Chapter 2

Simulation and Emulation Techniques for Performance Evaluation of Wireless Networks

2.1 Introduction

Performance evaluation gives a measure of the service delivered by a system [1] and performance is one of the most important non-functional aspects of any (hardware or software) system. Performance evaluation comprises of certain techniques such as direct measurements using testbeds, analytical or simulation modeling which can be applied to existing or envisioned systems like computer systems, communication networks, algorithms and protocols [2]. The basic and final goal of the performance evaluation study is to come to a conclusion, whether the system is working in a good condition or satisfactorily.

With the increase in user demands, the field of communication is continuously evolving and to reach these demands new standards are being developed by the organizations like Institute of Electrical and Electronics Engineers (IEEE), Third-Generation Partnership Project (3GPP). Designing and performance evaluation of these new emerging technologies is a major challenge, especially considering the fact that wireless medium is unpredictable. This makes the performance evaluation of the networks under different circumstances very crucial in order to understand the functioning of a newly upcoming standard.

Performance evaluation is needed at several stages in design phase of a standard. In early stages, when the design is being conceived, performance evaluation is used to make early design tradeoffs. Usually, this is accomplished by simulation models; with the aid of simulation results, several design decisions are made which is followed by the prototype building. Once the design is finalized and is being implemented, simulation is used to evaluate functionality and performance of subsystems of the standard. Later, performance measurement is done after the product is available in order to understand the performance of the actual system to various real world workloads and to identify modifications to incorporate in future designs.

Performance evaluation of new emerging technologies like WiMAX is very critical in order to understand and improvise it to the desired level. New protocols and algorithms or the modification to the existing one have to be analyzed thoroughly. The selection criterion of performance evaluation technique and tool mainly depends on the system to be studied and the availability of performance evaluation tool.

2.2 Performance Measures

Performance is a quantitative measure to judge the behavior of the system under different circumstances. The major goal of performance evaluation is to assess performance measures of interest such as delay, jitter, throughput, goodput, loss rate, utilization, processing time and blocking probability.

The measure *good performance* is subjective which may vary from an individual to individual user. The objective measures are also there to judge the performance; these are based on numerical values and can be reproduced if the evaluation is done by others. There are two basic types of performance measures; the first one is system oriented performance measures which are independent of specific applications, whereas the second category of application oriented measures may be dependent on system measures in complex way. For example if video conferencing over internet application is considered, application measures include frame rate, resolution etc and the system measures include throughput, delay etc.

2.3 Methods of Evaluation

Performance evaluation can be classified into two types: (i) performance measurement and (ii) performance modeling [3]. Performance measurement is possible only if the system of interest is available for measurement and only if one has access to the parameters of interest. Performance modeling is typically used when actual systems are not available for measurement or if the actual systems do not have test points to measure every detail of interest. Performance modeling may further be classified into simulation modeling and analytical modeling. Simulation models may further be classified into numerous categories depending on the mode/level of detail of simulation. Analytical models use probabilistic models, queueing theory, Markov models or Petri nets. The following methods of evaluation can be employed to compute various performance and dependability measures:

2.3.1 Measurement-based Evaluation

Desired metrics are obtained from measurements of the system measured in either an operational or a controlled environment. This technique inherently yields the most believable results. However, measurement based evaluation is a very expensive technique and moreover since it requires an actual system, evaluation is not applicable during system design.

2.3.2 Model-based Evaluation

This technique will almost always deliver results which are less accurate than the ones that can be obtained by conducting actual measurements on the system. However, model-based evaluation overcomes the two primary limitations of measurement-based evaluation, viz., it is less expensive and it can be used during the design phase. Models can be of two types: a) analytical models and b) simulation models.

Analytical Models: Mathematical models can be used to predict system behavior. Compared to simulation models, analytical models are less expensive. Also, with the availability of very powerful and effective general purpose modeling tools, analytical models are cost effective than simulation.

Simulation Models: The system behavior can be simulated and the desired performance and dependability measures can be estimated. Since the actual system is not needed, this technique can be applied during the system design phase. However, for credible simulation results, the system has to be simulated for a long time. The advantage of simulation over analytical modeling lies in the fact that very detailed system behavior can be captured.

2.4 Performance Modeling Techniques

The performance evaluation technique requires an intimate knowledge of the system being modeled and a careful selection of the methodology, workload and tools. The goal of every performance study is either to compare different alternatives or to find the optimal parameter value. Real networks can thus be replaced by mathematical models and its properties inferred from the properties of these models. Analytical models generally provide the best insight into the effects of various parameters and their interactions. Nevertheless, the results would only be as good as the models themselves and over the years it is learnt that the process of modeling proved to be difficult for complicated systems. In the modern era, computers provide alternative experimentation techniques to create logical models of systems. These models are executed in a computer environment so as to simulate system behavior. Use of computational power for model execution makes it possible to evaluate more complex systems in more detail, including network devices, applications and protocols [4].

2.4.1 Analytical Modeling

Analytical modeling uses mathematical notions and models describing certain aspects of a system. Modeling of computer systems and communication networks often use probabilistic models. It is a simple method but the results obtained by it are not reliable many times, as it assumes a simple modeling of the real system. It provides a cost effective method of network dimensioning and helps in understanding the relationship among the system, traffic and performance. Compared to the alternative methods of performance evaluation such as simulation, it gives the quick results and insights into the behavior of the system [5]. The modeling phase is a bit difficult part in this technique, as the system parameters and performance metrics have to be selected properly.

Analytical models rely on probabilistic methods, queuing theory, Markov models or Petri nets to create a model of the computer system. Analytical models are cost-effective because they are based on efficient solutions to mathematical equations. However, in order to be able to have tractable solutions, often, simplifying assumptions are made regarding the structure of the model. As a result, analytical models do not capture all the detail typically built into simulation models. Several different mathematical model types can be used to abstract the essential characteristics of computer systems and analyze or predict systems behavior. Each of these model types differ with respect to the type of systems and real-life situations that they can represent and also the kind of measures that can be computed.

Advantages of using analytical modeling

- Thorough understanding of the system is required.
- Quick evaluation can be done.
- Though it is a simple abstract of the real system, can provide the qualitative insights into the system [2].

Disadvantages of using analytical modeling

- Many times the systems are complex for analytical modeling which may require some level of simplifications and approximations to the system to be assessed. But this compromise may be expensive from the perspective of expected results.
- A strong background in mathematics and probability theory needed [2].

2.4.2 Simulation Modeling

A simulation model is a computer program, written in a general purpose or specific simulation language. Simulation implements the most important aspects of the system under study, often in a simplified manner and it allows for a greater level of detail than analytical modeling. Simulation is a reliable performance evaluation technique which abstracts a real system and models it in a basic level. It replicates the real system's behavior with real environment settings. Very often simulation results are trustworthy and can be used to assess the basic behavior of the system performance and to estimate the system capacity. Simulation is a low cost technique compared to real time measurement and hence widely used method of performance evaluation technique. Computer and communication systems are dynamic, stochastic, discrete-state and discrete-time systems. A discrete event simulation of such systems needs to modify the model's state only at discrete times, between these times the state is guaranteed not to change. Therefore the state of the art of simulation model should meet all the requirements of the technologies and systems.

According to [6], simulation is *the process of designing a model of a real system and conducting experiments with the model for the purpose of understanding the behavior of the system and/or evaluating various strategies for the operation of the* *system*. In order to evaluate the behavior and performance of protocols for wireless networks, simulations are a good compromise between cost and complexity on the one hand and accuracy of the results on the other hand [7].

Advantages of simulation modeling

- Simulation results are often much better reproducible than direct measurement results (e.g. when wireless channels are involved).
- All parameters are under control.
- It provides a better approach and visualizing effects of performance evaluation.

Disadvantages of simulation modeling

- A long simulation time is required in order to achieve a desired statistical accuracy for some experiments; in some cases it is impossible to get meaningful results within reasonable times.
- To cover a meaningful fraction of the whole parameter space.
- Model setup may take a long time, also validation and verification time can be significant. Validation makes sure that the assumptions behind the model suit the real system's behavior and verification confirms that the actual code of the simulation model fits the claimed model assumptions.

2.5 Network Simulation and Emulation Tools

Network simulators have grown in maturity since they first appeared as performance management and prediction tools. The major goal of network simulation is to model the network performance in order to verify the distributed functions, debugging the network protocols and to test the reliability of new components [8]. The network simulator provides an integrated, versatile, easy-to-use Graphic User Interface (GUI)-based network designer tool to design and simulate a system with network devices (hosts, hubs, bridges, routers, subnets, base stations and mobile units etc.,) [9]. The simulator accepts input as a scenario, which is a description of network topology, protocols, workload and control parameters. It produces an output statistics such as the number of packets sent by each source of data, the delay at each queuing point and the number of dropped and retransmitted packets [10]. A simulator uses lower quality reproduction or abstraction of the real system and focuses on simply replicating the real network's behaviour. A network simulation is a very low cost method for developing the early stages of network centric systems [11-13].

To evaluate the network performance, simulator uses virtual traffics, which are generated through a computer program. A more advanced way of performance evaluation is provided through *emulation,* which refers to the ability to introduce the simulator into a live network. Emulation is a hybrid experimentation technique intended to bridge the gap between simulation experiments and real world testing. The key idea of emulation is to reproduce the essential functionality of a system in real time and in a controlled manner. A network emulator imitates the functions of a real network so that it appears, interacts and behaves like a real network and can thus be evaluated. The emulator provides an exact, high quality reproduction of external behaviour so that the emulated system is indistinguishable from the real system. The major difference lying in network emulator from the simulator is that end-systems such as computers can be attached to the emulator, which will act exactly as they are attached to a real network. An emulator provides a cost-effective method of evaluating new network technologies before actual systems or networks are built. Network emulation helps in developing a net-centric system by providing an environment in which design decisions can be easily changed and their impact is evaluated before the actual system is built. This also sets realistic expectations of the communication network, i.e., it provides predictability [13].

A number of open source/commercial network simulators and emulators are available as listed in the table 2.1; some selected network simulators and emulators are discussed in the following sections.

Table 2.1 Network simulators and emulators

Type	Network simulator	Network emulator
Commercial	OPNET, QualNet,	EXata
Open source	NS-2, NS-3, OMNET++, NCTUns	NS3

2.5.1 Network Simulator-2 (NS-2)

Network Simulator-2 (NS-2) is one of the most popular open source discrete event network simulators. NS-2 provides substantial support for simulation of TCP, routing and multicast protocols over wired and wireless networks. The first version of

NS, known as NS-1, was developed at Lawrence Berkeley National Laboratory (LBNL) in the 1995-97 timeframe. Hence NS-1 was also known as the LBNL Network Simulator. NS-2 is the second version of Network Simulator (NS) project, which is supported through Defense Advanced Research Project Agency (DARPA). The current version of NS-2 is 2.34. In NS-2, a network animator (NAM) provides packet-level animation and protocol specific graph for design and debugging of network protocols [14].

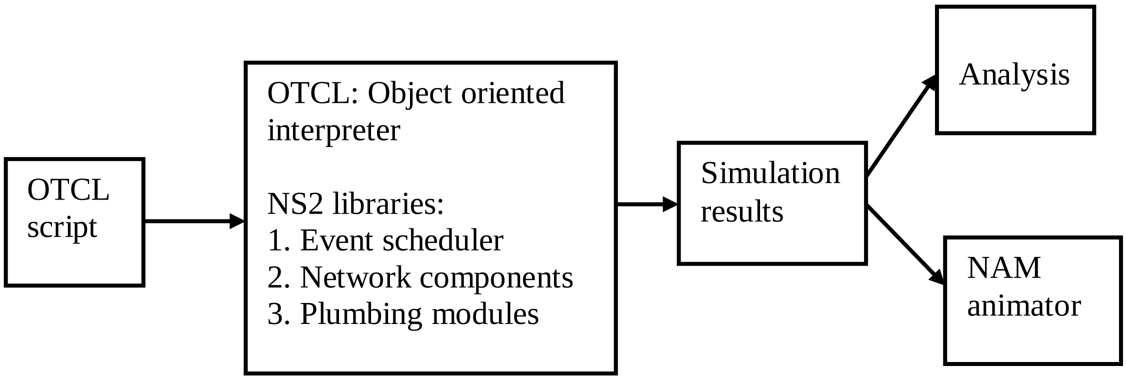

Figure 2.1 Architectural view of NS2

The NS-2 is widely used in academic research and it has lot of packages contributed by different non-benefit groups, a general view of NS-2 is shown in fig.2.1. For programming, NS-2 uses C++ and object oriented tool command language (OTcl) and utilizes advantages of both the languages. C++ is efficient to implement a design but it is not very easy for visual and graphical display. OTcl have the feature that C++ lacks. So the combination of these two languages proves to be very effective. C++ is used to implement the detailed protocol and OTcl is used for users to control the simulation scenario and schedule the events. For efficiency reason, NS-2 separates control path implementations from the data path implementation. The event scheduler and the basic network component objects in the data path are written and compiled using C++ to reduce packet and event processing time. The OTcl script is used to initiate the event scheduler, setup the network topology and inform traffic source when to start and stop sending packets through event scheduler; alternatively OTcl script can also be programmed. The event scheduler in NS-2 keeps track of simulation time and release all the events in the event queue by invoking appropriate network components. All the network components use the event scheduler by issuing an event for the packet and waiting for the event to be released before doing further action on the packet.

2.5.2 Network Simulator-3 (NS-3)

Network Simulator-3 (NS-3) is also an open sourced discrete-event network simulator which targets primarily for research and educational use. Both simulation and emulation studies can be carried out using NS-3. The architecture of NS-3 is shown in figure 2.2. The goal of the NS-3 project is to create an open simulation environment for networking research that will be preferred inside the research community. The first release, NS-3.1 was made in June 2008 and afterwards the project continued making quarterly software releases and more recently has moved to three releases per year. NS-3 made its fifteenth release (NS-3.15) in the third quarter of 2012.

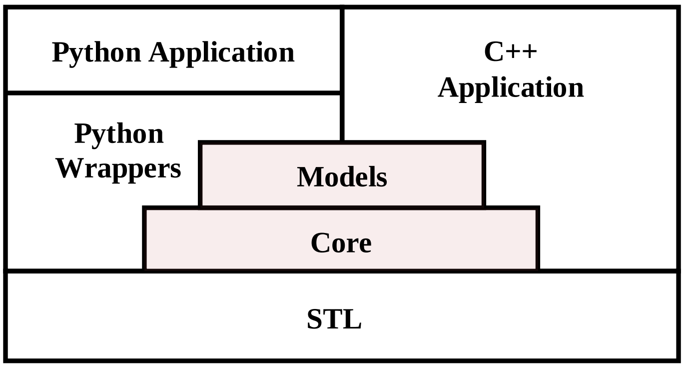

Figure 2.2 Architecture of NS3

NS-3 is built using C++ and Python and scripting is available with both languages. NS-3 is often criticized for its lack of support for protocols which were supported in NS-2, as well as for the lack of backward compatibility with NS-2. As with NS-2, NS-3 is also cumbersome to learn and practice compared to GUI-based simulators [15]. NS-3 redesigns a lot of mechanisms based on the successful and unsuccessful design experiences of NS-2.

2.5.3 NetSim

NetSim is a popular network simulation tool used for network lab experimentation and research. It is a stochastic discrete event simulator developed by Tetcos, in association with Indian Institute of Science with the first release in June 2002. Various technologies such as Wireless Sensor Networks, Wireless LAN, WiMAX, TCP, IP, etc. are covered in NetSim. The code for NetSim has been written in C and Java. NetSim provides network performance metrics at various abstraction level such as Network, sub-network, Node and a detailed packet trace.

NetSim comes with an in-built development environment, which serves as the interface between User's code and NetSim's protocol libraries and simulation kernel. Protocol libraries are available as open C code for user modification. De-bugging custom code during simulation is an advanced feature: i.e. a simulation can be started and then at user determined breakpoints in the code, users can perform single-step, step-in, step over etc. This can be carried out at various levels (depending on where the user code links) including at a per-packet interval.

2.5.4 J-Sim (Java-Simulator)

J-Sim is a network simulator written in Java and is built according to the component based software paradigm, which is called Autonomous Component Architecture (ACA). In J-Sim, each component can be atomic or composed of other components interconnected through ports. J-Sim ports support one-to-one, one-to-many and many-to-many connections. J-Sim protocol architecture comprises of two layers, the lower layer called Core Service Layer (CSL) consists of network to physical layers and the higher layer consists the remaining of Open System Interconnection (OSI) layers.

Although J-Sim does not have tool for network visualization, it allows generating trace files which conform to NS-2's NAM format. J-Sim offers good introductory material with overviews and examples for small scenarios. However, it lacks a comprehensive manual. J-Sim uses tool command language (Tcl) for configuration of simulation scenarios which requires a certain learning overhead.

2.5.5 OMNeT++

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators. OMNeT++ provides component architectures for models, which are programmed in C++ and then assembled into larger components/models. OMNeT++ is not a simulator in itself but rather a simulation framework. Instead of containing explicit and hardwired support for computer networks or other areas, it provides the infrastructure for writing such simulations. Specific application areas are catered by various simulation models and frameworks, most of them are open source. These models are developed completely independently of OMNeT++ and follow their own release cycles.

OMNeT++ has a well-written fairly large user manual and Application Programming Interface (API) documentation. OMNeT++ is the only simulator with online visualization allowing users to pause the simulation and inspect or even directly change values in the models. It is also possible to change a node's appearance (color, size, shape, etc.) to reflect an inner state which the user wants to visualize. Statistics can be written to a trace file and displayed with external but commonly available tools like *prove*. OMNeT++ is very complex, thus careful consultation of the available documents is needed.

2.5.6 Optimized Network Engineering Tools (OPNET)

Optimized Network Engineering Tools (OPNET) is one of the popular commercial network simulators comprising a suite of protocols and technologies [16]. OPNET provides a sophisticated development environment which is specialized for network research and development. It can be flexibly used to study communication networks, devices, protocols and applications. OPNET offers relatively much powerful visual or graphical support for the users. The graphical editor interface can be used to build network topology and entities from the application layer to the physical layer. Object-oriented programming technique is used to create the mapping from the graphical design to the implementation of the real systems; an example of the Graphical User Interface (GUI) of OPNET can be seen in figure 2.2.

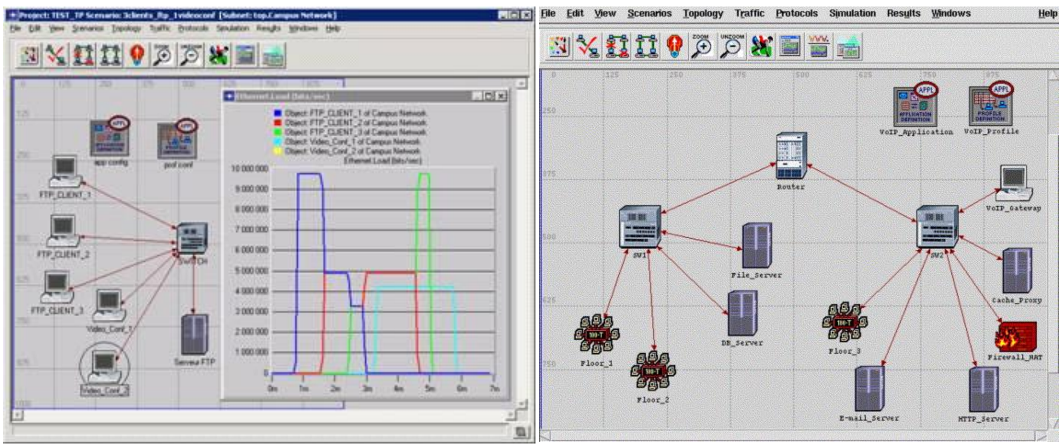

Figure 2.3 Graphical User Interface (GUI) of OPNET

In OPNET all the topology configuration and simulation results can be presented very intuitively and visually. The parameters can also be adjusted and the simulations can be repeated easily through GUI. OPNET is based on a mechanism called discrete event system which means that the system behavior can simulate by modeling the events in the system according to the order of scenarios set up by user. Hierarchical structure is used to organize the networks. As other network simulators, OPNET also provides programming tools for users to define the packet format of the protocol. The programming tools are also required to accomplish the tasks of defining the state transition machine, defining network model and the process module. As of all, OPNET is one of a popular simulators used in industry and academia for network research and development.

The main features of OPNET are listed below,

- Fast discrete event simulation engine
- Lot of component library with source code
- Object-oriented modeling
- Hierarchical modeling environment
- Scalable wireless simulations support
- 32-bit and 64-bit GUI
- Customizable wireless modeling
- Discrete event, hybrid and analytical simulation
- 32-bit and 64-bit parallel simulation kernel
- Grid computing support
- Integrated, GUI-based debugging and analysis
- Open interface for integrating external component libraries

2.5.7 QualNet

QualNet is a planning, testing and training tool that "mimics" the behavior of a real communication networks. QualNet provides a comprehensive environment for designing protocols, creating and animating network scenarios and analyzing their performance. It is a comprehensive suite of tools for modeling large wired and wireless networks provided by Scalable Networks Technologies (SNT) Inc [12]. It uses simulation and emulation to predict the behavior and performance of networks to improve their design, operation and management. Many contributed models are added by non benefit groups and researchers to QualNet. The latest version of QualNet is 6.1. The architecture of QualNet (figure 2.4) is composed of simulation kernel, extensive model libraries like advanced wireless library for WiMAX, Wireless Sensor Networks (WSN), Universal Mobile Telecommunication Systems (UMTS), satellite communication etc. and a well built GUI.

QualNet is composed of the following components,

QualNet architecture: A graphical simulation design and visualization tool. Architect has two modes; design mode for designing simulations and visualize mode for running, visualizing simulations. In design mode, network models can be created by using intuitive, click and drag operations. In Visualize mode, as simulations are running, users can watch packets at various layers flow through the network and view dynamic graphs of critical performance metrics.

QualNet analyzer: A graphical statistics analyzing tool which is useful to plot the obtained results by running the simulation. Multi-experiment reports are also available. All statistics are exportable to spreadsheets in CSV (Comma Separated Value) format.

QualNet packet tracer: A graphical tool to display and analyze packet traces. Trace files are text files in XML format that contain information about packets as they move up and down the protocol stack

QualNet file editor: A text editing tool.

QualNet command line interface: Command line access to the simulator.

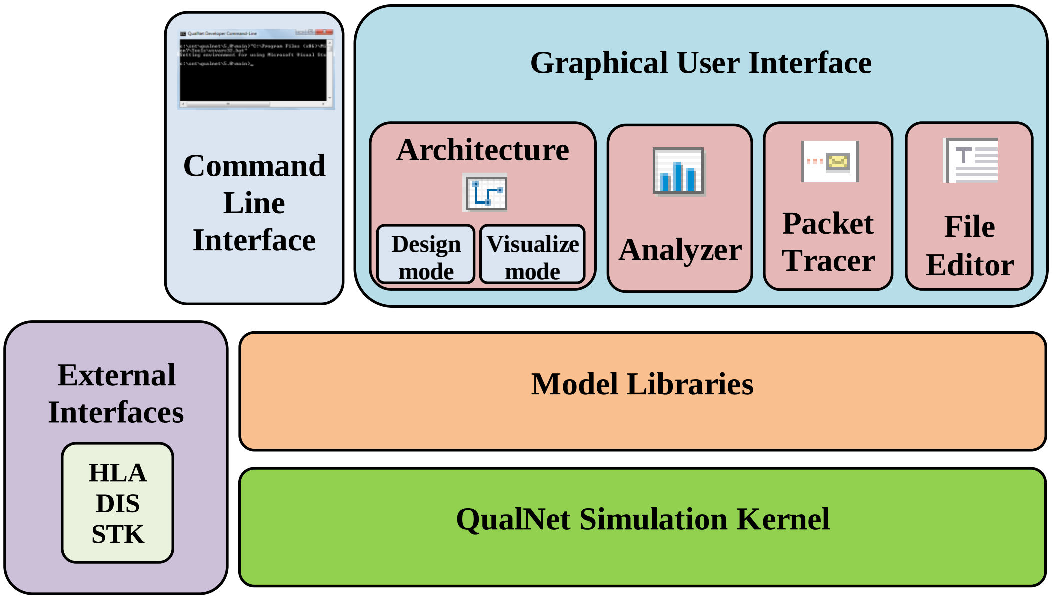

Figure 2.4 Architecture of the QualNet

QualNet enables users to design new protocol models and to optimize new and existing models. Users are allowed to design large wired and wireless networks using pre-configured or user-designed models and analyze the performance of networks. The performance measures of the models are analyzed to optimize them. The general procedure of simulation in QualNet is demonstrated by figure 2.5.

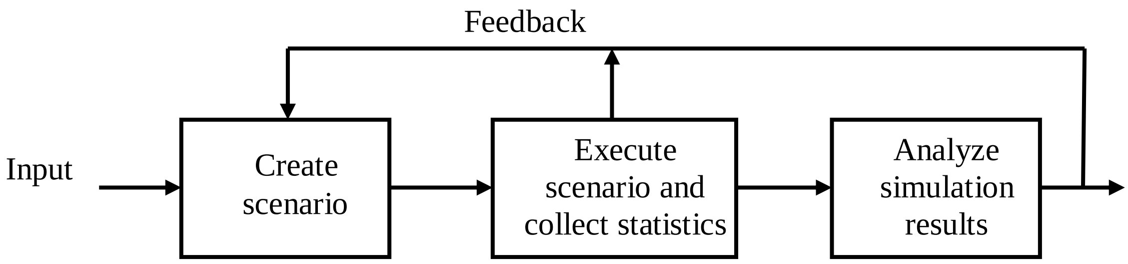

Figure 2.5 General procedure of simulation in QualNet

The key features of the QualNet, which make it one of the popular tools among network simulators for research community, are as discussed below.

Speed: QualNet can support real time speed to enable software-in-the-loop, network emulation and hardware-in-the-loop modeling. Faster speed enables model developers and network designers to run multiple *what-if* analyses by varying model, network and traffic parameters in a short time.

Scalability: QualNet can model thousands of nodes by taking advantage of the latest hardware and parallel computing techniques. QualNet can run on cluster, multi-core and multi-processor systems to model large networks with high fidelity.

Model fidelity: QualNet uses highly detailed standards-based implementation of protocol models. It also includes advanced models for the wireless environment to enable more accurate modeling of real world networks.

Portability: QualNet and its library of models run on a vast array of platforms including Windows XP, Mac OS X, Linux operating systems, distributed and cluster parallel architectures for both 32-bit and 64-bit computing platforms. Users can develop a protocol model or design a network in QualNet on Windows XP computer and then transfer it to a powerful multi-processor Linux server to run capacity, performance and scalability analyses.

Extensibility: QualNet can connect to other hardware and software applications, real networks and third party visualization software to greatly enhancing the value of the network model [12].

2.5.8 EXata

Emulator is a high fidelity abstraction model, which mimics the functions and behavior of real system so that it appears like the system to be studied, communicates and behaves like a real system. An emulator is similar to a simulator, except that instead of redefining many levels of the OSI model, an emulator only redefines the particular layer, for e.g. if the real time traffic is applied then application layer. Hence all the other features of the node participating in emulation remains same, except the separately modeled layers' features. This solution offers an adequate middle ground between simulators and testbeds [17]. Required feature of the experiment can be emulated in hardware, allowing for better performance and accuracy compared to simulators. At the same time, an emulator eliminates many of the practical problems with testbeds by completely controlling external factors that may influence an experiment.

EXata [13] is a network emulator and simulator provided by SNT, that lets user evaluate on-the-move communication networks faster and with more realism than any other tool. It uses a Software Virtual Network (SVN) to digitally represent the entire network, the various protocol layers, antennas and devices. The system can interoperate, at one or more protocol layers, with real radios and devices to provide hardware-in-the-loop capabilities. EXata can also be connected to systems with real applications, which run on the SVN just as they would run on real networks.

EXata is a superset of QualNet with the hands-on capacity of emulation; it is a comprehensive suite of tools for emulating large wired and wireless networks. It uses simulation and emulation to predict the behavior and performance of networks to improve their design, operation and management. Fig 2.6 reveals the general architecture of EXata and fig.2.7 shows the general procedure of simulation/emulation in EXata.

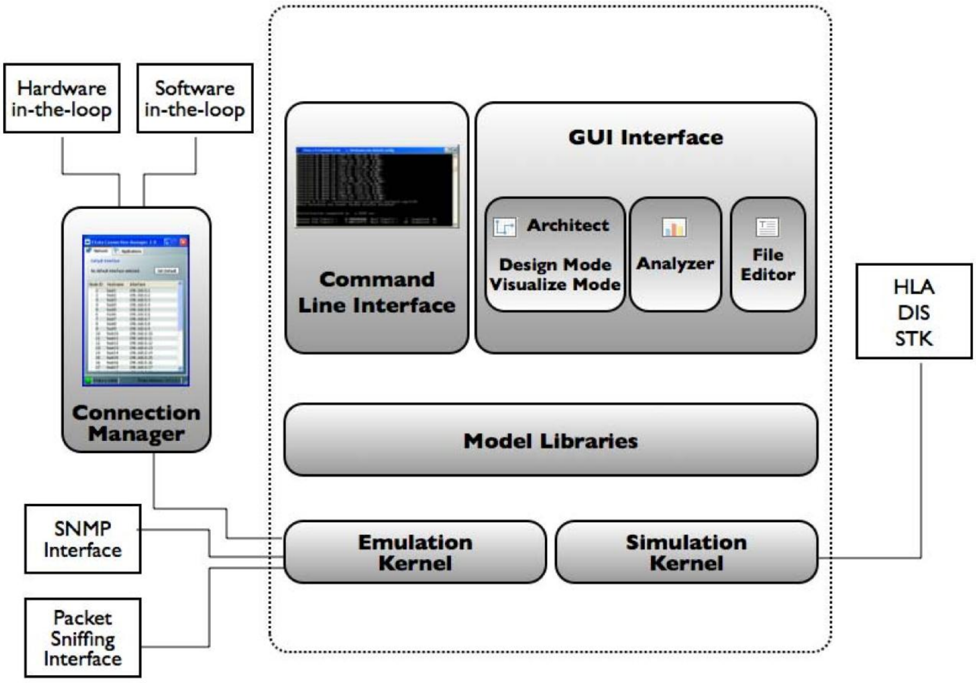

Figure 2.6 Architecture of EXata

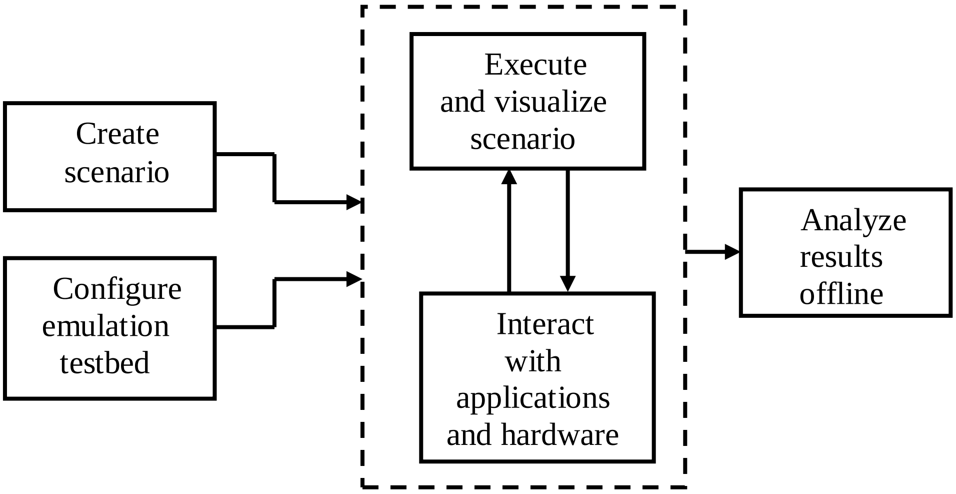

Figure 2.7 General procedure of simulation and emulation in EXata

EXata is very similar in its architecture to QualNet, but with some additional blocks to support emulation. The additional blocks are discussed below.

- The kernel consists of both simulation and emulation kernels.
- ***EXata connection manager***: EXata connection manager is the companion module of the main EXata emulation engine. The EXata emulation engine creates a digital replica of the target network and EXata connection manager is used to run applications on the emulated network. The connection manager makes EXata advanced emulation technology easy and simple to use.

Applications need no modification or customization to use the realistic emulated network in EXata. Connection manager supports a large variety of applications such as: internet browsers, tactical communications, situational awareness information, sensor data, instant messaging, VoIP, streaming video and multiplayer games.

- ***EXata external interface***: EXata supports a packet sniffer interface to enable capture and analyze the network traffic using standard packet sniffer/analyzer tools like Wireshark or Microsoft network monitor. Additionally, EXata can be managed using standard simple network management protocol (SNMP) network managers like HP Open view, IBM Tivoli or Solar Winds Orion.

A comparative study of popular simulation tools is summarized in Table 2.2.

Table 2.2 Comparative study of different Simulation tools

	NS-2	**OMNET++**	**OPNET**	**QualNet**
License Model	Open Source	Open Source for academics	Commercial License	Commercial License
Language “Model”	Combination of Tcl and ANSI C	OO-Design / C++ OO-Designed models	ANSI C / C++ Procedural model	ANSI C / C++ Procedural model
Model Library	Main modules without interoperability are available	Main modules available.	Main modules available.	Main modules available.
Access to Source Code	Full access to simulation kernel and modules source code.	Full access to simulation kernel and modules source code.	No access to simulation kernel source code. Full access to modules source code.	No access to simulation kernel source code. Full access to modules source code.
Available Documentation	Well documented	Well documented	Well documented	Well documented
Assumed Period of professional Adjustment	Very high	Moderate to high	Moderate to high	Moderate

2.6 Simulation Studies of WiMAX using QualNet

QualNet consists of well defined comprehensive advanced wireless library module to support the WiMAX network simulations. This tool facilitates to study the performance of the different network parameters such as throughput, average end-to-end delay and average jitter, etc. at the nodes.

QualNet PHY802.16 supports OFDMA PHY, variable channel bandwidth, different FFT sizes, multiple cyclic prefix time and different modulation schemes such as QPSK, 16QAM and 64QAM with convolutional encoding at variety encoding rates. The MAC802.16 model of QualNet has implemented features defined in both IEEE 802.16 and IEEE 802.16e. The detailed list of implemented features is:

- Point to Multi-Point (PMP) mode
- Time Division Duplex (TDD) mode
- Network entry and initialization
- Bandwidth management
- Five service types: UGS, ertPS, rtPS, nrtPS and BE
- Adaptive Modulation and Coding (AMC)
- Support both IPv4 and IPv6
- IEEE 802.16e Mobility Support
- Power saving modes
- Interface other networks such as ATM, 802.3, 802.11 at network layer
- Simple admission control
- ARQ

QualNet has model for IEEE 802.16j standard, contributed by the authors of [18], in which basic 802.16j MAC, Mobile relay base station (MR-BS) and transparent relay station (T-RS) are implemented.

2.6.1 Mobile WiMAX Simulation

An example scenario is considered in order to evaluate the performance of WiMAX network for mobile subscriber stations (SSs) and fixed SSs. Figure 2.8 shows the snapshot of example scenario captured from QualNet GUI. The scenario consist of five WiMAX subnets, each subnet corresponds to one WiMAX cell. Each WiMAX cell consists of a single base station (BS) and three SSs. All the BSs of these subnets are connected to a switch using point to point links. The SSs transmits Constant bit rate

(CBR) traffic to other SS (the application mapping is indicated with the green line). Simulations are executed by keeping the SSs fixed and also by giving random way point mobility. The results are compared by considering throughput and delay as performance metrics. The simulation parameter settings are listed in the table 2.3. Figure 2.9 and 2.10 show the throughput and delay performances at destinations respectively.

Table 2.3 Simulation parameters

Property	Value
Simulation time	300 Sec
BS/SS transmission power	20 dB
Channel frequency	2.4 GHz
Pathloss model	Two ray
Shadowing model	Constant
Antenna model	Omnidirectional
MAC frame duration	20ms
TDD Uplink to downlink ratio	1:1
BS Height	10m
SS Height	1.5m

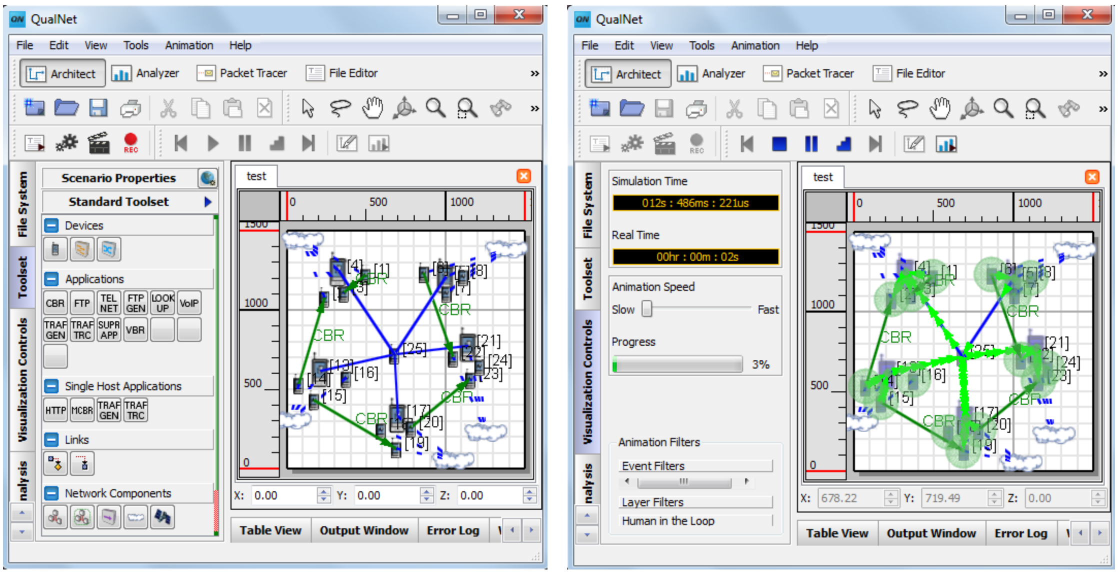

Figure 2.8 Multi cell scenario in mobile WiMAX

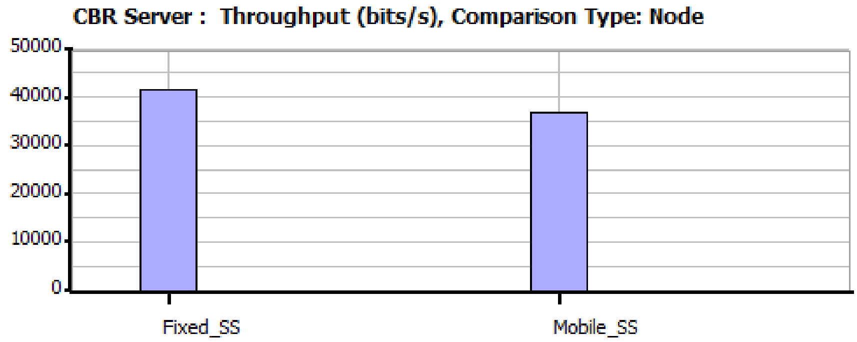

Figure 2.9 Throughput performance at destinations

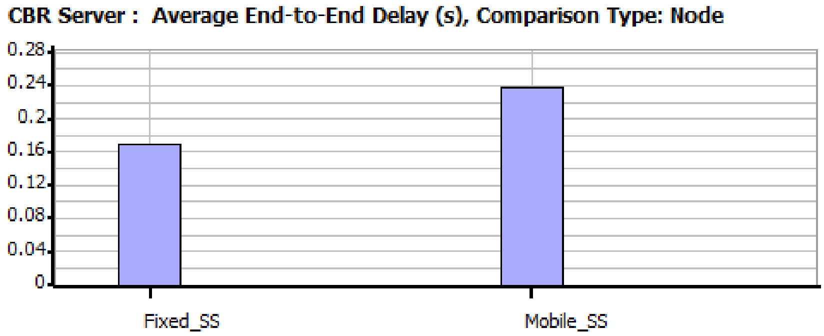

Figure 2.10 Average end to end delay at destinations

2.7 Conclusion

Performance evaluation of a new emerging technology like WiMAX is very imperative in the design, deployment and capacity planning. Performance analysis results can be used to improve the existing algorithms or may help in proposing new algorithms. With parallel execution and highest fidelity models, QualNet has the scalability and interactivity required to test and evaluates the most advanced network-centric systems.

References

[1] Jean-Yves and Le Boudec, “Performance Evaluation of Computer and Communication Systems”, *EPFL Press, Lausanne, Switzerland*, 2010.

[2] Raj Jain, “The Art of Computer Systems Performance Analysis - Techniques for Experimental Design, Measurement, Simulation and Modeling”, *Wiley Professional Computing, John Wiley and Sons*, New York, Chichester, 1991.

[3] Swapna S. Gokhale and Kishor S. Trivedi, “Analytical Modeling”, *In Encyclopedia of Distributed Systems*, Kluwer Academic Publishers, 1998.

[4] Razvan Beuran, “Introduction to Network Emulation”, *Pan Stanford Publishing Pte. Ltd*. 2013.

[5] Aleksi Penttinen, “Algorithms and Performance Evaluation Methods for Wireless Networks”, *Doctoral thesis, Helsinki University of Technology Networking Laboratory,* 2006.

[6] R. E. Shannon, “Introduction to the art and science of simulation”, *in Proceedings of the 30th conference on Winter simulation*, 1989.

[7] Johannes Lessmann, Peter Janacik, Lazar Lachev and Dalimir Orfanus, “Comparative Study of Wireless Network Simulators”, *7th International Conference on Networking (ICN), IEEE Computer Society,* 2008.

[8] Murat Miran K¨oksal, “A Survey of Network Simulators Supporting Wireless Networks”, *Technical report, Middle East Technical University,* October 2008.

[9] http://www.webnms.com/simulator/network-simulator-ds.html

[10] http://www.cs.cornell.edu/skeshav/real/overview.html

[11] Thontadharya H.J., Shwetha D., SubramanyaBhat M. and Devaraju J.T., “Simulation and Emulation Approach for the Performance Evaluation of Adaptive Modulation and Coding Scheme in Mobile WiMAX Network”, *International Journal of Computer Applications* (0975 – 8887) Vol. **44**, No16, April 2012.

[12] Documentation of QualNet User’s Guide

[13] Documentation of EXata 2.0.1 User’s Guide

[14] http://www.isi.edu/nsnam/ns/

[15] http://en.wikipedia.org/wiki/Ns-3

[16] http://www.opent.com\documentation

[17] Sam Burnett, “Combining Wireless Network Simulation and Emulation”, *Undergraduate thesis, Carnegie Mellon University*, May 2008.

[18] Vasken Genc, “Performance Analysis of Transparent Mode IEEE 802.16j Relay-based WiMAX Systems*”, Doctoral thesis, University College Dublin,* 2010.