Some features of computer network modelling in the OMNeT++ environment

D Y Polukarov¹, A P Bogdan¹

¹Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086

e-mail: PLKW@mail.ru

Abstract. There are many currently available tools for computer network simulation. However, all of these are intended for use by network engineers and are not designed for research tasks. Such simulation models have insufficient flexibility and degree of parameter variability to be used for research purposes. Here, it is proposed to create a simulation model for modern computer networks which allows for the research experimentation. It is proposed to use one of the open packages available which facilitate computer network simulation modelling (for example, OMNeT++) as a base platform. Open-architecture packages allow the researcher to create specialized simulation models. With the help of models created specifically for the needs of research, it will be possible to conduct properly rigorous experiments. This paper utilizes one of the existent implementations of the Barabasi-Albert model for generating computer network topologies. This simulation model can be run and tested within the OMNeT++ discrete event simulator.

1. Introduction
Many computer network simulators do not allow the simulation of routers using arbitrary routing protocols. However, the OMNeT++ modeling software [1] is a modified eclipse programming environment [2]. This environment provides flexibility to the computer modeling process.

Research in the field of computer networks is developing across three areas:
- field tests (too expensive and cumbersome);
- analytical studies (but it is not always possible to build an accurate and adequate mathematical model);
- simulation modeling (software-based experimentation).

Of the above areas, the most rational to pursue is the last one, and such computational experiments are used in various fields of research, for example [3], [4]. But in order to carry out simulations, it is necessary to have good computer models of modern networks.

In order to model novel components (for example, routing protocols), you need to create a computer network model that meets the following requirements. On the one hand, such a model should correspond to real networks, and on the other hand, it should meet clear formal criteria. If the network to be studied is small, then it is possible to create such a model manually. If the simulated network is large enough, then the process of its creation must be automated.
The following study demonstrates just how large a network can be. In 2003, Barrett Lyon launched a project whose goal was to show how complex and voluminous the structure of the World Wide Web was [5]. Figure 1 shows one of the images created for this project. Each point represents an IP address, the distance between these points represents the delays and the color coding represents geographical location.

![Figure 1. World Wide Web visualization example [6].](image)

At first glance, such networks seem random, but in 1999, Albert-László Barabási and Réka Albert found that this was not the case. In their study [7], they analyzed various large networks: the World Wide Web, the Internet, air traffic networks, citation networks, etc. And they found that all these networks were quite similar and share the following property: most of the links are concentrated among a small number of nodes. Such networks were called scale-free.

The simulation of computer networks is usually achieved using simulation modeling, and to be more precise, discrete-event modeling. Discrete event modeling is based on the processing of a chronological sequence of events, created by the system, which causes the system to change its state. In computer networks, such events are packets that cause device activity (packet processing). Currently, there are quite a large number of different computer network simulators, but these all assume that the network topology is known in advance. In the case of large networks, such as the Internet, the researcher must either abandon this concept entirely or manually create such a network – a very time-consuming task. In order to solve this problem, it was decided to create a system that automatically creates a large network.

2. Related works
A number of papers are devoted to a comparative analysis, and the optimal choice, of computer network modeling systems. In [8], the author is at pains to make an informed analysis and selection of telecommunications systems modeling environment. As a result, the author makes a choice in favor of OMNeT++.

Another paper [9] describes the advantages of the OMNeT++ environment for modeling arbitrary network structures.

Also, a number of papers are devoted to the creation of new elements for the OMNeT++ modeling environment.

In [10], [11], the authors present an overview of the theory behind the EIGRP routing protocol. The main contribution made is the new OMNeT++ modeling module, which simulates the implementation
of the Cisco EIGRP protocol, based on the available specification and the results of reverse engineering. The paper presents a simulation and related results in order to demonstrate consistency with the Cisco IOS reference implementation.

The article [12] describes the Babel dynamic routing protocol and its implementation as a simulation model within OMNeT++. A new simulation model is developed and implemented that simulates the behavior of a complete Babel implementation for OMNeT++. In addition, the accuracy of the simulation results are verified, taking into account the real basic indicators of the wired network. The Babel simulation model developed is also a tool which can be used in future work. It is planned to include this model in the official release of INET.

The paper [13] presents the structure which has been implemented within the OMNeT++ simulator to facilitate the implementation of routing protocols in a flexible way. The framework divides opportunistic routing into four main stages and abstracts these as virtual functions. Some other functions which are used by most flexible routing protocols are also included in the framework. Using this framework, two flexible routing protocols are implemented: ExOR and MORE. The performance of ExOR and MORE is analyzed and simulated in detail. Situations in which opportunistic routing may be advantageous compared to traditional MANET routing mechanisms are demonstrated. A detailed comparison is made between some of the most representative opportunistic (ExOR and MORE) and proactive (OLSR) routing protocols. This comparison is based on two common implementations of the protocols with both conditional and geographical routing.

3. Models of scale-free networks
There are several models used for scale-free networks.

The Barabashi-Albert [7] model is based on two ideas: the growth of the network and the principle of preferred connection. The growth of the network implies the constant appearance of new nodes which are connected to the already existing ones. The principle of preferred joining implies that the probability with which a new node joins another node depends on the degree of the latter node.

The fitness model [14] is based on the idea of a fitness indicator for each node. In the case of computer networks, various parameters, such as geographic location and accessibility, can serve as an indicator of suitability.

The Bianconi – Barabashi [15] model is based on both the fitness model and the Barabashi – Albert model. When connecting new nodes, the degree of the node and the fitness indicator are taken into account. As a result, the Barabashi-Albert model was chosen for its ease of implementation and synergy with real networks.

Below is a comparative table of the three models.

| Criteria             | Model Barabashi-Albert | Fitness model | Bianconi – Barabashi model |
|----------------------|------------------------|---------------|---------------------------|
| Power distribution   | \( P(k_i) = \frac{k_i}{\sum_j k_j} \) | \( P(\eta_i, \eta_j) = \frac{\delta \eta_i \eta_j}{1 + \eta_i \eta_j} \) | \( P(k_i) = \frac{\eta_i k_i}{\sum_j k_j} \) |
| The main criterion   | Node rate              | Fitness score | Node rate, fitness score  |
| for joining          |                        |               |                           |

Summing up, we can say that at the moment there is no model that is similar in all respects to real computer networks. For all the models listed there are drawbacks, but the most widely suitable is the Barabashi-Albert model. Compared to the others, it is relatively simple (this is not the case with models using the fitness coefficient) and it has sufficient similarity to real networks.

4. Design and Implementation
For creating networks using this (the Barabashi-Alber) model, the following primitives will be used:
- Target device (Host) - a node with one connection;
- Router - a node with several connections; and
- Link - to connect nodes.

It is worth noting again that one of the ideas of the Barabashi-Albert model is the constant growth of the network. However, it is not possible to provide a network with this property when describing it using the built-in OMNeT++ language NED (NEtwork Description) - which does not allow for the creation of modules during simulation. Therefore, it was decided to allow a simplification. The network will grow until the number of devices in it becomes equal to a certain number predetermined by the user. Also, the initial number of nodes in a network will always be two.

According to the official documentation [16], the creation of your own module involves the following actions:
- the creation of a class describing the module; the class must be inherited from omnet::cModule;
- overriding the virtual void initialize() method, in which you need to implement the required functionality;
- registering your module in OMNeT++ using the DefineModule(<name>) macro.

In the course of development, it was found that the INET Framework initializes components in several stages. Since the manual creation of modules violates the internal initialization order of OMNeT++, it is necessary to override the virtual bool initialize(int stage) method, thereby adding initialization support to the module in several steps. Then, it is necessary to skip the initialization of all network modules, create a network according to the selected model, add modules and channels to the network to communicate with each other, and finally initialize them.

Creating a network which accords to the Barabashi–Albert model involves a number of steps:
- the creation of an adjacency matrix in which all links between the modules will be stored;
- the establishment of communication between the initial nodes;
- the adding of a new node that joins the already existing ones according to the principle of preferred connection;
- repeating step 3 until the number of nodes in the network becomes equal to the specified maximum number; and
- the analysis of the resulting adjacency matrix, the adding of a module to the network (if the node has only one connection - it is considered the final device, otherwise, it is a router) and the connection between them (by the adjacency matrix).

The realization of some of these steps is given below.

```cpp
for (int i = 0; i < countOfModules; i++) {
    std::string name;
    int tmp = 0;
    for (int j = 0; j < countOfModules; j++)
        if (connections[i][j])
            tmp++;
    if (tmp > 1) {
        name = "router";
        name.append(std::to_string(i));
        createModule(ROUTER, name.c_str());
    } else {
        name = "host";
        name.append(std::to_string(i));
        createModule(MOST, name.c_str());
    }
}
```

Figure 2. Creating modules.

Below is the pseudocode and flowchart of the algorithm for filling the adjacency matrix.
5. Testing
Further testing was performed on the software which was created. This testing was divided into two stages.
1) Generating the network and checking the correctness of its topology.
2) Checking the operability of the resulting network.

5.1 Network generation
As a test case, a network of 400 devices was created. This is presented in Figure 5.
As an example, below are several characteristics of Router298 in Figure 6.
The Neighbor Table illustrates the physical connections of this router (Figure 7).
Figure 5. Network of 400 devices.

Figure 6. Router298’s Network Layer.

Figure 7. Router298’s Neighbor Table.

The routing table illustrates the logical connections of this router.

Also, to verify the compliance of the network with the topology created by the use of the Barabashi-Albert model, networks of 500, 750 and 1000 nodes were generated. The average path length in a network with the Barabashi-Albert topology is determined by the formula:
\[ l = c \frac{\ln N}{\ln \ln N}, \]  

(1)

where \( c \) - coefficient of proportionality; \( N \) - network size (number of nodes); \( l \) - average path length (hops).

For the generated networks, we know \( N \), and it is also possible to determine the average path length \( l \). Further, by formula (1) it is possible to determine the coefficient of proportionality \( c \). All these data are summarized in Table 2.

| \( N \) | 500 | 750 | 1000 |
|-------|-----|-----|------|
| \( l \) | 4.3518 | 4.7367 | 4.8427 |
| \( c \) | 1.3533 | 1.3523 | 1.3548 |

From Table 2 it can be seen that the coefficient \( c \) is approximately constant. Differences begin only in the third decimal place. This indicates the correspondence of the generated networks with the Barabashi-Albert topology.

5.2 Generated network running

Next, simulations of the created networks were launched. To perform this simulation, each simulated computer was given the task of sending UDP packets to other computers on the network within 100 seconds. The simulation was successfully launched and completed, which indicated that the network files were being created correctly.
6. Conclusion
In this paper we have presented an overview of several models for scale-free networks. What we believe was a balanced choice was also made in favor of the Barabashi-Albert model.

The process of creating a simulation model for generating computer network topologies that implement the Barabashi-Albert algorithm is also described. We introduce a test of the health of the generated simulation model in relation to this topology.

The implemented Barabashi-Albert model has been presented in our project, BANG (Barabashi-Albert Network Generator). All the source code of the BANG project can be downloaded from the GitHub repository [17].

7. References
[1] OMNeT++ Discrete Event Simulator URL: https://omnetpp.org/ (23.01.2019)
[2] Enabling Open Innovation & Collaboration The Eclipse Foundation URL: https://www.eclipse.org/ (23.01.2019)
[3] Kazanskiy N L and Serafimovich P G 2012 Cloud computing for nanophotonic simulations Lecture Notes in Computer Science 7715 54-67 DOI: 10.1007/978-3-642-38250-5_7
[4] Kazanskiy N L, Kharitonov S I, Khonina S N, Volotovsky S G and Strelkov Y S 2014 Simulation of hyperspectrometer on spectral linear variable filters Computer Optics 38(2) 256-270
[5] The Opte Project 2017 LyonLabs LLC Barrett Lyon Retrieved URL: http://opte.org/maps/tests/
[6] Opte Project Wikipedia URL: https://en.wikipedia.org/wiki/Opte_Project/ (23.01.2019)
[7] Albert R and Barabási A L 2002 Statistical mechanics of complex networks Reviews of modern physics 74(1) 47
[8] Evaluating Network Simulation Tools FinMars Consulting Ltd URL: http://www.finmars.co.uk/blog/4-evaluating-network-simulation-tools/ (23.01.2019)
[9] OMNeT++ General 'Network' Simulation FinMars Consulting Ltd URL: http://www.finmars.co.uk/blog/8-omnet-general-network-simulation/ (23.01.2019)
[10] Veselý V, Bloudiček J and Ryšavý O 2014 August Enhanced interior gateway routing protocol for OMNeT++ 4th International Conference On Simulation And Modeling Methodologies, Technologies And Applications (SIMULTECH) 50-58
[11] Veselý V, Rek V and Ryšavý O 2015 Enhanced Interior Gateway Routing Protocol with IPv4 and IPv6 Support for OMNeT++ Simulation and Modeling Methodologies, Technologies and Applications 65-82
[12] Veselý V, Rek V and Ryšavý O 2016 Babel Routing Protocol for OMNeT++ - more than just a new simulation module for INET framework arXiv preprint arXiv:1609.05215

[13] Zhao Z, Mosler B and Braun T 2012 October Performance evaluation of opportunistic routing protocols: a framework-based approach using OMNeT++ Proceedings of the 7th Latin American Networking Conference 28-35

[14] Caldarelli G, Capocci A, De Los Rios P and Munoz M A 2002 Scale-free networks from varying vertex intrinsic fitness Physical review letters 89(25) 258702

[15] Bianconi G and Barabási A L 2001 Competition and multiscaling in evolving networks EPL (Europhysics Letters) 54(4) 436

[16] OMNeT++ SimulationManual URL: https://doc.omnetpp.org/omnetpp/manual/ (01.05.2019)

[17] The BANG project URL: https://github.com/Anoneth/BANG/ (29.06.2019)

Acknowledgements

This work falls within the public tasks allotted to the Ministry of Education and Science of the Russian Federation (2.974.2017/4.6).