

Study of Modelling Tools for Analysis of Wireless Network

Pranav¹, Priyanshu Kuma², Sarika Singh³

College of Computing Science and Information Technology

Teerthanker Mahaveer University

Delhi Road, Moradabad, India

¹*pranavsharma111@gmail.com*

²*ambujkumar709@gmail.com*

³*chaudharysarika123@gmail.com*

Abstract— Computer network technology growing day by day. To study about computer network many users and companies has been developed the different modelling and simulation tools. There are many wireless network simulators but it is difficult to decide which simulator to choose. These modelling and simulation tools are used for education and reserch. This paper shows a classification of the tools used in networks. Modeling and simulation methods are employed by scientists and engineers to gain insight into system behavior that can lead to faster product time-to-market and more robust designs.

I. INTRODUCTION

Network is a mix of applications, protocols, technologies, routing algorithm. Network design is a challenging task, requiring designers to balance user performance expectations with costs and capacities. One of the obvious approaches to deal with complexity is the use of modelling and simulation (MS) techniques. There are currently available various MS tools created by the separate companies and groups of researchers which are intended for use as practical and educational tools for network design. There is a need to study the existing network MS tools, their functionality, advantages and function procedure and special features. A computer network, or data network, is a digital telecommunications network which allows nodes to share resources. In computer networks, computing devices exchange data with each other using connections between nodes (data links). These data links are established over cable media such as wires or optic cables, or wireless media such as WiFi. Modeling and simulation (M

& S) tools can be quite powerful in gaining insight into the behavior of a complex system. Generally, a network system can be represented by a model implemented in hardware, software, or a combination of both. A simulation represents the execution of that model, consisting of a typical set of inputs, algorithms, and routines that model the system behavior and a set of outputs that provide insight into the system performance.

This purpose of this paper is to review these MS tools so that researchers can select the right tools for their experiments. Different tools are analysed in this paper and according to their characteristics they are classified.

II. CLASSIFICATION OF THE NETWORK MS TOOLS

There are different types of network MS tools available and they are used in many ways. According to the uses and availability (we analysed 100 tools), the network tools can be classified into four groups: analytical tools, simulation tools, topology discovery tools and topology generation tools. The potential candidate tools are considered and selected according to some criteria. The primary criterion is that the tools must have text-formatted input and/or output file(s) so that the information (stored data) of the model can be retrieved. The secondary criterion is that the tools have to have rich networking features (e.g. supported description of protocols and applications,

various types of networks (LAN/MAN/WAN); then the usefulness of the tool, i.e., the output of the tools and its analysis capabilities. Finally, the availability of the tools in the market has been considered.

Analytical tools help to design a network model and calculate different factors (e.g. reliability, utilisation) of the model. On the contrary, simulation tools simulate dynamic behaviors (e.g. packet passing, link failure, TCP protocol) of a network model besides modelling. The topology generation tools help to generate small as well as large topologies based on different algorithms. Finally, the topology discovery tools extract the actual network information from an existing system and map them graphically and/or in text format. This type of tool is used for network management purposes. Quantitative network design tools produce more accurate and defensive results than qualitative methods. On the other hand, analytical tools may design and analyse the network models more efficiently and provide solutions more quickly than a simulation tool but do not always achieve the required accuracy. Most often, the dynamic behaviour of the protocol cannot be visualised using the analytical tools but can be visualised using simulation. The topology which is generated or discovered by a specialised tool can be visualised and analysed by simulation and analytical tools.

Both analytical tool and topology discovery tool can be classified into two: educational (free) and commercial tools. Network simulation tools can be divided into three: educational, commercial and specialised tool. The Fig. 1 shows the classification of the tools.

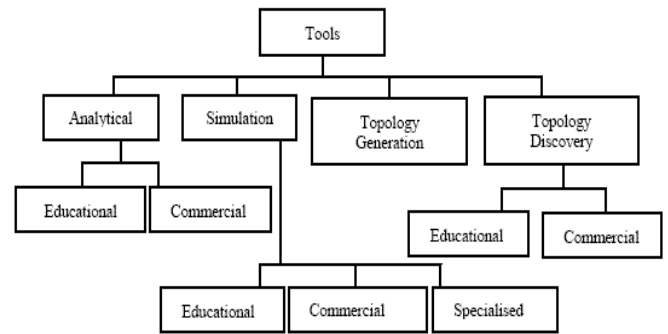


Fig. 1: Classification of the Network Design and Simulation tools

III. ANALYTICAL TOOLS FOR NETWORK DESIGN

The analytical tool typically formulates the network planning problems as optimisation models where given cost functions are minimised (maximised) under a set of constraints [7]. Normally a single model tries to capture many relevant aspects of the problem. When an optimal solution to the network model is found, the values of the decision variables can be used to decide the optimal action to be taken.

A. Educational Tools

DEsign tool LITE, Delite [8] is an educational and practical wide area network (WAN) design tool, which can produce network designs of limited size using a set of the embedded network design algorithms. Delite can produce graphical displays representing network nodes and links as well as some additional analysis data (delay analysis, reliability analysis, average delay analysis of the link, average number of hops, link utilisation for every separate link between nodes, utilisation of each node, overall network model utilisation, etc.). There are seven files that are handled by the Delite tool for each network design. However, the most important for the users are .gen (original node information, coordinates and available link types) and .net (additionally has table of links between nodes i.e. actual design) files. Links between nodes can be generated using a few design algorithms. Thus, various designs may have different costs, delays, reliability, average number of hops etc.

Other files are used for different purposes: noeip.tbl and param.tbl are predefined (used to supply some network parameters), .cst file describes the costs associated with link types (e.g. T1, T3, D96 etc.), .req file describes capacity of each link type, .inp file contains the names of files related to a particular model. The users can edit all of the files mentioned above as they are in ASCII text format. The algorithms used for designing the topology of a network model supported by Delite are: Prim, Primdijkstra, Tour (Nearest Neighbour), Tour (Farthest Neighbour), Esau-Williams, Sharma, Multispeed Tree Design, Nearest Neighbour-Esau_Williams, Multicenter Esau_William, Mentor, Incrementour. Contributor: Robert S. Cahn. Supports: different types of network design algorithms (mentioned above). Languages: C. Advantages: 1) Text formatted input and output files so user can manually change the input files to see the effects; 2) Population is taking part in designing the models; 3) World map can be included in the model to see the suitability of the model; 4) can include more algorithms in this tool. *Limitation*: 1) limited number of nodes supported as this is a small scale network tool; 2) documentatoin is no rich enough.

Cappuccino [3] is a web based non-commercial network design tool. Using the tool, users can design network model, calculate maximum traffic of each link and a lower bound of the link cost of the network. The switches can be scattered randomly inside the rectangular area (0, 0, 400, 400). The properties of the switches are expressed by switch name, X coordinate, Y coordinate, Alpha (originating capacity for a switch) and Omega (termination capacity for a switch), which can be changed by the users. The Alpha denotes the maximum communication rate, which is permitted by the network to originate from switch to the rest of the network. The Omega denotes the maximum communication rate, which is permitted by the network to terminate to a switch from the rest of the network. Traffic between switches is limited by using set-pair-constraints, which specifies the maximum traffic between two switch sets. While

designing a network model, initially 20 switches are scattered in a region. User can add, delete and modify the switches as required. Topology is chosen using predefined algorithms implemented in the tool or manually. Then traffic constraint, which specifies maximum traffic between two sets of switches, is chosen. The cost coefficient is proportional to the link length and a link has two-bandwidth capacity (incoming and outgoing). The tool uses shortest path routing algorithm for traffic. The algorithms for network topology implemented in the tool are: best star, minimum panning tree (Krustral's algorithm), delaunay triangulation and complete graph. Using shortest path routing and taking account the constraints, the tool calculates the maximum traffic and a lower bound cost on every link so that user can see how far away the network is from the optimal. The lower bound cost of the link is calculated based on complete graph. Users can also analyse the traffic of each link, switch and traffic constraints of each using window menu. User can edit the switch information and update the calculation. The tool is very user friendly. It is platform independent. This tool will help network users/designers create their non-blocking network model. *Languages*: Java. *Advantages*: This is web based. *Limitation*: Very limited applications.

B. Commercial Tools

XNetMod [10, 11] (based on NetMod tool), a LAN modelling tool that allows users to analyse the performance of a configuration before implementing it. Three approaches to performance evaluation of this tool are analysis, simulation and measurements. XNetMod can be used networking environment consisting of thousands of computers sites. XNetMod provides customised forms to enable one to specify input parameters for the elements. XNetMod supports two types of topology: ring and bus and two connectors: Router and Bridge. To facilitate the concept of bottom-up design, XNetMod allows users to create subnetworks. At present there is only one element type, Subnet in the Subnetwork element group. XNetMod supports

analytic and simulation techniques. It can summarise the entire results of an analysis or simulation of just the result for a particular node. *Supports:* FDDI, token ring, token Bus, generic bridge, generic router, subnetting. *Languages:* C. *Advantages:* 1) graphically display and manipulate network topology; 2) ease of subnetwork definition; 3) three types of analysis of data are supported; 4) flexibility of easily add or delete subnetworks; 5) ability to provide substantial user interaction for either simple parameter change of major reconfiguration of LANs and backbone; 6) hierarchical modelling capability for extremely large models; 7) query and generate reports based on currently displayed model.

TND-Tool (Topological Network Design Tool) [12] is used to find out an effective solution to the network topology design problem. The idea behind developing the TND-Tool are twofold: first providing network designer and operator with a tool to facilitate their work, second offering topology design researchers an environment for testing and representing the output of their network design algorithms. It permits to select between node types, link types and data bases for different kinds of networks. A Network viewer allows to display the current topological network structure during the optimisation process. Users, engineers are able to set up a network interactively by inserting nodes and links, typing in network information, e.g. traffic values, moving, adding and removing some part of the network. According to the tariffs defined, the TND-Tool calculates the cost for an existing network on the screen or a network resulting from an optimisation algorithm. The TND-Tool consists of four major modules: 1) representation module is the graphical user interface of the system providing all important facilities needed by the network engineer; 2) optimisation module enables integration of network optimisation algorithms; 3) cost calculation module is the entity providing the rules for calculation of network cost, e.g. tariff tables and cost functions; 4) data base module provides all the needed data for the functionality of the system. Currently, the TND-Tool is used for

designing GSM (Global System for Mobile communication) and DCS1800 (Digital Cellular System 1800) fixed networks as part of the cellular mobile communication networks. The TND-Tool is not limited to cellular networks and has been designed as a platform for designing all kinds of networks such as Wireless Local Loop, Satellite networks, ATM networks and so on. *Supports:* WAN, Wireless, ATN, Satellite. *Advantages:* 1) an interactive menu and mouse-driven tool; 2) Web based. *Limitation:* 1) Limited only to calculate the cost of design; 2) Limited number of topology design algorithm supported.

NetRule [13] is designed to combine the best features of the mathematical analysis and simulation tool. It is mainly performance prediction tool to design the actual network. NetRule can model and stress test very large networks (1,000 to 10,000 nodes). It has a customised GUI like a simulator. It uses closed-form mathematical analysis that evaluates network load and performance in seconds with detail and accuracy that matches the best practice of mathematical analysis. The tool lets users to create or open network models and libraries, edit them graphically, and evaluate their performance. The diagram can animate message flow, highlight bottlenecks and shows utilisation and delays bars as components. It uses shared objects definitions to reduce modelling time and allow global changes. The NetRule library includes: 1) LANs, including many CSMA/CD, token ring and polling protocols; 2) Frame relay clouds with access ports and virtual circuits; 3) ATM switches and SONET links; 4) Routers, hubs, bridges and Ethernet switches; 5) Point-to-point T1 and other speeds; and 6) FDDI; 7) ISDN. Reports (output) are available for object definitions as well as for evaluation statistics that can be sorted or can be exported to spreadsheet, database, presentation or graphics packages. The most interesting features of the NetRule are period objects, which let one to establish different workloads during different times of day (day and night period) or time of years. Histogram can be produced for performance analysis. Normally the report includes: computer

delay/job, computer utilisation, job cost, job delay, link delay/pkt, link utilisation. Another important feature of NetRule is that it can import network topology that has been discovered with a network management tool. *Contributor:* Analytical Engines Inc. *Supports:* CSMA/CD, Frame relay, ISDN, FDDI, ATM, SONET, point to point T1 connection. *Languages:* Java. *Advantages:* 1) supports very large scale network (1,000 to 10,000 nodes); 2) user can import ALL or any part of a network and its parameters to build or modify a NetRule model. E.g .cap files, .csv format file, MIB data; 3) Supports generalisation option for clients, servers, and/or applications (for example, mathematically determining the typical client); 4) it can extract from traffic files and make a new file for all computers which can potentially be servers, all applications (using port numbers), user traffic flow.

XNP (Extensible Network Planner) [14] is a comprehensive network design tool for both extensible and conventional networks. It is implemented using several methods and incorporated them into a previously unpublished java-based software tool, Cappuccino. Its objective is to configure a least-cost network that accommodates any traffic pattern satisfying the given traffic constraints. XNP allows network designers to quickly create, configure and evaluate network designs by providing a convenient graphic-based interface and automated functions. There are five essential building blocks in designing a network in XNP. Users can create a trail topology in the blank drawing area of the XNP main window. XNP allows two types of links (unidirectional and bidirectional). Application format (number of processing steps, number of machine instructions and bandwidth required) can be added or removed by user-friendly buttons. XNP provides three ways to specify traffic expectations. XNP process will iterate through all links and processing nodes to compute the capabilities for them. XNP will redraw the topology to show the differences in resources capabilities again and again. The thickness of each link shows its capability relative to other link's

capabilities. Users can evaluate the network configuration by computing a lower bound on the cost of the best network configuration. By adjusting options of lower bound and therefore by considering design restrictions for using resource again and again, one can obtain more accurate and useful measure. XNP allows three different ways to view the drawing area-only the trail topology and the design space together or the design space only. XNP also allows to manage multiple designs at a time so than one can try for different trial topologies and compare them. *Supports:* supported algorithms are: 1) Complete network; 2) Delaunay triangulation; 3) Delaunay triangulation with trimmed links; 4) Link complement; 4) Minimum spanning tree; 5) Random link adder; 6) Random node adder; 7) Star network; and 8) Symmetric link adder; 9) Geographical planar projection- longitude and latitude; 10) Grid network – N x M grid network; 11) Torus network- N x M torus network. 12) Localised traffic constraints; 13) Proportioned Pairwise traffic constraints. *Languages:* Java. *Advantages:* 1) Extensible.

IV. NETWORK SIMULATION TOOLS

Simulation is the discipline of designing a model of an actual or theoretical or physical system and manipulating the model in such a way that it operates on time or place to compress it, thus enabling one to practice the interaction. Many unimportant details can be abstracted away and simulations can be completely repeatable. In a simulation, a mathematical/logical model is numerically evaluated over the period of interest and performance measures are estimated from model-generated data.

A. Educational Tool

ns-2 (Network Simulator) [15, 16, 17, 18], a VINT (Virtual Inter-Network Testbed) project from U.C.

Table 1: Summary of network analytical tools
Berkeley/LBL/Xerox PARC, is a discrete

Tools	Usages	Design Algorithm(s)	Target Network Model	Outputs	Stored Data File	Platform
Delite	Educational	Some predefined algorithm	WAN	Graphical Link analysis, node utilisation, overall utilisation in percent, delay analysis	Special formatted Text file	Windows
Cappuccino	Educational	Predefined Design algorithms	WAN	Link cost, capacity etc	Graphical output, no text file	Windows Web based
XNetMod	Commercial	User defined topology	LAN, Subnetwork	Delay, average packet queue length, packet delay time, utilisation,	Graphical output, No text file	Windows Unix
TND-Tool	Commercial	User defined topology	WAN, Wireless, ATM, Satellite	Analysis of Cost of deigned network	Graphical output. No Text file	Windows
NetRule	Commercial	User defined topology	WAN, LAN, ATM switches, ISDN, FDDI	Computer Delay, Computer Utilisation, Jobs cost, Job delay, Link Delay, Link Utilisation, histogram representation	Report can be imported to spreadsheet, database	Windows
XNP	Commercial	Predefined Design algorithms	WAN	Resource capacities, Cost analysis of links	XNP formatted text file	Windows Unix

event simulator targeted at network research, which provides substantial support for simulation of TCP, routing and multicast protocols. The simulator is using a Tcl/Otcl (Tool Command Language/Object Oriented Tcl) as a command and configuration interface. There are four types of files related to ns-2 simulator. Model is described in files .tcl or .ns which have common subset of commands but not exactly compatible between each other's. While simulator runs a model defined in .tcl/.ns file, simulation trace file (.tr) and animation file (.nam) are created during the session. Network Animator (.nam) files are used to visualise the behaviour of the network protocols and traffic of the model. Once created, users can play with the .nam file just like a media player and check the behaviour repeatedly. ns-2 facilitates the three broad themes of network research simulations: 1) selecting a mechanism from several options; 2) exploring complex behaviour; and 3) investigating unforeseen multiple-protocol interaction. *Supports:* TCP family, UDP, CBR, FTP, HTTP, Pareto, Exponential protocols, wires, wireless, unicast,

multicast. *Languages:* both C++ and Otcl languages. *Outputs:* dynamic output. *Advantages:* 1) user can design model both manually or writing code; 2) easy configurable and fast simulation by using two different languages (Otcl and C++); 3) many protocol already implemented; 5) dynamic behaviour can be visualised using nam editor; 6) Open source code and can be extended for future. *Limitation:* 1) Long time to get used to usign it; 2) does not large scale simuaton (e.g., internet). 3) incomplete API; 4) no real time simulation; 5) no performance analysis is possible 6) badly documented source code; 7) no direct support of mobility and shared wireless radio channels; 8) unable to scale to networks of Internet size due to the computational requiremets of fine-grain packetlevel simulation and memory needed to maintain queues at each network link.

Network Workbench [19] is a discrete event network simulator developed for the academic investigation of Internet protocol. It contains a complete protocol stack, abstracted from the Internet stack and a set of exercise that focus on critical protocol algorithms in the Internet stack.

Network Workbench has several version, which are: version 0(1994), version 1(1994), version 2(1997), and version 3(1998). This tool supports several network topology, DLC error control, CSMA/CD collision backoff, optimal route computation, reliable transport, multicast, and LAN/MAN integration. It supports FDDI (Fiber Distributed Data Interface), OSPF (Open Shortest-Path-First). Workbench is abstracted from the internet (TCP/IP) stack. It contains five layers (application, transport, network, datalink control, and physical). The Network Workbench tool can be used to study of WAN topology, DLC (datalink control layer) frame formatting, DLC flow and error control, CSMA/CD local area network, network layer routing, reliable transport, multicast networking etc. Summary of statistics of the performance is produced at the end of the simulation. *Contributor:* J. Mark Pullen. *Supports:* network topology, DLC error control, CSMA/CD collision backoff, optimal route computation, reliable transport, multicast, and LAN/MAN integration, OSPF, FDDI, Ethernet, TCP/IP. *Languages:* C++. *Limitation:* Very limited implementation of protocols.

Netsim (M.I.T.'s Network Simulator) [20, 21] is a single process discrete event simulator used for the investigation of many aspects of Local Area Network (LAN). *Netsim* has three main goals: 1) flexibility of experiment of specifying the network and traffic; 2) simulate the accurate behaviour of Ethernet; and 3) has features that make running sequences of related experiments easier. The experimental data of *Netsim* simulation is stored in an experiment description file which contains: the layout of the network to be simulated, the traffic generation behaviour of the station on the network, information about repetitive of each run and about the sequence of parameters to be used for a series of experiments, etc. The entries specify the physical characteristics (e.g., length, packet size, data rate, number of attached stations) and traffic generation behaviour of the attached station. There are five types of traffic distributions in *Netsim*: exponential, uniform, deterministic, continuously queued and user defined discrete distribution (e.g. biomodal).

Netsim collects information on many aspects of the operation of the simulated network. For example, average packet delay, histogram of packet queuing delays and collisions per packet transmitted, overall throughput (percentage utilisation), actual data rate, average, queuing delay, variance of queuing delay, observed packet transmission rate, average message size, total observed collision rate and total number of packet, maximum packet size, minimum packet size, maximum data rate etc. *Contributor:* MIT LCS Advanced Network Architecture group. *Supports:* Ethernet link, a point-to-point link, a switch (switches packets between several links), a host (about the same as a switch), Purdue's implementation of TCP, data supplier and consumer from TCP, a simple Poisson traffic source and a packet sink. *Languages:* C. *Advantages:* 1) *Netsim* is a publicly available ATM simulator originally developed at MIT; 2) its source code is freely available and modifiable; 3) *Netsim* has a user friendly Graphical Interface (GUI). *Limitation:* 1) The GUI however is very primitive and is inadequate for instructional purposes. It is not fault-tolerant to novice users. An incorrect sequence of keys or mouse clicks could cause the simulation to crash; 2) Its uses are limited.

MaRS (Maryland Routing Simulator) [22, 23] is a discreteevent simulator proving a flexible platform for the evaluation and comparison of network routing algorithms. *MaRS* allows physical network, routing algorithm and traffic sources. *MaRS* is structured in two parts: a simulation engine, which manages the event list and user interface and a set of components for modelling the network configuration and handling certain simulation functions. *MaRS* has been used to evaluate and compare several next-hop routing algorithms: two distance-vector algorithms- a) Loop-free Bellman-Ford Routing Protocol Without Bouncing Effect, a Failsafe Distributed routing Protocol; b) a link-state algorithm (the New Routing Algorithm) for ARPANET. *MaRS* is limited due to the lack of scrolling capability in the user interface window. Thus, users are limited in what can be displayed to the size of the non-scrolling window. In addition,

MaRS constrains itself to a point-to-point network structure and therefore is not readily suitable for simulating networks with broadcast communication. Contributor: University of Maryland. Supports: network routing protocols (SPF, Merlin_Segall, Bellman_Ford), FTP, Telnet, Workload (Poison, uniform). Languages: C. Limitation: 1) Its support for transport layer protocol and application source models is very limited; 2) a primitive GUI is provided; 3) not available now.

pdns (Parallel/Distributed ns) [24] simulator consists of extensions to the widely used and publicly available ns network simulator. The extensions allow many existing ns simulations to be run in a distributed environment with minimal changes. The *pdns* implementation also takes advantages of the large body of existing models found in ns and uses those without modification. Also included in the *pdns* extensions is novel packet routing method called NIXvector that allow routing decisions without the necessity of routing tables, resulting in substantial memory saving. While distributing the ns tool to several processors, for connecting each sub-model, IP address and a network mask is added to physical end point (nodes) for communication in logic as well as physical connection. For routing, each simulated node will start with routine table priori and dynamic routing information in the simulation time using Boarder Gateway Protocol (BGP) to adapt to any change in the simulated network topology. For event time management and event distribution among the parallel simulation sub-models, RTIKIY and LBTS (lower bound time-stamp) etc. time management run-time library are used. For synchronous event communication, multicast group management (MCAST) strategy is used. Contributor: The PADS research group at Georgia Institute of Technology. Advantages: 1) Very largescale simulator (hundreds of thousands of nodes); 2) IP Addresses are supported.

B. Commercial Tools

OPNET (Optimised Network Engineering tool) [19, 20, 25] was launched in 1987 as a first commercial available simulation tool for communication networks. It provides a comprehensive development environment for the specification, simulation and performance analysis of communication network. It can simulate all kind of wired networks and a 802.11 compliant MAC layer implementation is also provided. A large number of communication systems from a single LAN to global satellite networks can be supported. The most important features of *OPNET* are: modelling and simulation cycle (to assist user to go through three phases in design circle-building model, execution of simulation and the analysis of output), hierarchical modelling (describes different aspects of the complete model being simulated), specialised in communication networks (support for existing protocols and allow users to either modify these existing models or develop new models), automatic simulation generation. An *OPNET* model consists of three layers: the network models, the node model and the process model. The tool allows the different layers in a protocol stack running on individual nodes to be represented. *OPNET* contains various tools for data collection. A probe editor allows specifying: 1) which statistics are to be collected from where (Probe editor); 2) their own statistics (analyse tool); 3) animation view and formats (Filter Tool, Animation viewer). *OPNET* contains objects that are capable of generation vast amount of output data during simulation. It can generate error rate and throughputs, delay queue size. Packet trace may be done. Output can be plotted in graph, such as end-to-end delay vs. queue buffer capacity, loss ratio vs. queue buffer capacity. Probability distribution function, cumulative distribution function as well as histogram can be plotted for several data sets. *OPNET* is extensively used for the study of TCP transport across different types of ATM bearer capabilities and diffserv per hop behaviour. Contributor: Mil 3 Inc. Supports: routing protocols (OSPF, RIP, EIGRP, BGP, IGRP, DSR, TORA IS-IS, PNNI), Diffserv, MAC,

mobility of nodes, ad hoc connectivity, different application models. Node failure models, modelling of power-consumption etc. *Languages:* C, Java. *Advantages:* 1) extendible; 2) large customer base; 3) professional support; 4) very well documented; 5) ships with a large number of built-in protocols. *Limitation:* 1) relatively high price; 2) complex, takes time to learn; 3) there is restriction to its portability.

COMNET III [26, 27, 28] is a commercial integrated discrete event object-oriented simulator for modelling and performance analysis of computer network. With COMNET III, users can create a variety of network architectures, including LANs, MANs WANs, packet switching, ATM, frame relay and so on. COMNET is entirely driven by graphical user interface. Users are able to graphically display traffic modelling patterns, use of realistic network objects to reflect real networks and apply the network concepts. COMNET III allows the users to model, tune and analyse the performance of various types of networks. It provides an extensive library of nodes, links, protocols and traffic objects. The library of link objects include two classes: point to point links and multi-access links. Multi-access protocols that can be modelled include CSMA/CD, CSMA, ALOHA, Token BUS, Token Ring, FDDI and Polling. It supports subnets. It has two kinds of sources: application sources and traffic sources. There are four types of traffic generators: message sources, session sources, response sources and call sources. COMNET III is able to produce about 100 reports for different model building blocks. Most commonly produced reports include node utilisation and application delays, link delays, channel utilisation, message delays, packet delays, calls blocked, disconnected and preempted, session set up delay, collision statistics, token ring statistics, buffer statistics etc. *Contributor:* CACI Company. *Supports:* LAN (Ethernet, Token ring, FDDI), MAN, internet, packet switch, circuit switch, ATM, bursty traffic, CSMA, CSMA/CD, ALOHA, Polling, Protocols (TCP/IP, IPX, SNA, DECNet). *Advantages:* 1) a graphical package, allow quickly

and easily analyse and predict the performance of network; 2) allows users the flexibility to try an unlimited number of "what if" scenarios; 3) realistic and accurate results. *Limitation:* 1) source code is not available; 2) new modules were very difficult to add; 3) It is restricted to experimenting with the set of networking protocols provided by the package; 4) It is not programmable by the users.

REAL (REalistic And Large) [29] is a network simulator originally intended for studying the dynamic behaviour of flow and congestion control schemes in packet-switched data networks. It provides around 30 modules that exactly emulate the actions of several well-known flow control protocols (such as TCP), and 5 research scheduling disciplines (such as Fair Queuing and Hierarchical Round Robin). There are nearly 30 source types, corresponding to 30 or so transport protocol and workload types. The sources can be categorised into one of two types: flow-controlled and non-flowcontrolled data sources. REAL has a graphical user interface (GUI) written in Java. The simulator takes as input a network scenario (a description of network topology, protocols, workload and control parameters) described in NetLanguage (ASCII). It supports sources or sinks node, gateways (synonymously with routers, bridges and switches). Users have to specify some network parameters, the transport protocol (in particular, the flow control) and the workload at each source. Finally, users must specify control parameters such as the latency and bandwidth of each communication line, the size of trunk board buffers, packet sizes etc. *Contributor:* S. Keshav at Cornell University. *Supports:* TCP/IP, XNS, FTP, Telnet, ill behaved, FIFO, FCFS, Fair Queuing (FQ), DEC congestion avoidance and Hierarchical Round Robin. *Languages:* C, JAVA (GUI only). *Advantages:* 1) the GUI allows users to quickly build simulation scenarios with a point-and-click interface; 2) source code is provided so that interested users can modify the simulator to their own purposes; 3) extendible. *Limitation:* 1) NEST (initial tool) didn't not allow for timers, REAL sends out a timer packet from a source back to itself

to return after some specified time, but timers cannot be reset using this method; 2) misses some of the behaviours in the common protocol implementation; 3) do not support common implementation of protocols (direct execution). Instead it uses codes that simulates the major characteristics of the protocols.

SSF (Scalable Simulation Framework) [30, 31] is designed to model very large-scale networks, which are described in generic modelling language called Domain Modelling language (DML). *SSF* can be run in a distributed environment (called *DaSSE*) on a tightly coupled shared-memory symmetric multiprocessor. *DaSSF* achieves good parallel performance by using a periodic time-stepped approach. All processors can safely process messages between synchronisation cycles without fear of erroneous results due to unsafe events. *SSF* supports multicast in-channel (many to one communication) as well as multicast out-channel (one to many) and bus-style channel mapping (many to many). *SSFNET* is the first collection of SF-based models for simulating Internet protocol and networks. The *SSFNET* libraries include component models for network elements (hosts, routers, network interface card, local area networks) and network protocols (currently IP, UDP, TCP, BGP and OSPF). *SSFNET* models are self-configuring, that is each *SSFNET* class instance can autonomously configure itself. *SSF* has been demonstrated on networks of several hundred thousand nodes. *Contributor*: Renesys Corporation. *Supports*: TCP, UDP, BGP protocols. *Languages*: Java, C++. *Outputs*: packet-by-packet simulation. *Advantages*: 1) Open source code; 2) Free for education; 3) Platform independent; 4) This has open source code and based on OOP, it can be enhanced in future. *Limitation*: Dynamic simulation cannot be visualised using this tool.

TeD (Telecommunication Description Language) [32, 33, 34] is a language for describing telecommunications networks, coupled with an optimistic network simulation engine, based on Georgia Tech Time-warp. The *TeD* language

specification is split into two distinct parts- *MetaTeD* and “External Language”. *MetaTeD* defines a set of concepts for modelling the dynamic interactions of entities and their compositions. When *MetaTeD* is appropriately combined with any regular general-purpose programming language (say C++) then complete language is formed. *TeD* is process-oriented where the number of processes easily exceeds one million, warranting efficient support for largescale process orientation. It has demonstrated good performance and scalability when modelling ATM cell switches and private network-network interface (PNNI), Internet and wireless network. *TeD* has been demonstrated on network models consisting of tens of thousands of nodes. Conservative and Optimistic synchronous processes have been implemented in *TeD*. *Contributor*: Kalyan Perumalla, Andrew Ogielski, Richard Fujimoto at Georgia Tech. *Year*: 1996. *Supports*: TCP/IP, ATM Private Network to Network Interface (PNNI) signaling networks, Multicasting protocols, wireless networks. *Languages*: C++ and Meta-language (*MetaTed*). *Outputs*: Simulation of models. *Advantages*: 1) *TeD* itself is independent of the underlying parallel simulator and can be used with other parallel simulator (e.g. *Nops*, recently developed at Darmouth); 2) *TeD* achieves high parallel performance on multiprocessor machine, speeding up the simulation by a factor proportional to N for N-processor machine. *Limitation*: 1) Limited protocols are implemented; 2) Not freely available.

USSF (Ultra-Large Simulation Framework) [35, 36] simulator is based on the *WARPED* simulation engine. *WARPED* is a parallel discrete event optimistic simulator based on Time Warp. The syntax and semantic of the input topology models for *USSF* are described in a Topology Specification Language (TSL). Using TSL, large topologies can be built from smaller sub-topologies and sub-sub topologies. The topology is parsed into an Intermediate Format (TSL-IF). The analysed TSL-IF by the help of static analyser and code generator modules is then used to generate an optimal simulatable network topology. *USSF* is used to

model large networks using a network of Dual-CPU Pentium processors connected by an Ethernet network. Decoupling of the data and states are provided for swapping data and states in and out of the main memory based on demand, freeing up the memory for better performance of the simulation. USSF has been demonstrated on network models of hundreds of thousands of nodes. The current implementation of USSF is in C++ language in UNIX system. *Contributor:* Dhananjai Madhava Rao, Philip. A. Wilsey. *Supports:* LAN, MAN, Languages: C++. *Advantages:* Very large-scale simulator (hundreds of thousands of nodes).

C. Specialised Tools

ATM-TN (ATM Traffic and Network simulator) [37] is designed to characterise cell level network behaviour. The simulator incorporates three classes of ATM traffic source models: an aggregate Ethernet model, an MPEG model, World Wide Web transaction model and six classes of ATMswitch architectures including output buffered, shared memory buffered and cross bar switch models. The ATM-TN simulator can be used to characterise arbitrary ATM networks with dynamic multimedia traffic loads. Call set up and tear down via ATM signalling is implemented in addition to the various types of cell traffic streams generated by voice, video and data. The simulator is built on a simple, efficient simulation language called SimKit, which is capable of supporting both fast sequential and parallel execution. Parallel execution is supported using WarpKit, an optimistically synchronised kernel that is aimed at shared memory multiprocessor platforms. The main design principles of the ATM-TN are: 1) accurately mimic ATM network behaviour at the cell level for specific traffic loads; 2) create a modular extensible architecture; and 3) achieve responsible execution times for ATM networks that consist of hundreds of traffic sources. The structure of ATM-TN consists of the components: traffic models, switch models, an ATM Modelling Framework (MF), SimKit, WrapKit, OSS and the Telecom Modelling

Framework (TMF). Several distinct components are to construct traffic models: input traffic sources (e.g. FTP, Telnet, Mosaic, JPEG, MPEG), ATM Adaptation Layer (AAL) for converting source data to cell packets, access control mechanisms (e.g. leaky bucket). There are three types of traffic models are used: MPEG, Ethernet and WWW. The basic switch models in ATM-TN simulator have two components: the control module and the switch fabric. Besides dynamic behaviour of ATM network, analytical outputs can be estimated which include: propagation delay, link capacity in bit/sec, error rate, work load and several parameters. *Contributor:* Telesim project led by Brian Unger at University of Calgary. *Supports:* ATM, Data sources (FTP, Telnet, Mosaic, TCP/IP), Video sources (JPEG, MPEG, video-conferencing), WWW model, Qnet model. Languages: C++. *Outputs:* Link utilisation, QoS, delay analysis. *Advantages:* 1) Overhead is low; 2) High performance and realistic model; 3) The amount of computation associated with processing such events is very low; 4) ATM-TN has a GUI that provides easy simulation scenario configuration, data set organisation, and control over simulation execution and report generation. *Limitation:* At present, workload is balanced manually.

Glomosim (Global Mobile System simulator) [17, 38, 39,40] is a scalable simulation environment for wireless mobile network systems. It is designed using the parallel discreteevent simulation capability provided by PARSEC. The protocol stack includes: models for the channel, radio, MAC, network, transport, and higher layers. It supports TCP, IEEE 802.11 CSMA/CA, MAC, UDP, HTTP, FTP, CBR, Fishleye, LARScheme-1, ODMRP, WRP, DSR, MACA, Telnet, AODV, etc. protocols. There is a visualisation tool VT designed to view and help debugging the protocols. Glomosim currently supports protocols for only wireless network. In the future, it is planned to add functionality to simulate a wired as well as a hybrid network. There are eight files related to a network model designed in Glomosim network simulator. Six input files are used to execute a network model:

Configuration file, Nodes file, Mobility file (movement of the nodes and traffic), Routers file, Application file, Ber_bpsk file (Bit Error Rate). Two files are related to the output of the network model. All of the statistics (selected simulation events) is compiled together into a file called glomo.stat that is produced at the end of the simulation. Trace file is produced (if 'write trace' option is chosen) which can be played as many times as user wants and this is faster than real time. Basically, Glomo.stat and trace files store same set of data. But trace file can be simulated any time, which is not possible with Glomo.stat. Supports: TCP family, UDP, CBR, FTP, HTTP protocols, wireless. Languages: Java, C. Outputs: Dynamic output in VT tool, trace file. Advantages: 1) Easy configurable and fast simulation; 2) Many protocol already implemented; 3) Well documented; 4) Dynamic behaviour can be visualise using VT tool; 5) Large scale simulator; 6) Open source code; 7) can be extended for future. Limitation: 1) Currently wired network is not implemented; 2) Incomplete API.

IV. CONCLUSIONS

In this paper, a survey of the research work on network MS tools with a classification is presented. We have tried to review all the recent research work very briefly that have been brought to our attention, identified which are the main contributions that have been made, and what are the issues that seem to be open to further research.

Topology of a network or a group of networks (e.g. Internet) has a strong bearing on many management and performance issues. Investigation of the topological characteristics of computer network and its practical uses is much more accurate when right tools are applied. In this paper, a review of the research work on network topology generation and discovery tools is presented.

Our overall conclusion from this survey is that parallel, very large-scale network simulators are relatively new and still rapidly evolving field where Internet is trying to map. Existing research has certainly made significant contribution in this field. However rapid advances in simulation technology enable to execute highly interactive and dynamic, parallel simulations/animations on the network technology.

Table 2: Summary of the small and large scale network simulation tools

Tools	Usages	Target Network Model	Outputs	Stored Data Format	Platform
Ns-2	Educational, Research	WAN	Simulation of several protocols	Text File (ns/tcll format)	Unix
Network Workbench	Educational, Research	LAN,MAN	Simulation of network, summary of interlayer statistics	Text file	Windows Unix
NetSim	Educational	LAN (Ethernet)	Simulation, histogram of packet queuing delay, offered load, overall throughput (%), actual data rate, average queuing delay, variance of queuing delay, packet transmission rate, and total number of packets.	Text file	Windows
MaRS	Educational	WAN, link state and distance vector routing, SPF, ExBF, Segal routing	Simulation of application traffic-ftp, telnet, simple, various meters (binary, bar graph, histogram, line graphs, etc), periodic and event updated statistics, throughput, delay, jitter, dropped packets, routing load etc., no trace file, Hope count, utilisation, delay and hope-normalised delay.	Not known-require further investigation	Unix, Windows
OPNET	Commercial	LAN Satellite, radio modelling	Simulation of several protocols, delay, utilisation etc. performance analysis.	Text file (C/C++ format)	X Window
COMNET III	Commercial	LAN, MAN, WAN	Node utilisations, application delays, link delays and utilisation, message delays, packet delays, calls blocked, disconnected and pre-empted, session setup delays etc.	Graphical input/output. No text file for modelling	Unix
REAL	Commercial	LAN	Statistics such as the number of packets sent by each source of data, number of packet received by sink, the queuing delay at each queuing point, and the number of dropped and retransmitted packets.	Text file	Unix
ATM-TN	Specialised	LAN, MAN, WAN	Simulation of ATM network	Text file in unix system	Unix
GloMoSim	Specialised	Wireless network	Simulation of wireless network	Text file (two configuration files)	Windows Unix

REFERENCES

- [1] Bragg A.W. "Which network Design Tool Is Right for You?" IT Professional, September/October, 2000, p2331
- [2] James, D. McCabe. Network Analysis, Architecture and Design. Morgan Kaufman Publishers. San Francisco, Inc. Francisco, Ca, 2003.
- [3] Todd R. Andel, Alec Yasinsac. On the Credibility of Manet Simulations. Computer Magazine, IEEE Computer Society, July 2006, pp48-54.
- [4] Stuart Kurkowski, Tracy Camp, and Michael Colagross. MANET simulation studies: the incredibles. ACM SIGMOBILE MCCR Vol. 9, Issue 4, pp. 50-61, Oct. 2005.
- [5] Rahman M.A., A. Pakstas and F.Z. Wang. "Network Modelling and Simulation Tools", Proc. of the 8th EPSRC Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (EPSRC PGNet 2007), Liverpool John Moores University, June 28-29, 2007, Liverpool, UK.
- [6] Humayun Akhtar, "An Overview of Some Network Modeling, Simulation & Performance Analysis Tools." Proceedings of the 2nd IEEE Symposium on Computers and Communications (ISCC '97), 1997 p344-348.
- [7] Jukka K. Nurminen. Models and Algorithms for Network Planning Tools - Practical Experiences. System Analysis Laboratory Research Reports, Helsinki University of Technology, May 2003. (<http://lib.hut.fi/Diss/2003/isbn9512265745/article6.pdf>)
- [8] Cahn R.S. "Wide Area Network Design: Concepts and Tools for Optimization". Morgan Kaufman Publishers, San Francisco, 1989.
- [9] Ma H. (Bob). "Network Design Tool. Computer Science and Engineering", Washington University in St. Louis, MO., 1989. (<http://www.cs.wustl.edu/~javagr/p/networkdesign-tool.html>).
- [10] Bachmann D.W., M.E. Segal, M.D. Srinivasan, T. J. Teorey. "NetMod: A Design Tool for Large-Scale Heterogeneous Campus Network". Center for Information Technology Integration (CITI), University of Michigan. (IEEE J. on Selected Areas in Communications. January 1991, Vol 9, No 1, 1990, p1524)
- [11] Deboo K. "XNetMod: A Design Tool for Large-Scale Networks". CITI Technical Report TR-93-6. University of Michigan, 1993. (<http://www.citi.umich.edu/techreports/reports/citi-tr-936.pdf>).
- [12] Shahbaz M. "Network Design Tool: TND-Tool". ComNets Annul Report. Communication Networks, Aachen University of Technology, Germany, 1996. (<http://www.comnets.rwth-aachen.de/report96/nodel29.html>).
- [13] Post. S.D. "Network Planning with a PerformancePrediction Tool". International Journal of Network Management, Vol 9, Issue 3, 1999, p167-173.
- [14] Choi S. "Resource Configuration and Network Design in extensible Networks". PhD Thesis. Department of Computer Science and Engineering, Washington University, 2003. (<http://www.arl.wustl.edu/~syc1/web/paper/thesis.pdf>)
- [15] Ning Weng, Ramaswary Ramaswary, Tilman Wolf. Considering Processing Cost in Network Simulators. Proc. of Workshop on Models, Methods and Tools for Reproducible Network Research (MoMeTools) in conjunction with ACM SIGCOMM, Karlsruhe, Germany, 25 & 27 August 2003, p47-56.
- [16] Vern Paxson, Sally Floyd. Why We Don't Know How To Simulate the Internet. Proceedings of the 29th Winter Simulation Conference, December 1997, p1037-1044.
- [17] Bresla L., D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Ya Xu and H. Yu. "Advances in Network Simulation". The VINT Project. IEEE Computer, N5, 2000, p59-67.
- [18] Fall K. and K. Varadhan. 2003. The ns Manual. The VINT Project, 13 December 2003. (http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf)
- [19] Network Simulators, by Sally Floyd. (<http://www.icir.org/models/simulators.html>)
- [20] Network Simulation (Wikipedia) (http://en.wikipedia.org/wiki/Network_simulation)
- [21] Pullen J.M. "The Network Workbench: Network Simulation Software for Academic Investigation of Internet Concepts". Computer Networks. Elsevier Science Ltd, March 2000, Vol 3, Issue 3, 2000, p365378.
- [22] Barnett III B.L. "An Ethernet Simulator for Undergraduate Networking". ACM SIGCSE Bulletin, Vol 25, Issue 1, 1993, p145-150. (Proceedings of the 24th SIGCSE Technical Symposium on Computer Science Education. Indianapolis, Indiana, USA.)
- [23] Griffin D. "Traffic Engineering for Quality of Services in the Internet, at Large Scale". Technical Report, CEC Deliverable Number: 201/UCL/bl, TEQUILA Consortium, 2000.
- [24] Alaettinoglu C., Dussa-Zieger, I. Matta, A.U. Shankar and O. Gudmundsson. "Introducing MaRS, a Routing Testbed", ACM SIGCOMM, pp 95-96.
- [25] Riley G.F. and M.H. Ammar. "Simulation Large Networks: How Big is Big Enough?". Proceedings of IST International Conference on Grand Challenge for Modeling and Simulation, San Antonio, TX, 2002, p3945.
- [26] G. F. Riley. The Georgia Tech Network Simulator. Proc. of the ACM SIGCOMM 2003 Workshops, Germany, August 2003, p5-12.
- [27] Wireless IP Simulator ©Copyright 2005 -OSTG Open Source. Technology Group. (<http://sourceforge.net/projects/wipsim/>, 2005)
- [28] Chang X. 1999. "Network Simulations with OPNET". Proceedings of the 1999 Winter Simulation Conference, Vol 1, 1999, p307-314.
- [29] OPNET Technologies, Inc. 2004. "OPNET Modeller". (<http://www.opnet.com/products/modeler/home.html>)
- [30] Ahjua S.P. "Comnet III: A Network Simulation Laboratory Environment For A Course In Communications Networks". Frontiers in Education Conference (IEEE/ASEE conference), Tempe, AZ, 1998. (<http://fie.engr.pitt.edu/fie98/papers/1205.pdf>)
- [31] CACI Products Company. "COMNET III User's Manual". January 1995.
- [32] Goble J.G. and R. Mills. COMNET III: Object-Oriented Network Performance Prediction. Proceedings of the 1994 Winter Simulation Conference, 1994, p443-445.
- [33] Keshav S. "REAL 5.0 User Manual". Cornell University, Ithaca NY. August 1997. (<http://www.cs.cornell.edu/skeshav/real/user.html>)
- [34] Cowie J.H., D.M. Nicol and A.T. Ogieliski. "Modeling the Global Internte". Computing in Science and Engineering. Vol 1, Issue 1, 1999, p42-50.
- [35] Cowie J.H., D.M. Nicol and A.T. Ogieliski. "Toward Realistic Million-Node Internet Simulations". International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99), Las Vegas, Nevada, 1999, p2129-2135.
- [36] Bhatt S. and B. R. Fujimoto. "Parallel Simulation Techniques for Large-Scale Networks". IEEE Communication Magazine, 1998, N8, 1998, p42-49.
- [37] Perumalla K., R. Fujimoto and A. Ogielski. "TeD - A Language for Modeling Telecommunication Networks". ACM SIGMETRICS Performance Evaluation Review, Vol 25, Issue 4, 1998, p4-11.
- [38] Perumalla K.S. and R.M. Fujimoto. "Efficient Largescale Process-oriented Parallel Simulations". Proceedings of the 30th Winter Simulation Conference, 1998, p459-466.
- [39] Wilsey P.A. and D.M. Rao. "Simulation of Ultra-Large Communication Network". Proceedings of 7th International Symposium on Modeling. Analysis and Simulation of Computer and Telecommunication Systems, MASCO'T'99, 1999, p112-119.
- [40] Wilsey P. A. and D. M. Rao. "An Ultra-Scale Simulation Framework". Journal of Parallel and Distributed Computing, January 2000, Vol 10, No 1, 2000, p18-38.