Network Topology Generation and Discovery Tools

Muhammad Azizur Rahman[1], Algirdas Pakštas[2], Frank Zhigang Wang[3]
[1,2]London Metropolitan University, 166-220 Holloway Road, London N7 8DB, England
[3]Cranfield University, Cranfield, Bedfordshire MK43 0AL, England
[1]mam406@londonmet.ac.uk, [2]a.pakstas@londonmet.ac.uk, [3]f.wang@Cranfield.ac.uk

Abstract: **Achieving a deep understanding of the Internet topology has proved very challenging since it involves solving difficult task, e.g. mapping the actual topology, characterising it and developing generation models that capture the fundamental properties. There are two categories of tools which were developed to deal with network topology, namely, topology generations tools and network/topology discovery tools. Topology generation tools are used for studying and simulating the Internet. Discovery tools are mostly useful for more practical purposes such as real data acquisition for network monitoring and management but also can be helpful if there is a need to simulate network before planning and applying certain changes. Both categories are targeted as components for the integrated Network Design and Simulation Environment (NeDaSE) developed by the authors. The purpose of this paper is to make an overview of the capabilities of some of these tools.**

I. Introduction

Computer network is a complex mix of hosts, applications, operating systems, traffic flows, communications protocols, link technologies and topologies. Network design process is a challenging task, requiring designers to balance user performance expectations with costs and capabilities [1, 2, 3].

There are various tools which may help in network design process [4] but they rarely can "talk to each other", import models created by other tools or export their own models in portable format. The integrated *Network Design and Simulation Environment NeDaSE* [5] attempts to bring various tools together [6], resolve incompatibilities between the tools [7] and achieve a task of having a common network description language [8] which is based on the special ontology [9].

Various categories of tools were identified and evaluated for possible integration in the process of the research and development of the tool NeDaSE. Some of them have only historical interest but other are still in the current use in academia for teaching or industry for practical network design. These categories are:

- Modeling and analysis tools;
- Simulation tools;
- Topology generation tools;
- Discovery tools.

The modelling/analysis and simulation tools exists for some time and they have been surveyed in the literature (e.g. [10] lists 42 various tools). However this is not the case with the network topology generators and discovery tools.

The network topology generators are often used by researchers in order to generate realistic topologies for simulation purposes and investigation of network performance characteristics.

Network discovery tools are primarily serving a task of collecting real network data (topology, names and addresses of network elements, applications, etc.) for producing network maps and other documentation as well as data for network monitoring and management tools. Some of the network management tools also have discovery function in order to avoid manual data entering to the tool. Additionally, discovery tools can be useful for producing input data for network simulation before planning and applying changes.

Current version of NeDaSE integrates 4 tools: general-purpose packet level network simulator ns-2 [11], wireless simulator Glomosim [12], wide are network modelling and analysis tool Delite [1,13] and topology generator BRITE [14, 15]. The purpose of this paper is to survey some topology generation and discovery tools what can be useful for the future extension of NeDaSE as well as may help other researchers to familiarise with features of the tools.

The rest of the paper is organized as follows. Section II defines network topology and is discussing its importance. Section III describes topology generation tools. Discovery tools are discussed in Section IV. Finally, conclusions are presented in Section V.

II. Network Topology And Its Importance

Network topology is a representation of the interconnection between directly connected peers (elements) in a network [16]. In the modern IP networks *physical* network topology represents connections between ports (peers) on devices connected by a physical transmission link. *Logical* topologies are related to the physical topology in many ways, each at different level of abstraction. For instance, distance between peers such as hosts/routers can be one IP hop but when workgroup abstraction level is considered, the peers might be connected by a logical link.

Also, an accurate knowledge about network topology is needed for the following purposes [16]:

- *Simulation* of real networks is impossible without it.
- *Network Management* is relying heavily on the actual up-to date knowledge about situation in the network.
- *Planning of Network Sites* can be done using network maps, knowledge about locations of the Internet Service Providers and possible latency related to access them as well as available link capacity.
- *Topology-aware Algorithms* may improve performance by exploiting knowledge of physical connectivity (e.g. topology-sensitive policy and QoS routing, group communication algorithms with topology-aware process group selection).

Good models of the topological structure of a network are essential for developing and analysing internetworking techniques. A topology is very important because design of

ISBN: 1-9025-6013-9

effective protocols, solving internetworking problems (routing, resource reservation, administration), create accurate model for simulation, derive estimates from topological parameters, study fault tolerance and anti-attract properties, etc.

Understanding the topology of the Internet is also very important for protocol design (e.g. build better protocols), protocol evaluation (e.g. find out how a new protocol would perform if deployed), protocol analysis (e.g. find and fix problems in existing protocols). Performance of networks critically depends on the topology (e.g., convergence of route information).

III. NETWORK TOPOLOGY GENERATION TOOLS

There are not too many works which are focused on the overview of topology generation tools and if they do, it mostly relates to the algorithms on which tools are based (see e.g. [17, 18]). It appears that there are the following historical periods in development of topology generators and their use for Internet research:

- Before 1999 when there was a strong belief that Internet is hierarchical [18] (Waxman algorithm, tools Tiers, Transit-Stub);
- 1999-2001 after it was discovered [19] that the Internet's degree distribution is a power law and most of the work was focused on producing and simulating such topologies (see e.g. [17]);
- Since 2001 [18] when attention was shifted again from local properties well represented by degree distributions towards large-scale properties which naturally are better represented by hierarchical generators.

Most of these tools found are discussed briefly in this section.

Waxman [20] is one of the first topology generators which produces random graphs based on the Erdos-Renyi random graph model, but it includes network specific characteristics such as placing the node on a *plane* and using a *probability function* to interconnect two nodes in the Waxman model that is parameterized by the distance that separates them in the plan.

Tiers [17, 21] is a multi-tier network topology generator that implements models trying to imitate the structure of the Internet. The generated model of Tiers is based on a three-level hierarchical structure aimed reproducing the differentiating between WANs, MANs and LANs comprising the Internet. To generate a random topology using Tiers, one specifies a target number of LANs and MANs. Currently Tiers cannot generate more than one WAN per random topology. For each level of hierarchy, one also specifies a fixed number of nodes per network. A *minimum spanning tree* is computed to connect all edges, then other edges are created based on user-specified average inter-level and intra-level redundancy. Edge formation favors close-by nodes, resulting in topologies with large diameters. Tiers is written in C++.

Transit-stub (TS) [22] is a package for generating and analyzing graph models of internetworks. According to the edge count, the Transit-Stub model produces the connected sub-graphs by repeatedly generating graph and checking the graph for connectivity and unconnected graph are cancelled. This method ensures that the resulting sub-graph is from all possible random graphs. Several types of information are related to nodes and edges for the augmentation of the basic topology (e.g. label (string) of node for properties of node, an identifiers of each node for indicating the stub or domain, global identifier for the belonging domain, a domain-local identifier). Each edge has a routing policy (shortest path) weight that can be used to find routes that follow the standard domain-based routing. TS model does not currently support representation of host systems. The TS generation software is written in C language.

GT-ITM [23] is a popular topology generator that produces topologies based on several different models. The GT-ITM topology generator can be used to create flat random graphs and two types of hierarchical graphs, the N-level and transit-stub. The main characteristics of GT-ITM are that it provides the Transit-Stub (TS) model that focuses on reproducing the hierarchical structure of the topology of the Internet. In the TS model, a connected random graph is first generated. Each node in that graph represents an entire *Transit domain*. Each transit domain node is expanded to form another connected random graph, representing the backbone topology of the transit domain. Next, for each node in each transit domain, a number of random graphs are generated representing stub domain that are attached to that node. Finally, some extra connectivity is added, in the form 'back-door' links between pairs of nodes, where a pair of nodes consists of a node from a transit domain and another from a stub domain or one node from each of two different stub domains. GT-ITM also includes five flavours of flat random graphs.

Inet [24] and **PLRG** [25] are two generators aimed at reproducing the connectivity properties of Internet topologies. These generators initially assign nodes degree from a power-law distribution and then proceed to interconnect them using different rules, Inet first determined whether the resulting typologies will be connected, forms a *spanning tree* using nodes if degree greater than two, attaches nodes with degree one to the spanning tree and then match the remaining unfulfilled degrees of all nodes with each other. PLRG works similar to Inet in that it takes as an argument the numbers of the nodes to be generated and exponent value of alpha. This exponent value is the parameter n power law distribution which is used to assign a prior degree to the nodes of the topology.

BRITE [14, 15] is a generator based on the AS power-laws. Furthermore, BRITE also incorporates recent findings on the origin of power-laws and observations of *skewed network* placement and *locality in network connections* on the Internet. By studying a number of existing topology generators, the authors of BRITE claim that the preferential connectivity and incremental growth are the primary reasons for power-laws on the Internet. For completeness, topologies are generated that incorporate both skewed node placement and locality in network connections as well as topologies with just incremental growth and preferential connectivity. To generate a topology on a plane, the plane is first divided into HSxHS squares, then the number of nodes in each square is assigned according to the node placement (NP) which is either a uniform random distribution or a bounded Pareto distribution. The bounded Pareto distribution gives a skewed node

Table I: Network Topology Generation Tools.

Tools	Scale of topology	OS	Generated Topology	Implemented Languages	Type of Output
Waxman	Large	Unix	Random graph	C	Text file
Tiers	Large	Unix	Three level-hierarchy model for LAN, MAN, WAN	C++	Text file
Transit-Stub	Large	Unix	Random graph	C	Text file
GT-ITM	Very Large	Unix	Transit-stub model	C	Text file
Inet	Very large	Unix	Internet, Spanning tree	C	Text file
PLRG	Large	Unix	Spanning tree	C	Text File
BRITE	Very Large	Unix	Seven types of model	C++, Java	Text fole
KOM ScenGen	Large	Windows, Unix	Network Scenario (topology and traffic)	Java	Text File

placement where a non-negligible number of squares have a large number of nodes in them. Each square is further divided into LSxLS smaller squares and the assigned nodes are then uniformly distributed among the smaller squares. A backbone node is selected from each of the top-level squares populated with nodes and a spanning tree is formed among the backbone nodes. Nodes are then connected one at a time to nodes that are already connected to the backbone. A new node can have preferential connectivity in its choice of neighboring nodes: locality-based, outdegree-based or both. The locality-based preferential connectivity uses a *Waxman probability function* to connect nodes in the topology. In outdegree-based preferential connectivity, the probability of a new node connecting to an existing node is the ratio of the existing node's outdegree over the sum of all outdegrees of nodes in the connected network. Finally, when mixing both locality based and outdegree-based preferential connectivity, the probability of connecting to an existing node under outdegree-based preferential connectivity is weighted by the Waxman probability between the new node and the existing node. Each new node introduces new links.

KOM ScenGen [26] is a topology generator that supports the manual and automatic creation of experimentation scenarios for network research from the topology creation over traffic generation to evaluation. The scenario includes all parameters needed for the simulation and experiment, e.g. topology, link and node properties, traffic mix, parameters, measurement points etc. In ScenGen, in the first step, a topology is created manually or automatically. Then the properties of the links and nodes (e.g. capacity, queuing algorithm) are set manually or automatically. Also the traffic parameters for the scenario have to be set. Next the network load which is the traffic of all nodes is created. This step can be followed by a plausibility check where several things critical for the scenario can be checked for plausibility. An example would be estimating the capacity necessary for the generated traffic and comparing it with the available capacity. If much more bandwidth is needed than offered, the operator might want to change the scenario parameters. After the plausibility check the scenario is exported to ns-2 for simulation and/or to a collection of scripts and configuration files that are used to setup the scenario in a testbed. The next step is to manually adapt the ns-2 files or the scripts and configuration files for specific needs. After that the simulation or experiment can be conducted and in the last step be evaluated. In this tool, there are several traffic models, sink models, load generators, traffic generators. There is a converter with ScenGen which can import the topologies generated by the tools Tiers, BRITE, GT-ITM, Inet, and NLANR. Currently, there are two export modules available in ScenGen: one for ns-2 and other for ScenGen's own testbed.

Another set of topologies for which special generators are not required are regular topologies such as the mesh, star, tree, ring, lattice, etc. These topologies have the advantages that they are very simple and are generally used for simplicity or to simulate specific scenarios such as LANs or other shared communication media.

Table I shows the summary of network topology generation tools.

IV. NETWORK DISCOVERY TOOLS

There are many network discovery tools available. Some of the leading discovery tools are briefly discussed in this section. Discovery tools normally use the network protocols like the *Internet Control Message Protocol (ICMP)* and the *Simple Network Management Protocol (SNMP)*.

A Educational Network Discovery Tools

Fremont [27] is a network discovery tool that uses a combination of non-SNMP protocols and techniques to discover the network: watching ARP (Address Resolution protocol) packets; sending ICMP ping and netmask requests and using traceroute; watching RIP (Routing Information Protocol) packets between routers; and reading DNS reverse-lookup information and using similar naming-conversion heuristics as Scotty to locate multi-homed machines. The use of so many techniques to discover the network has the advantage of increasing the accuracy of the discovery. On the contrary, it relies on non-standard heuristics, not to mention the amount of work required to properly implement and coordinate the multitude of the protocols.

Scotty [28] is built upon a custom Tcl-based API. The network discovery tool itself is called tkined and .tkined uses ICMP (ping, traceroute, netmask request) and DNS heuristic to discover a network. The advantage of this method is that the protocols are generally well supported. The disadvantages are that accuracy can suffer, since the heuristics are based on common practice in use, not on well-defined standards.

NetMap-there are different discovery tools with the same name "NetMap". As described in [29], NetMap is an attempt to solve the problem of mapping out the interconnections of networks and machines. For maintaining network, an up-to-date map is needed of the network that shows the topology and any hardware attached. NetMap relies on a comprehensive network model that is not limited to a specific network level. NetMap uses only Internet Control Message Protocol (ICMP) and SNMPv1 for the system information. ICMP was chosen because NetMap focuses only on IP-layer detection and the features of ICMP that are virtually universally supported. NetMap can discover any network to which it has

IP connectivity due to its non-reliance on protocols such as ARP. The output of the NetMap is in text format.

When all discovering is completed, Netmap will print out the network table it has constructed as well as some statistics, such as the number of machines found, the number with valid agents and so on. If a machine is found in the ping phase but does not support SNMP, it will not be added to the network map. As SNMP does not provide much the accuracy (about 50%), NetMap is not very much used in practice. NetMap is implemented in C++ on Solaris 2.5.1 platform. It uses SNMP API library to facilitate SNMP access.

Another tool also called NetMap is described in [30] with a comparison with other network discovery tools.

Big Brother [31] is a loosely-coupled distributed set of tools for monitoring and displaying the current status of an entire network and notifying network administrator if something should be done. Big Brother consists of local clients that test system conditions and the availability of *network services* and send these status reports to one or more display servers where these reports appear as little dots on a web page, or pager servers that notify administrators about system problems. The most important features of this tool are: simple testing of the network connectivity via ping; discovery of the availability of ftp, http, smtp, pop3, dns, telnet, imap, nntp, and ssh servers; local system clients monitor disk space, CPU usage, messages, and can check that important processes are up and running; support for multiple DISPLAY and PAGER servers for high-availability; warning and alarm levels are all easily redefinable; Web display can be easily modified; support for custom external tests; Integration with other systems like MRTG; many custom tests available to test things like Oracle databases; notify via e-mail, numeric pager, alphanumeric pager, or custom pager; notify based on machine name, test type, time of day, test result; delay notifications until a problem has existed for a predetermined amount of time; require notifications be acknowledged; disable repeated pages from the web display; automatic escalation should a problem exist for longer than a predetermined amount of time.

B Commercial Network Discovery Tools

LANsurveyor [32] makes it easy to map, manage, and report on entire network. LANsurveyor provides four essential functions in one cost-effective application: automatic network maps, asset management reports, network monitor, and remote administration and distribution of software. Once network nodes are discovered, LANsurveyor compiles the information into a cohesive, easy to view network map with lines representing network connectivity and each node represented with an icon. LANsurveyor generates a map of the entire network automatically using several different methods, including ICMP (ping), NetBIOS, and SNMP. Maps can be printed or exported for display or editing in any editor. LANsurveyor allows administrators to create reports that include more than 100 different pieces of information.

NetView [33] allows IP discovery, visualization, automatic update, event notification and so on. NetView performs TCP/IP discovery and displays network topologies for administrators. Also, the software manages events and SNMP traps and performs network monitoring by identifying network failure root causes and gathering trending and analytical data. It too seems use SNMP for gathering its system information. NetView uses a Web-based interface, so the application's data is easily accessible from any Web browser. NetView maintains a device inventory, easing network administrators' asset management tasks. NetView is available for a variety of platforms, including AIX, Linux, Solaris, and Windows NT/2000.

Nessus [34] is a completely new security auditing tool which aims to be an up-to-date and easy to use tool. It is a network scanner that can check for *vulnerabilities* by attempting to exploit them. This makes it more accurate, but also more heavy-handed, than other scanning tools that assume *well-known port numbers*. Nessus supports port scanning, and attacking, based on IP addresses or host name(s). It can also search through network DNS information and attack related hosts at the bequest. Tests are implemented as plug-ins, which are grouped into families, for example dealing with distributed denial of service tools. Individual plug-ins or families can be installed or not to give good control of what vulnerabilities are scanned for. Plug-ins are frequently updated to cover new vulnerabilities. Key features of Nessus are multihost testing, multithreading, plugin support, easy-to-write plugins, easy-to-use reporting system etc. The plug-in architecture of Nessus allows users to customize it for their systems and networks. As with any scanner, Nessus is only as good as the *signature database* it relies upon. Fortunately, Nessus is frequently updated. It features full reporting, host scanning, and real-time vulnerability searches. It has a client/server architecture, the server currently runs on Linux, FreeBSD, NetBSD and Solaris, clients are available for Linux, Windows and there is a Java client. There could be false positives and false negatives, even in a tool as powerful and as frequently updated as Nessus.

Nmap [35] is a popular much more fully-featured host scanning tool that can be used to determine the topology of a network. Nmap has been available for many years and is probably the most often used tool when gathering information. It features advanced techniques such as TCP-IP fingerprinting, a method by which the returned TCP-IP packets are examined and the host OS is deduced based on various quirks present in all TCP-IP stacks. Nmap also supports a number of scanning methods from normal TCP scans (simply trying to open a connection as normal) to stealth scanning and half-open SYN scans (great for crashing unstable TCP-IP stacks). Administrators can use Nmap on a network to find host systems and open ports on those systems. Nmap is a competent first step in vulnerability assessment. User can map out all the hosts within his network and even pass an option that will allow it to attempt to identify the operating system running on a particular host. Nmap is a good foundation for establishing a policy of using secure services and stopping unused services. Nmap can be run from a shell prompt or using a graphical frontend. A shell prompt accepts the nmap command followed by the <hostname> or <IP address> of the machine user wants to scan. Nmap tests the most common network communication ports for listening or waiting services. This knowledge can be helpful to an administrator who wants to close down unnecessary services.

Table II: Network Discovery Tools.

Tool	Usages	Description	Node Discovery	Topology Discovery	Node Management	Output
Fremont	Educational	Topology discovery tool	Yes	Yes	No	Text file
Scotty	Educational	Network management Tool	Yes	Yes	Yes	Text file
NetMap	Educational	Port scanning, network analysis tool	Yes	Yes	No	Text file
Big Brother	Educational, commercial	Network monitoring tool	No	No	Yes	Text file
LANsurveyor	Commercial	Topology discovery, Network mapping.	Yes	Yes	Yes	Text file
NetView	Commercial	Topology discovery tool	Yes	Yes	Yes	Text file
Nessus	Commercial	Vulnerability scanning tool	Yes	No	No	No text file
Nmap	Commercial	Port scanning	Yes	No	No	No text file
Open View	Commercial	Network management tool	Yes	Yes	Yes	Database
Intermapper	Commercial	Port scanning, topology discovery, network analysis tool	Yes	Yes	Yes	Text file

OpenView [36] is a Hewlet-Packard's package that covers a wide range of network and system management tasks. The tool specially covers network discovery. Based on configuration choices, it uses SNMP to gather information about network hardware, but how it determines the existence of the machines in the first place is unclear. HP OpenView is quite mature and has many partner developers. Thus, it is very comprehensive package and naturally, it is also quite expensive. In addition, due to its size, it uses a significant amount of resources and its operation has impact on the networks.

InterMapper [37] is a networking monitoring and alerting software that shows the potential network problems before end-users and customers suffer downtime or poor performance. SNMP probes discover and query elements across the distributed network - whether it spans several rooms, a building, an office park, or distributed locations. Synthetic transactions test critical applications and alert use to email, web, or directory server problems.

Table II shows the summary of the network discovery tools.

V. CONCLUSIONS

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.

The overall conclusion from this survey is that tools for very large-scale network topologies are relatively new and this area of research is rapidly evolving. At the same time this survey confirms that BRITE, which is currently integrated with NeDaSE, is a good choice due to its comprehensive functionality.

Network discovery tools are pretty diverse and can do tasks from purely topology discovery to network mapping, to port scanning and even vulnerability scanning. Discovering physical IP network connectivity is not easy task and despite the critical role of topology information in enchancing the manageability of modern IP networks, none of the network management platforms currently available on the market can offer a general-purpose tool for automatic discovery of physical IP network connectivity [38].

While OpenView or NetView are the most comprehensive tools they does not allow to get topology information easily extracted for integration to NeDaSE. Making a choice among other tools which may produce output in text format is still a subject of investigation.

REFERENCES

[1] Robert S. Cahn, *Wide Area Network Design: Concepts and Tools for Optimization,* Morgan Kaufman Publishers, San Francisco, 1998.

[2] Thomas G. Robertazzi, P*lanning Telecommunication Networks,* IEEE Press, 1999.

[3] James D. McCabe, *Network Analysis, Architecture and Design,* Morgan Kaufman Publishers. San Francisco, 2003.

[4] Arnold W. Bragg, "Which network Design Tool Is Right for You?" *IT Professional,* September/October, 2000, p23-31.

[5] Muhammad Azizur Rahman, Algirdas Pakštas, Frank Zhigang Wang, "NeDaSE: A Bridge Between Network Topology Generator, Network Design and Simulation Tools", *Proc. of the EUROCON 2005 - The International IEEE Conference on Computer as a tool,* November 21-24, 2005, Sava Center, Belgrade, Serbia & Montenegro.

[6] Muhammad Azizur Rahman, Algirdas Pakštas and Frank Zhigang Wang, "An Approach to Integration of Network Design and Simulation tools", *Proc. of the 8th International IEEE Conference on Telecommunications (ConTEL 2005),* Zagreb, Croatia, June 15-17, 2005, p173-180.

[7] Algirdas Pakštas, Muhammad Azizur Rahman, "Incompatibility of Network Design and Simulation Tools and An Approach to Integration", *Proc. of the 6th EPSRC Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (EPSRC PGNet 2005),* Liverpool John Moores University, June 27-28, 2005, Liverpool, England, p529-535.

[8] Muhammad Azizur Rahman, Algirdas Pakštas, Frank Zhigang Wang, "LNMet-X: A language for Network Metadescription Used for Integration of Tools", *Proc. of the 2005 International IEEE Conference on Software, Telecommunications and Computer Networks (SoftCOM 2005)*, September 15-17, 2005, Split, Marina Frapa, Croatia.

[9] Muhammad Azizur Rahman, Algirdas Pakštas, Frank Zhigang Wang, "Towards Communications Network Modelling Ontology for Designers and Researchers", *10th International IEEE Conference on Intelligent Engineering Systems 2006 (INES 2006)*, June 26-28, 2006, London Metropolitan University, London. (to appear)

[10] H. Akhtar, "An Overview of Some Network Modeling, Simulation & Performance Analysis Tools", *Proc. of the 2nd IEEE Symposium on Computers and Communications (ISCC'97)*, 1997, p344-348.

[11] K. Fall, K. Varadhan, *The NS Manual, The VINT Project,* 13 December 2003. (http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf)

[12] GloMoSim Manual (ver. 1.2). 07 February 2001. University of California, Los Angeles. (http://pcl.cs.ucla.edu/projects/glomosim/GloMoSimManual.html)

[13] Delite Software Manual, (http://www.cs.stevens.edu/~sghosh/courses/netdesign/DeLite/DeLite-README.html, 2006)

[14] A. Medina, A. Lakhina, I. Matta, J. Byers, "Brite: Universal topology Generator from a User's Perspective", *Technical Report, BUCS-TR-2001-003*, April 12, 2001, Boston University. (http://www.cs.bu.edu/brite/publications/usermanual.pdf)

[15] Alberto Medina, Anukool Lakhina, Ibrahim Matta, and John Byers, "BRITE: An Approach to Universal Topology Generation", *Proc. of MASCOTS 2001*, Cincinnati, OH, August 2001.

[16] R. Siamwalla, R.Sharma, and S.Keshav. (1998, July) *Discovering Internet Topology*. (http://www.cs.cornell.edu/skeshav/papers.html, 2006)

[17] Giuseppe Di Fatta, Giuseppe Lo Presti, Giuseppe Lo Re, "Computer Network Topologies: Models and Generation Tools", *Technical Report No. 5/2001*, University of Palermo, Italy, July 2001.

[18] H. Tangmunarunkit, R.Govindan, S.Jamin, S.Shenker, and W.Willinger, "Network Topology Generators: Degree-Based vs. Structural", *Proc. of the ACM SIGCOMM, 2002.*

[19] C.Faloutsos, P.Faloutsos, and M.Faloutsos, "On Power-Law Relationships of the Internet Topology", *Proc. of the ACM SIGCOMM, 1999*.

[20] B. M. Waxman, "Routing of Multipoint Connections", *IEEE Journal of Selected Areas in Communication,* Vol. 6, No. 9, December 1988, p1617–1622.

[21] M. Doar, "A Better Model for Generating Test Networks", *Proc. of IEEE Global Telecommunications Conference (GLOBECOM),* London, November 1996.

[22] K. Calvert, M. Doar, and E. Zegura, "Modelling Internet Topology", *IEEE Communications Magazine*, June 1997.

[23] K. Calvert, M. Doar, E. Zegura, "Modeling Internet Topology", *IEEE Transactions on Communications,* December 1997.

[24] Cheng Jin, Qian Chen, Sugih Jamin, "Inet: Internet Topology Generator", *Technical Report Research Report CSE-TR-433-00,* University of Michigan at Ann Arbor, 2000.

[25] William Aiello, Fan Chung, Linyuan Lu, "A Random Graph Model for Massive Graphs", *Proc. of the 32nd Annual Symposium on Theory of Computing,* 2000.

[26] Oliver Heckmann, Krishna Pandit, Jens Schmitt, Ralf Steinmetz, "KOM ScenGen The Swiss Army Knife for Simulation and Emulation Experiments", *First International Workshop on Multimedia Interactive Protocols and Systems (MIPS),* Napoli, Italy, 18-21 November 2003, p91-106.

[27] Davis C. M. Wood, Sean S. Coleman, Michael F. Schwartz, "Fremont: A System for Discovering Network Characteristics and Problems." *Proc. of the USENIX Winter Conference,* San Diego, California. 25-29 January 1993, p335-348.

[28] J. Schonwalder and H. Langendorfer. Tcl Extensions for Network Management Applications. *In Proc. 3rd Tcl/Tk Workshop*, Toronto, Canada, 6-8 July 1995, p279-288.

[29] Nelson Tang, Binary Sugla, NetMap: A Network Discovery Tool. Report for Network & Service Management Research Lab, *Bell Laboratories, Lucent Technologies,* 21 September 1998. (http://www.cs.ucla.edu/~tang/papers/lucent_discovery.pdf)

[30] Giovanni Vigna, Fredrik Valeur, Jingyu. Zhou, Richard A. Kemmerer, "Composable Tool For Network Discovery and Security Analysis". *18th Annual Computer Security Application Conference*, San Diego California, 09–13 December, 2002, p14-24. (http://www.acsac.org/2002/papers/108.pdf).

[31] Big Brother Professional Edition. Quest Software Inc., 8001 Irvine Center Drive, Irvine, CA 92618. Big Brother Homepage. (http://www.bb4.org/, 2006)

[32] LANSurveyor, Neon Software, Inc., 244 Lafayette Circle, Lafayette, CA 94549. (http://www.neon.com/, 2006)

[33] IBM Tivoli NetView discovery tool. IBM Tivoli Netview Software. Tivoli Systems Inc. (http://www-306.ibm.com/software/tivoli/ products/ netview/, 2006)

[34] Renaud Deraison. Nessus Security Scanner. The "Nessus" Project. Nessus Org. Nessus Homepage. (http://nessus.org/, 2006).

[35] Fyodor. Nmap– Security Scanner, INSECURE.ORG. (http://www.insecure.org/nmap/index.html, 2006).

[36] Hewlett-Packard Company. HP OpenView Software. (http://www.openview.hp.com, 2006)

[37] Intermapper. Dartware, LLC, 10 Buck Road, PO Box 130, Hanover, NH 03755-0130 USA. (http://www.intermapper.com/, 2006)

[38] Y. Breitbart, M.Garofalakis, B.Jai, C.Martin, R.Rastogi, A.Silbershatz, "Topology Discovery in Heterogenous IP Networks: The NetInventory System", *IEEE/ACM Transactions on Networking,* 2004, Vol.12, No 3, p401-414.