 0.0
#
```

Fading Model

The Fading Models available in QualNet are narrowband flat fading models which implement the Rayleigh and Ricean distributions? The Rayleigh distribution is useful in highly mobile cases and signals without line of sight. The Ricean distribution is more useful in the presence of line of sight. The PROPAGATION-FADING-MODEL parameter specifies the distribution to use. This can be either a global default, or specified per channel, using the Instance ID specification, as in the example below.

```
#
PROPAGATION-FADING-MODEL [0] NONE
PROPAGATION-FADING-MODEL [1] RAYLEIGH
# PROPAGATION-FADING-MODEL RICEAN
# PROPAGATION-RICEAN-K-FACTOR 0.0
```

REFERENCES

[1] sales@scalable-networks.com
[2] (http://www.scalable-networks.com)
[3] support@scalable-networks.com
[4] (http://www.scalable-networks.com)
[5] support@scalable-networks.com http://www.scalable-networks.com