

WinNet - a network tool

Sanda Dragos Radu Dragos

Abstract

Building Internet-like network topologies and performing simulations and/or emulations on them is a task that all network-related topic researchers came across during their study. We propose here a new tool which incorporates network topology generation, visualization, simulation and an analyzing mobile agent based tool. This application can be used for research purposes as well as an educational instrument. Even if it is mainly aimed for computer networks, this application can be also used to solve non-computer network related problems such as logical tasks or database searches.

Keywords: Network Simulator, topology generator, mobile agent, Wave

1 Introduction

The explosive growth of the Internet has been accompanied by a wide range of problems ranging from routing to resource reservation or administration. New algorithms and policies that try to solve such problems need to be tested on an abstraction network or on an actual network. Researchers rarely use real networks as their test-beds because networks large enough to be representative are also very expensive and difficult to control. A more efficient way to perform the tests is on emulated networks.

At first, researchers used very simple topologies (e.g. mesh, star, tree, ring, lattice, grid, cube) for their simulations. However, they do not reflect any real network, and are generally used now only to simulate specific scenarios such as LANs or other shared communication media. Simulating Internet-like network topologies is a very difficult task as the Internet does not fit into some specific formats, and it is constantly changing. Thus, topology generators were developed. There are a wide variety of network topology generators available. Waxman [1] developed one of the first topology generators, which is concerned with general random networks. One of the most popular topology generators available is GT-ITM [2]. It focuses on reproducing the hierarchical structure of the topology of the Internet, but also includes about five types of flat random graphs. Tiers [3] is based on a three-level hierarchy aimed at reproducing the differentiation between Wide-Area, Metropolitan-Area and Local-Area networks comprising the Internet. Inet [4] and PLRG [5] are two topology generators concerned with emulating the connectivity properties of Internet topologies as reported in [6].

Network simulators use topology generators like those mentioned above. For instance, the Network Simulator (NS) [7] uses Inet, GT-ITM and Tiers topology generators, along with topologies specified by hand, while OMNeT ++ [8] uses Inet. Most network simulators have already implemented existing data-link, network and transport protocols (e.g. IP, TCP, UDP, PPP, Ethernet, MPLS with LDP and RSVP-TE).

In this context we propose a new simulation tool, called WinNet, that can be also used as a planning instrument and topology analyzer. It uses GT-ITM as a network topology generator and Network Simulator (NS) as a simulating instrument. It also creates, by using a mobile agent system called WAVE, “virtual” networks on which to perform numerous analyzing tasks.

The Wave technology [9, 10] is based on parallel spreading of recursive program code (or *waves*) in *open systems*¹, accompanied by dynamic creation of virtual Knowledge Networks (KNs). Such networks can persist and reflect any declarative or procedural information. Moreover, they may become active and capable of self-evolution, self-organization and self-recovery. Other *waves* can navigate, control and modify KNs. All these actions are performed without a central memory or a centralized control. Another important Wave feature is that routines such as synchronization, message passing and garbage

¹Constantly evolving and changing in time and space, while intensively exchanging information with other systems and with the environment [9]. Examples of open system are computer networks.

collection are implemented within the Wave Interpreter which resides on physical nodes rather than being implemented within mobile agents as with other mobile agent technologies. This and its syntax make Wave code very compact, perhaps 20 to 50 times shorter than equivalent programs written in C++ or Java [11].

2 The WinNet tool

WinNet is a Linux-based application that creates random network topologies by using GT-ITM on which it may perform network simulations using NS by automatically building a network simulating application written in Tk/Tcl. It extends GT-ITM by associating different (single or multiple) costs to each link. Such costs are randomly chosen from given intervals. The most important WinNet feature, however, is that, by using a mobile agent system called WAVE, it is able to create and modify virtual networks having specified topologies and to perform analyzing tasks (such as finding all articulation points, determining the diameter, finding all and the maximum cliques) on these topologies.

The WinNet name is formed of two particles starting with capitals: **Win** and **Net**. The **Win** particle contains the initials of the three instruments used: **WAVE**, **GT-ITM** and **NS**, while **Net** indicates that it is an networking tool.

The application starts with a selection window depicted in Fig. 1 which allows the selection of any starting stage. That is because files created with this application during any previous utilization can be also used.

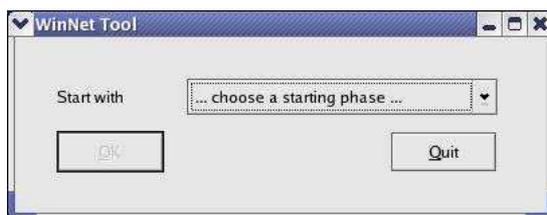
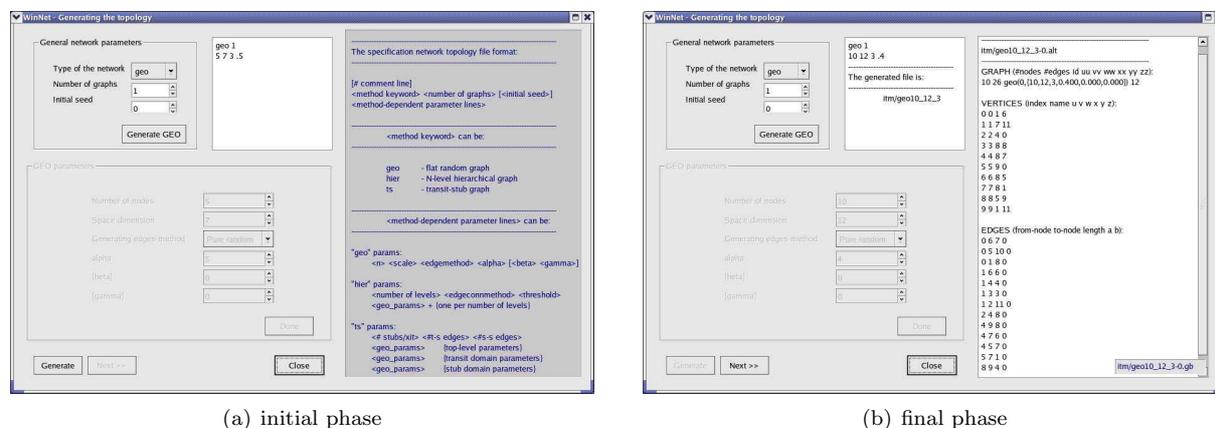


Figure 1: The main application window

The first application stage consists in building the random topology using GT-ITM. A very short help page is also provided on the right hand side of the window (see Fig. 2(a)). Selecting values for variables that may exist only within specific intervals are implemented by using combo-boxes. They also facilitate the creation of the topology description file, which can be seen and modified if needed in a textbox environment.



(a) initial phase

(b) final phase

Figure 2: Generating a random topology

After specifying the topology description file, the topology can be created by pressing the “Generate” button. A configuration file specifies the number (one or more), the type (additive, multiplicative,

concave) and the order (increasing, decreasing) of metrics to be associate with each link. Their accepted value intervals can also be specified in the configuration file. The newly created file specifying the generated topology can be viewed on the right hand side of the same window (see Fig. 2(b)) in the exact place previously occupied by the help page. The name of the topology file is generated using information from the topology specification file (i.e. *<type of the network><number of nodes>.<space dimension>.<method used for generating edges>*) and it is listed on the right-down corner of the textbox containing the content of the same file. If more files are generated, they all will be listed there.

The next application stage is converting the random topologies created in the previous stage into WAVE and NS programs as depicted in Fig. 3. Random topology files generated using any previous utilization of the WinNet application can also be used and are accessible for selection. The application allows the user to select either the Wave or the NS conversion or both of them. Such conversion will apply to all random topology files existent in the *selected files* environment. The code generated for WAVE and/or NS programs is listed in the right hand side of this interface window.

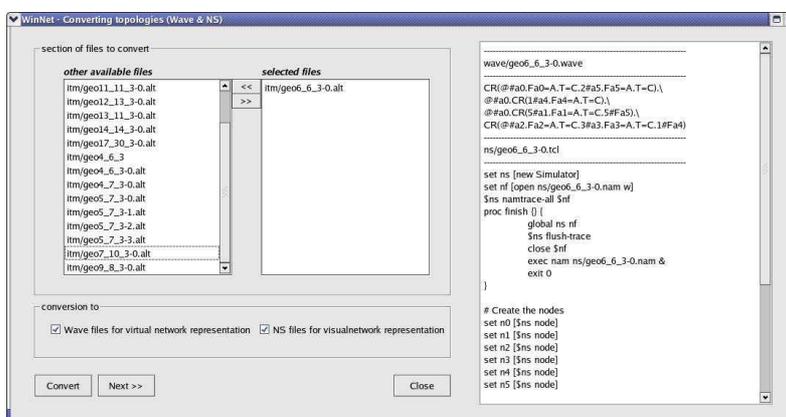
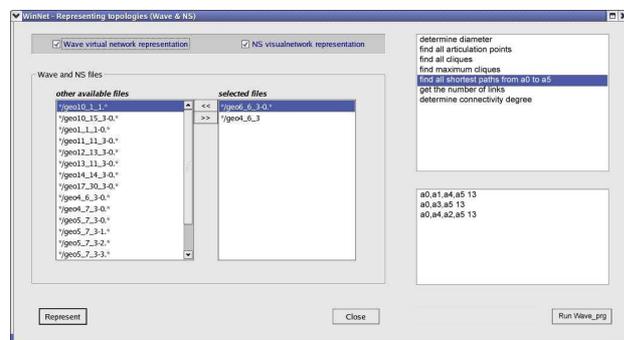
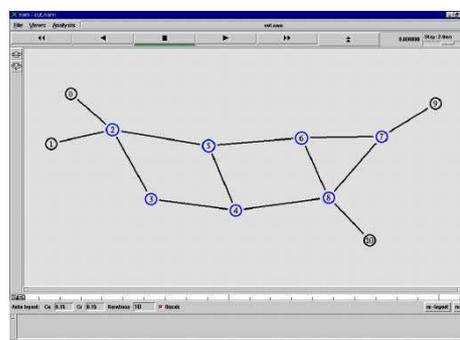


Figure 3: Converting random topologies into Wave and/or NS programs

The third application stage is the Wave and/or NS representations. An interface window (see Fig. 4(a)) allows choosing a specific topology to be represented either visually by NS, or virtually by WAVE. By using NS the user would be able to see the network topology as presented in Fig. 4(b) and to watch the packet flows over different scenarios that may be specified in the configuration file. Such scenarios may include setting a TCP or UDP connection, setting LSP's² over different paths, watch the packet queues and observe if packets are dropped or not, and so on. This scenarios can also be correlated with the findings offered by the Wave programs. One such example is to set up LSP's over shortest paths.



(a) Representation interface and Wave programs selection



(b) Network visualization using NS

Figure 4: Representation of random topologies

Selecting the WAVE representation results in creating a virtual network on one or more physical

²Label Switched Path used by the Multi-Protocol Label Switching (MPLS).

network nodes. Such network contains specified information associated with each link (as presented previously on the first application stage). The virtual network will reside on the nodes where it was created and it can be navigated by small mobile agents, called *waves*, programmed to perform a very large range of operations. Examples of such operations that can be performed by *waves* implemented by WinNet are described below and can be viewed in Fig. 4(a).

determining the diameter - The diameter is the maximum distance³ between any two network nodes. Such metric might be used in estimating Time-to-Live (TTL)-like variables used by different protocols.

finding all articulation points - An articulation point is a node which if removed divides the initial graph. The number and the exact location of articulation points can help in avoiding congestion points. Moreover, if more adjacent nodes are articulation points, their common link may become over-utilized.

finding all cliques - A clique is any complete (full-mesh) subgraph/topology within the initial topology. Knowing this information is useful for determining and avoiding possible cycles.

all shortest paths between two nodes - *Shortest*⁴ paths can be determined based on the metrics associated with links. The path computation is performed in a parallel and distributed manner using *waves*. This allows determining “shortest” paths based on multiple metrics. Shortest path trees starting from one node can also be easily determined. They are mainly used in multicast routing.

get the number of the links - This feature can be used as feedback in order to determine how different GT-ITM configuration files can affect the generated network topology in terms of number of links.

determine connectivity degree - The connectivity degree is twice the number of links over the number of nodes and represents an average of the node degree, meaning the average number of incident links to a node. This feature is important, for instance, in mobile agent programming because every mobile agent may multiply within every node with the number of incident links, and thus in highly connected networks they might generate too much traffic.

Other non-computer networks related problems, but which can be represented in a network-like model, can also be solved using this application. The configuration file can specify the assignment of non-numerical values to each link and/or node. Example of such applications are logic tasks and parallel database searches.

The WinNet flowchart is depicted in Fig. 5. It presents the main stages of our application, the files used, the places where the configuration file is consulted and the dual finality of this application.

We used this tool to perform tests on our hierarchical routing protocol called Macro-routing [12] and also to evaluate the performance of the Extended Full-Mesh aggregation [13] that we proposed for finding multiple-constraint hierarchical paths.

3 Summary and Conclusions

We presented in this paper an application called WinNet which uses GT-ITM to generate random network topologies based on which there can be created simulations by using NS and/or virtual network representation, using WAVE, used to analyze the network topology along with the random single or multiple metrics associated with network links. Thus, WinNet can be used:

- as a planning tool by simulating various network configurations and traffic loads.
- as a network analysis tool, to answer questions such as: where will the bottlenecks be in the circuits?; where are the shortest paths?; or what is the diameter and the connectivity on the network?
- as a networking educational tool.

³The *distance* between two nodes is the number of nodes to be traversed in order to reach from one node to the other.

⁴Shortest is equivalent, in this context, with BEST in relation with the metric used (i.e. by considering the order and the type of metrics).

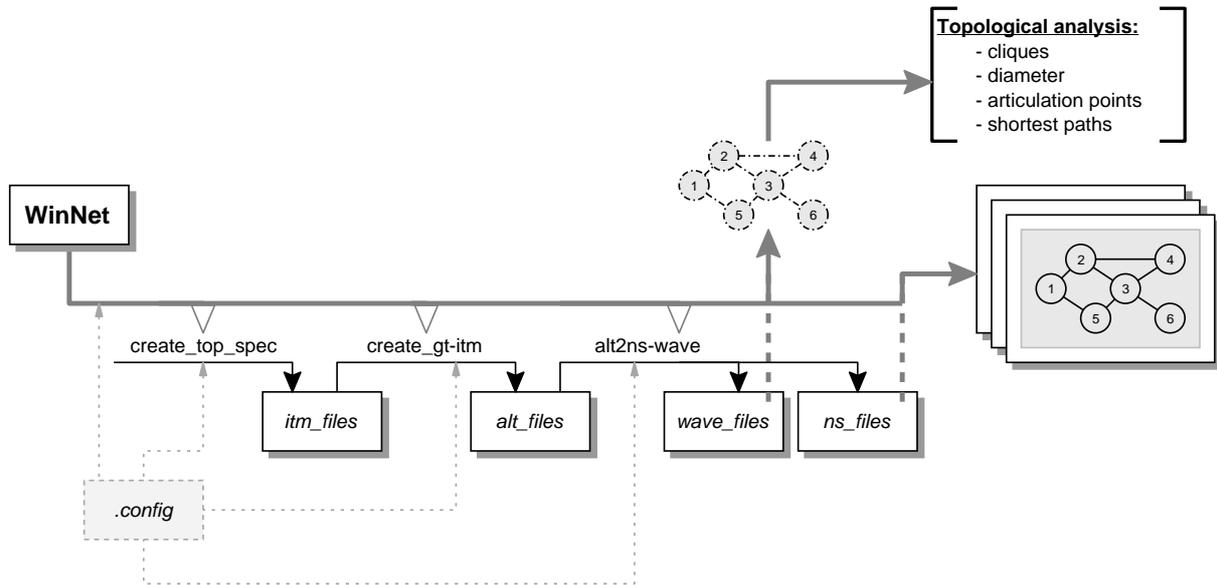


Figure 5: The WinNet flowchart diagram

- for solving other problems than the computer network ones (e.g. logic tasks, database searches).

The WinNet application has three main stages which can be accessed separately because they can use files created previously by the application. Moreover, the system can be adjusted to specific needs using a configuration file. Thus, the application can be extended to solve also non-computer network related problems. Moreover, the WAVE representation allows assigning numerical or non-numerical values to nodes as well as to links. This is an important feature for instance, in hierarchical routing, where traversing an aggregated node has an associated cost as well as traversing a link. WinNet's modularity facilitates future changes and extensions for this application.

References

- [1] B. Waxman, "Routing of Multipoint Connections," *IEEE Journals on Selected Areas in Communications (JSAC)*, 1988.
- [2] K. Calvert and M. Doar and E. Zegura, "Modeling Internet Topology," *IEEE Transactions on Communications*, pp. 160-163, 1997.
- [3] M. Doar, "A Better Model for Generating Test Networks," *Proceedings of IEEE GLOBECOM*, November, 1996.
- [4] C. Jin and Q. Chen and S. Jamin, "Inet: Internet Topology Generator," *University of Michigan*, Technical Research Report, No. CSE-TR-433-00, 2000.
- [5] W. Aiello and F. Chung and L. Lu, "A Random Graph Model for Massive Graphs," *Proceedings of the 32nd Annual Symposium in Theory of Computing*, 2000.
- [6] M. Faloutsos and P. Faloutsos and C. Faloutsos, "On Power-Law Relationships of the Internet Topology," *ACM Computer Communication Review*, September, 1999.
- [7] "Network Simulator - ns - 2," <http://www.isi.edu/nsnam/ns/>, 2002.
- [8] Andrs Varga, "OMNeT ++," *IEEE Network Interactive*, Software Tools for Networking, VOL. 16, No. 4, 2002.
- [9] Peter S. Sapaty, *Mobile Processing in distributed and Open Environments*, Wiley, ISBN: 0-471-19572-3, 2000.

- [10] Peter S. Sapaty, "The WAVE paradigm," Dept. of Informatics, Univ. of Karlsruhe, Germany, No. 17/92, July, 1992, Also published in Proc. Post-Conference Joint Workshop on Distributed and Parallel Implementations of Logic Programming Systems, JICSLP'92, pp. 106-148, Washington, D. C., Nov. 13-14, 1992.
- [11] Son T. Vuong and Ivailo Ivanov, "Mobile Intelligent Agent Systems: WAVE vs. JAVA," *Proceedings of the 1st Annual Conference of Emerging Technologies and Applications in Communications (etaCOM'96)*, Portland, Oregon, May, 1996.
- [12] Sanda Dragos and Martin Collier, "Macro-routing: a new hierarchical routing protocol," *Proceedings of the IEEE Global Telecommunications Conference (Globecom)*, November/December, 2004.
- [13] Sanda Dragos and Martin Collier, "The Extended Full-Mesh aggregation technique," *submitted to ACM Sigcomm 2006*, September, 2006.

Sanda Dragos and Radu Dragos
Babes-Bolyai University
Department of Computer Science
No. 1, Mihail Kogalniceanu street, Cluj-Napoca, 400084, Romania
E-mail: sanda@cs.ubbcluj.ro, bradu@ubbcluj.ro
Received: March 28, 2006