EXTENDED NAM: AN NS2-COMPATIBLE NETWORK TOPOLOGY EDITOR FOR SIMULATION OF WEB CACHING SYSTEMS ON LARGE NETWORK TOPOLOGIES

Roberto Canonico
Donato Emma
Giorgio Ventre
Università di Napoli Federico II
Dipartimento di Informatica e Sistemistica
Via Claudio, 21
I-80125 Napoli, ITALY
E-mail: roberto.canonico@unina.it

KEYWORDS
Network simulation, Topology generators, Web caching

ABSTRACT

Designing new network protocols for the Internet relies on the use of network simulation tools for validation. When simulation is aimed at testing the scalability of new protocols, working with large-scale realistic topologies is of paramount importance. In this paper we present an extension of the NAM Editor, a graphical editor to be used in conjunction with the well-known ns2 network simulator. Our extended NAM Editor allows the creation of synthetic but realistic network simulation scenarios and provides support for the simulation of web caching systems.

INTRODUCTION

Due to the high complexity of the modern Internet, the recourse to simulation tools for the evaluation of the effectiveness of new protocols and the feasibility of new applications, is of uttermost importance (Breslau et al. 2000). In particular, when simulation is aimed at testing the scalability of new protocols and applications, working with large-scale realistic network topologies becomes an indispensable requirement (Calvert et al. 1997; Cowie et al. 1999; Floyd and Paxson 2001).
In the last few years different network simulation tools have been developed. One of the most popular is the ns2 Network Simulator (Breslau et al. 2000). Ns2 is an open-source event driven network simulator. It provides support for simulation of IP-based network. In particular, ns2 provides researchers with:

- multicast and unicast routing protocols;
- different transport protocols (TCP, UDP, RTP, etc..);
- most common applications (FTP, Telnet, HTTP).

The ns2 official distribution is complemented by NAM, a graphical tool for visualization of simulation results (Estrin et al. 2000). More recently, NAM has been enriched with minimal editing functionality. This new tool, called NAM Editor, allows the interactive creation of simulation scenarios. The usefulness of NAM Editor, however, is quite limited, since its design capabilities are suitable for network of limited size only.
In order to simulate networks with realistic topologies, it is a common practice to use ad-hoc topology generators like *gt-itm* (Calvert et al. 1997; GT-ITM 2003) and *INET* (Jin et al. 2000). The output of these topology generators is usually not compatible with the ns2 syntax. Hence, several tools have been developed to translate topology descriptions generated by topology generators in ns2 scripts that can be used in the definition of a simulation scenario. Unfortunately, scripts produced in this way are not compatible with the NAM Editor, hence networks created by common topology generators cannot be modified interactively. Such a limitation is sometimes annoying, in particular when the automatically generated topology needs to be further adapted, e.g. by instantiating agents on particular network nodes.
Once the network topology has been generated, the simulation scenario must be completed with traffic data. This requires the definition of proper traffic patterns (according to the characteristics of the application that is being simulated), and the localization of traffic end-points (sources and sinks). In a complex simulation scenario (e.g. composed by hundreds or thousands of nodes) localizing traffic end-points is not a trivial operation, which can be usefully supported by a graphical interactive tool like the NAM Editor.
To overcome the above limitations we have implemented an Extended version of the NAM Editor with the following features:

1. integration of existing topology generators;
2. localization and visualization of sets of nodes on large network topologies according to different selection criteria;
3. instantiation of *agents* (i.e. the elements that implement transport-level or application-level functionality in ns2 terminology) of any type on all the nodes of a given node set;
4. customization of node parameters;
5. definition of new node types (e.g. web caches);
6. support for simulation of web cache systems.

The main features of our tool will be described in the following sections.

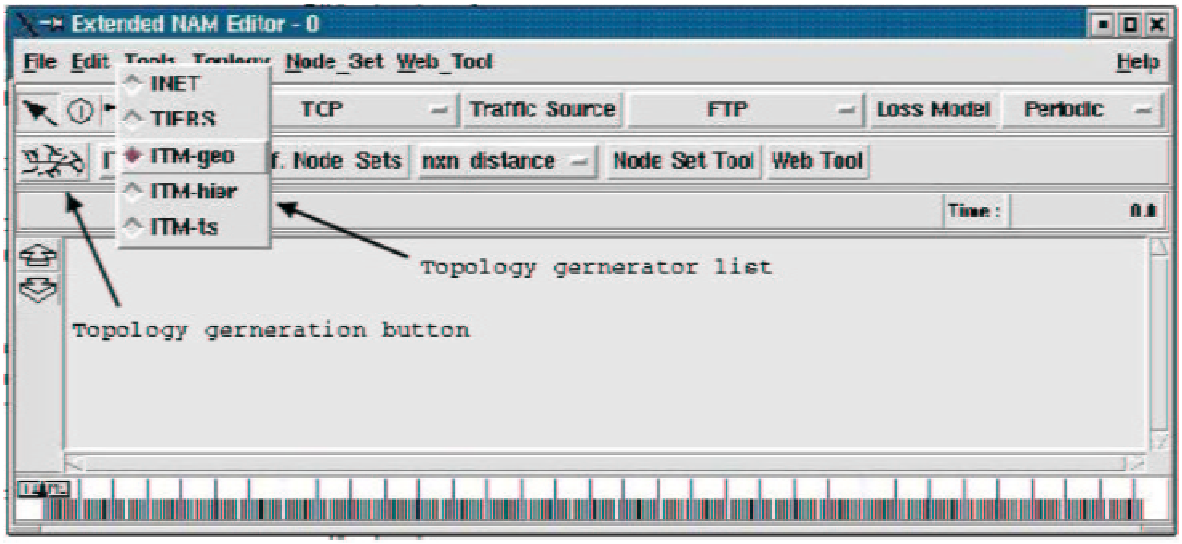

Figure 1: Topology Generation within Extended NAM Editor

INTEGRATION OF TOPOLOGY GENERATORS

Our Extended NAM Editor makes it possible to create, visualize and modify topologies created according to the following topology models:

- flat-itm;
- hierarchical-itm;
- transit-stub-itm;
- INET model.

Our tool not only provides the topology generators with the required input parameters, but it also translates the generated topology description in the internal representation of a network used by the NAM Editor.

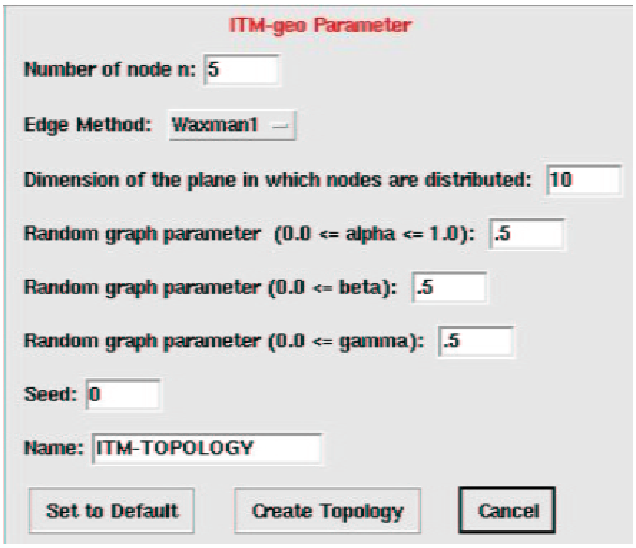

Figure 2: Input Parameters for the ITM-geo Topology

Figure 1 shows how this feature has been integrated in the NAM Editor GUI. Figure 2 shows the input parameters the user has to provide in order to generate a network topology with the gt-itm generator.

LOCALIZATION OF NODE SETS

Localization of a network element (*agent* in ns2 terminology) in a simulation scenario is a two steps process, which requires:

- the localization of the node in which the agent must be instantiated
- the instantiation of the agent.

To support agent instantiation in large topologies we have included in the extended NAM Editor the concept of *Node Set*. A Node Set is a set of nodes selected according to one of the following criteria:

- leaf node;
- mutual distance;
- randomly.

When a network topology is created with one of the supported topology generators, some Node Sets are automatically created, reflecting the topology model. For instance, in the case of a transit-stub topology, a Node Set is associated to each transit domain and a different one to each stub domain.

To make this tool even more flexible, we made the GUI of our Extended Nam Editor customizable by the end user. Thanks to this feature, it is possible to:

- add an user-defined agent to the agent list;
- customize the parameters of a user-defined agent;
- define new node type and its parameters

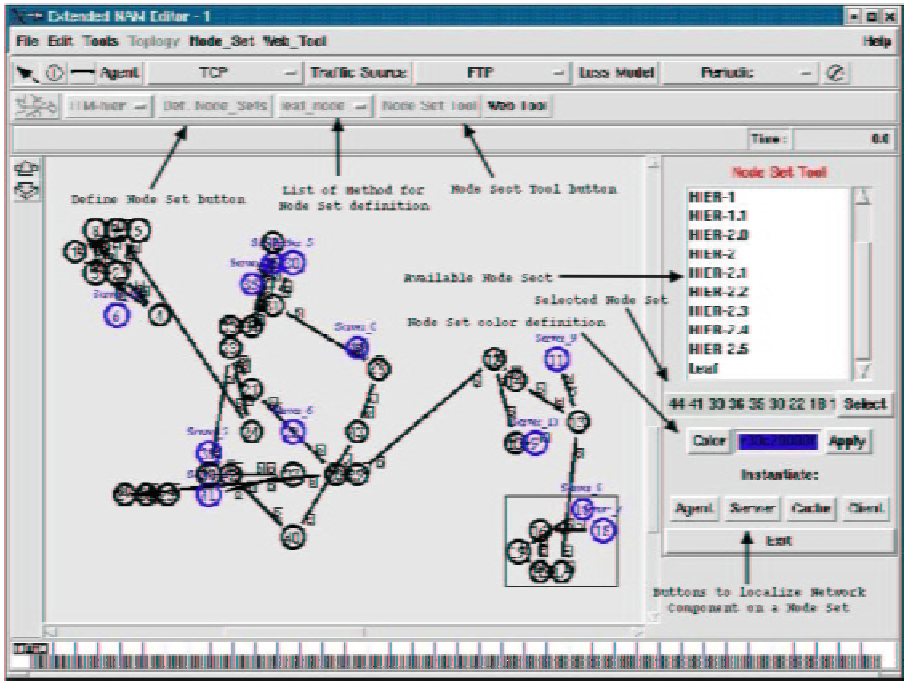

Figure 3: The Node Set Tool in the Extended NAM Editor

Features described in this section are implemented by the Node Set Tool (Figure 3) which allows to:
1.visualize all the nodes included in a given set;
2.associate a color to all the nodes included in a given set;
3.instantiate an agent on all the nodes of a given set;
4.instantiate a Web Server on all the nodes of a given set;
5.instantiate a Web Client on all the node of a given set;
6.instantiate a Web Cache on all the node of a given set.
The latter three functionalities, in particular, are meant to support the simulation of web cache systems on large network topologies.

SIMULATION OF WEB CACHE SYSTEMS

A special support as been provided in our Extended NAM Editor for simulating web cache systems. A proper tool can be used to create web-page pools and to deploy system of caches, servers and clients onto a previously generated topology. The configuration of any web element is made through a configuration window.
The combined use of this functionality with the possibility of generating realistic large scale topologies makes this tool extremely useful for the simulation of Content Delivery Networks or other systems in which management of replicated content is done through distributed caches.

CONCLUSIONS

In this paper we have presented an interactive graphical editor that can be used as a front-end for the well-known ns2 network simulator. Our tool integrates most popular network topology generators and also provides users with several features that are extremely useful for simulating large-scale systems of web caches and Content Delivery Networks.

ACKNOWLEDGEMENTS

This work has been partially carried out under the financial support of the Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) in the framework of the FIRB project "Middleware for advanced services over large-scale, wired-wireless distributed systems" (WEB-MINDS), and of the project "Scalability and Quality of Service in Web Systems".

REFERENCES

Breslau L.; D. Estrin; K. Fall; S. Floyd; J. Heidemann; A. Helmy; P. Huang; S. McCanne; K. Varadhan; Y. Xu; and H. Yu. 2000. "Advances in Network Simulation". *IEEE Computer*, Vol. 33 No. 5 (May), 59-67.

Calvert, K.; M. Doar; and E. W.Zegura. 1997. "Modeling Internet Topology". *IEEE Communications Magazine*, Vol. 35 No. 6 (June), 160-163.

Cowie J. H.; D.M. Nicol; and A.T. Ogielski. 1999. "Modeling the Global Internet". *Computing in Science & Engineering*, Vol. 1 No. 1 (Jan), 42-50.

D. Estrin; M. Handley; J. Heidemann; S. McCanne; Y. Xu; and H. Yu. 2000. "Network Visualization with Nam, the VINT Network Animator". *IEEE Computer*, Vol. 33 No.11 (Nov), 63-68.

Floyd, S. and V. Paxson. 2001. "Difficulties in Simulating the Internet". *IEEE/ACM Transactions on Networking*, Vol. 9, No. 4, (Aug), 392-403.

GT-ITM. 2003. Modeling Topology of Large Internetworks. University of Georgia Tech.
Available at: http://www.cc.gatech.edu/projects/gtitm/

Jin, C.; Q. Chen; and S. Jamin. "Inet Topology Generator". Technical Report CSE-TR-433-00, EECS Department, University of Michigan, 2000.