**Network Simulation and Emulation and Benchmarks**

**Introduction**

On small networks, you can use network monitoring to help with capacity planning or diagnose problems or predict the network's reaction to new hardware or software. However, as networks grow, their numerous devices and connections make understanding the network impossible and testing live production systems is fraught with danger as most companies rely heavily on an always available network. Too many conversations among too many devices via too many network routes occur simultaneously for you to accurately predict how one application's traffic will affect another part of the network. Simulation, emulation and benchmark testing can all help with the design, development or management of networks.

**Simulators and Network modelling**

# Systems Evaluation

Typically, there are three approaches to evaluate systems. Firstly, performance evaluation can be conducted by emulation through building a testbed or mini-scale of the system. Emulation basically can be defined as the series of activities that implement the real hardware and software that is going to be used in the real world scenario in smaller scale. This needs a relatively realistic amount of hardware and software. The real advantage of this method is that the evaluation results are expected to be very close to the real system’s performance as we use very similar hardware and software configurations when evaluating it. On the other hand, building a mini-scale of the proposed network architecture is often costly and time consuming. Probably, the time needed to build and develop the testbed can be close to or even the same as the time needed to implement the real system. In terms of facilities, in many large enterprises maybe it is not a major issue as many blue chip companies usually have test-bed facilities in place. However, it is a serious issue for small and medium companies (SME) which tend to have a limited budget and therefore limited resources.

Secondly, mathematical analysis or modeling with numerical analysis can be used to evaluate the system in question. Compared to emulation through building a mini-scale of the system, and performing the tests and performance evaluation on the mini-scale system can theoretically design the system and evaluate analytically (usually by creating a model and a set of computer codes) its performance. Obviously, the main advantage of the mathematical analysis is the engineering cost, as it does not require implementing hardware and software to evaluate the performance of the system in question. However, as many assumptions need to be made to build a reasonable tractable analytical model, the output may not be very close to the real system. Often, numerical analysis shows the borderline behavior of system characteristics or offer upper and lower bounds for specific research questions. However, more fine-grained analysis often leads to an unacceptable complexity of the analytical models.

Thirdly, performance evaluation can be carried out through simulations. Simulation is a compromise between emulation and analytical modeling, or even to get the best of both worlds: emulation and analytical modeling. Simulation offers a controlled environment in which a system can be investigated in more detail. Different parameter settings and scenarios can be analyzed with comparably limited effort. Through proper design and implementation, relatively accurate results can be obtained from the simulations. Modest cost, effectiveness and the short time needed are the main advantages of the simulation. Simulation results are also easier to analyze than experimental results because it is relatively easy to log and trace important information at critical points in order to diagnose system’s behavior. Moreover, simulation modeling techniques offer more flexibility, and accuracy compared with analytical modeling. Therefore, simulation modeling is the choice by researchers. In particular, simulation is used to analyze systems which are highly dynamic and whose properties are difficult to capture in a mathematical way.

In a computer simulation, a real-world system is "imitated" over time. Computer simulation is applied in many different fields. Among different types of computer simulation, like discrete-event simulation, event-based simulation, continuous simulation, Monte Carlo simulation, spreadsheet simulation, trace-driven simulation, etc., the dominant simulation technique is discrete-event or event-driven simulation. The main reason behind the popularity of discrete-event based simulation is that its simulation paradigm fits well to the systems concerned and is easily applied. Hence, discrete-event simulation provides a simple and flexible way to evaluate the systems behavior under different conditions.

To diagnose problems or test new applications on a complex network, you could simulate the network, i.e. use a simulation program (simulator) to build a software model of key network elements and test how well the model functions with various traffic loads or network designs. You can use simulation to estimate the effect that deploying a new application will have on your production network. To simulate a production network, you need to construct a reasonable representation of the network's topology, including the physical devices and logical parameters that comprise the network, determine how much traffic is on the network during the period you want to simulate, specify a question you want the simulation to answer and finally, you need to run the model through a simulator.

**Change Simulation**

You might want to perform a change-analysis (what-if) question on a network model. For example, you might want to analyze what would happen if you changed your network's WAN links, LANs, or routers or added a new application to your network. You might want the simulation to answer a question about the network's fault tolerance. For example, you might want to determine how the failure of a specific device or group of devices such as LANs, WAN links, or an entire facility of your organization would affect application demands. Answering these questions can help you in capacity planning, disaster recovery, and life cycle management (see later).

After you decide which question you want your simulation to answer, you might need to tweak the network model's topology and traffic to make sure the simulation addresses your question. Simulators automatically alter a network model to test its fault tolerance; they have built-in utilities that emulate the failure of network devices. You can select devices from a menu, and your simulation will predict what effects that device's failure would have on your production network. However, you must manually alter your model for your simulation to answer a change-analysis question. To add or move servers or users, you must change the volume of bytes an application produces in the model or add network devices or demands to the model before running your simulation. To simulate deployment of an application, you need to gather data about the application's conversations and manually add this data to your network model. Then, you must instruct the simulator to expand the information you've gathered about the application to simulate the activities of many users at many locations within your model.

After you develop a model of your network and define the question you want to answer, you're ready to run simulations.

**Discrete or Analytical Simulation**

Simulators use either a discrete event or an analytical approach to model network traffic. Discrete-event simulation analyzes the network traffic that each packet generates to determine the network's behaviour. The analytical approach makes assumptions about network traffic before a simulation. The analytical method can be as accurate as the discrete-event method, because discrete-event simulators can't store detailed information about each packet in simulations that include the many thousands of conversations (each of which can contain many packets) that most networks generate. Because the discrete-event method is much slower than the analytical method it is best to use an analytical simulation for large networks.

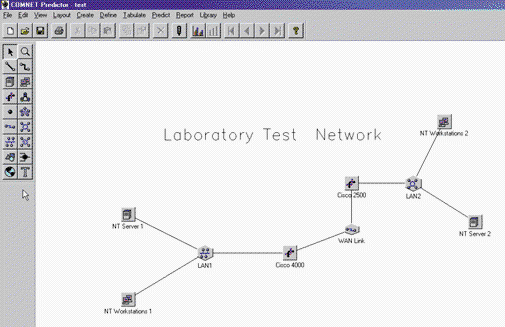
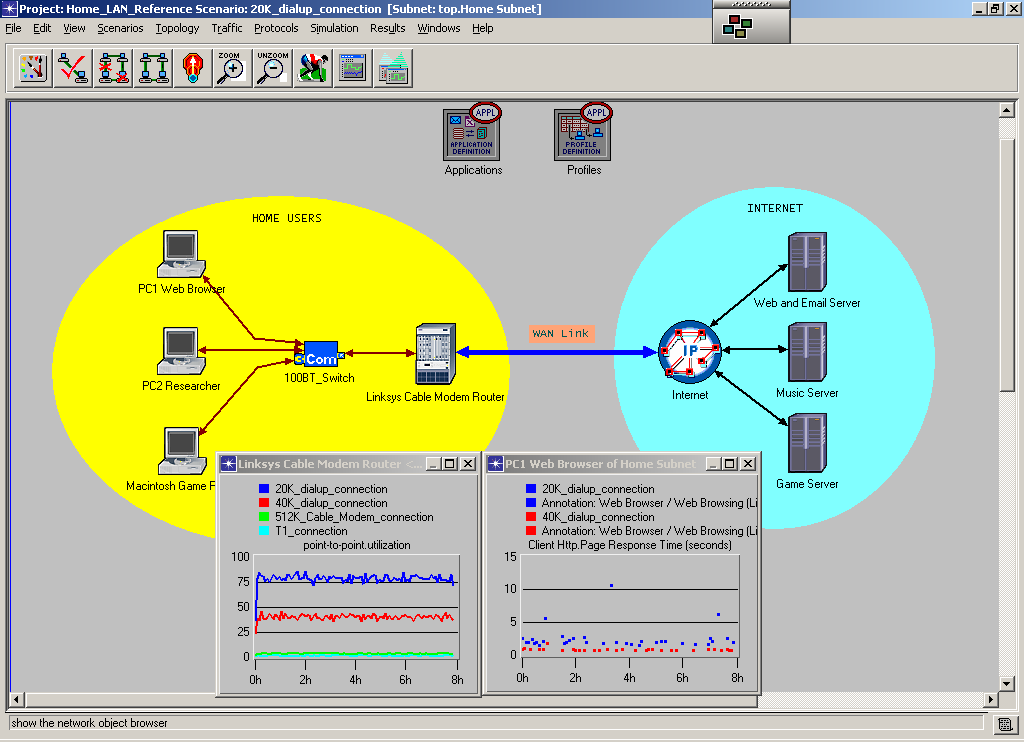
Looking at network traffic at the application level is useful. You can check latency for applications to see whether they are meeting a minimum quality of service. You can see how much throughput each application uses and because you know the source and destination of all traffic flow, you can determine where your intranet traffic is going and which users are using which resources. To simulate deployment of an application, you need to gather data about the application's conversations and manually add this data to your network model. Then, you can instruct the simulator to expand the information you've gathered about the application to simulate the activities of many users at many locations within your model.

**Simulator Packages**

Simulators have traditionally run on powerful UNIX workstations, but some simulators are now available for LINUX and Windows. Most commercial simulators are expensive and running them requires training.

*COMNET* and *OPNET as shown in Figure 1* are two popular commercial packages that employ a graphical interface to aid use. They allow networks to be simulated either to assist with the research and design of new networks or to model an existing network and help predict the effect of changes on network performance (change simulation or What if Analysis).

Figure 1 COMNET and OPNET graphical interface



There are also a number of Open sources or free programmes developed by various research agencies. For instance ns2 (network simulator 2) by the VINT (Virtual Internetwork Testbed) project is popular with the academic community but is not easy to use and requires a degree of programming proficiency.

**Life Cycle Management and Simulation**

Conducting simulations can help with the life cycle management for your network. You need to carry out simulations often, because most networks' topology and traffic are constantly evolving. Figure 2 illustrates how the different stages of simulations fit into the life cycle management of a network.

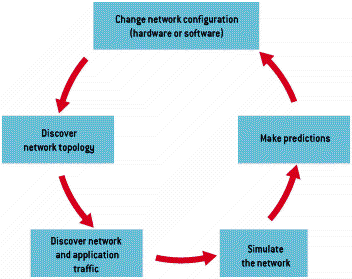


Figure 2 Life cycle management

To maintain a healthy network, use simulations to predict your needs for the future and develop cost-effective solutions to problems before a disaster strikes. Keep an up-to-date inventory of the devices and settings on your network. Conduct traffic analyses to keep abreast of LAN, WAN, and network device utilization. Determine how long users must wait for applications to respond, which might be part of a quality-of-service contract. Use periodic simulations to help plan your capacity for future growth and determine emergency plans to deal with network failures. In addition to your periodic tests, run simulations before you roll out new applications or hardware.

A simulation is neither the starting point nor the end point for answering your network design questions. Running simulations will incite your curiosity and generate more questions than the process answers. As you run simulations, you will learn to appreciate your network's complexity and gain a better sense of how all the components work together and how to optimise it.

**Simulation for Capacity Planning and Network Optimization**

The best way to optimize your network is not to tweak existing, ill conceived network designs, but rather to design a network that will perform to its optimum level from the start. Applying capacity planning can help. Proactive planning procedures involving baseline testing to get a snapshot of network performance and trending to predict the future needs of network resources are essential. But you can’t predict what will happen if you add more users or change hardware or applications. There are two ways to answer these questions are to make the changes to a production network and see what happens or use a simulation and watch what happens to the model of the network when the changes are made. The latter solution is cheaper and faster.

**Visualisation Tools**

The problem with simulators is that they can produce an overwhelming amount of information and results that describe dynamic events can be difficult to interpret when viewed statically. Programmes such as *nam* help with this problem by animating the network simulation to show the packet movement, dropped packets, queues and congestion to get a better idea of what is happening.

Packet Tracer from Cisco is a useful tool to help understand how networks function - you must understand how networks work before you can use a network simulator effectively.

*N.B. Don’t forget the old axiom “garbage in garbage out”- you must ensure that as much detail as possible is included in the model otherwise the results will be worthless. Validation of models is essential, preferably against real systems or at least against other simulators.*

### Network Emulation

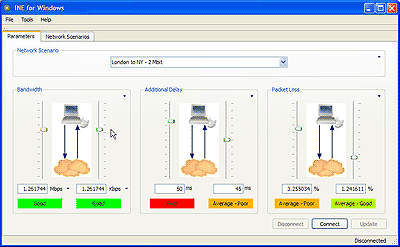
An alternative to simulation modelling is to have a test network that allows you to try out changes to topology, upgrades and new applications without affecting the production network. The main problem however is how to generate traffic that emulates the real world production network.

A network emulator is a device that sits on a network and mimics the behaviour of network devices such as routers or parts of the system such as subnets. Actual traffic measurements are made under the control of the emulator.

Emulators lie between simulators and live systems. They allow experiments with a high degree of reproducibility. For example, an emulator might duplicate or approximate the behaviour of an attached network device. A router emulator for instance might drop packets or inject traffic into the test network or be used to stress test the network.



The NIST Network emulation tool (NIST Net) is a general purpose tool that can be used to emulate the dynamics of an IP network. NIST Net allows you to use a LINUX system configured as a router to emulate a number of different scenarios. For example, you can programme fixed and variable packet delays and random re-ordering of packets, packets can be dropped, duplicated randomly or based on congestion. You can also programme in jitter (variable delay) to do quality of service measurements – useful for VoIP system testing. Recent years have seen the introduction of other Emulators such as ITrinergy Network Emulator for Windows.



**Traffic Generation**

Before you can emulate network traffic, you need to learn how information flows on your network. Monitoring your network and collecting traffic requires special tools that captures data or communicates with SNMP or Remote Network Monitoring (RMON) agents. To test realistic performance of a network you need to generate large amounts of data to see how the network devices respond to the increase in load. A number of programmes and network analysers have the ability to generate traffic to test network performance. These range from custom packet generators that allow you to create packets with a great deal of control as to what is injected onto the network to load generators that are used to stress test the network and do not give much control. For example the Ping utility could be used to flood a network with ICMP packets or obtain a special tool such as IPLoad to generate UDP packets or use an LAN analyzer such as Observer to generate generic broadcasts or address specific packets to stress test specific devices. The Microsoft RoboClient and RoboServer are useful script based tools for testing application servers. Ixia are a company that specialise in this area - their Chariot system is now widely used. Some network oriented benchmark programmes also have packet generator components.

### Benchmarks

Benchmarks are values that are used to measure the performance of products such as processors, video cards, disc drives, applications or whole systems. They are popular performance indicators that allow companies to compare themselves with the competition. They are used by sales and marketing to promote products but the real purpose of benchmarks is to indicate the level of performance you can expect when using the product, for instance Figure 3 shows benchmark of the performance of a business computer running different OS's.

Most bench mark programmes are provided by the vendors themselves but they can originate from a variety of other sources as well such as magazines or benchmark organizations. Ziff-Davis WinBench suite (inc. Winstone standard workloads) or SySoft’s Sandra for instance provide comprehensive tests of all aspects of a computer system and Veritest’s NetBench is designed to test client/server performance of file servers and WebBench for web servers.

Figure 3 Windows operating systems as measured by ZD Labs’ Business Winstone.

NetBench consists of a server and client programme that allows loading tests to be performed over varying lengths of time. Scripts are supplied for typical traffic conditions or custom scripts created to match particular systems. See Figure 4 for typical results for a NetBench test.

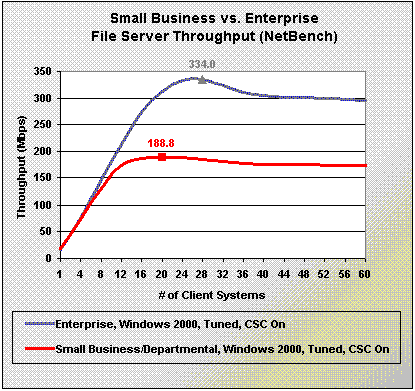


Figure 4 File server tests with NetBench

N.B. *Benchmarks can be of great value in your decision making process, but they should not be your only source of evaluating performance. When consulting benchmark results for capacity planning treat them as guidelines only and use care in their interpretation.*