

IMPLEMENTATION OF DEVS BASED DISTRIBUTED NETWORK SIMULATOR FOR LARGE-SCALE NETWORKS

Cobanoglu, B.; Zengin, A.; Ekiz, H.; Celik, F.; Kiraz, A. & Kayaalp, F.

Faculty of Technology, Sakarya University, 54167, Sakarya, Turkey

E-Mail: bcobanoglu@sakarya.edu.tr, azengin@sakarya.edu.tr, ekiz@sakarya.edu.tr,

fatihc@sakarya.edu.tr, akiraz@sakarya.edu.tr, fkayaalp@sakarya.edu.tr

Abstract

In this study, a tool was developed for design and implementation of a new discrete event based distributed simulation for large-scale networks. By using DEVS as a modelling approach, client / server based, scalable, platform-independent, flexible new network modelling and simulation tool called 'D-DEVSNET' was designed and performance analysis were conducted. In order to show the performance of the network simulation tool, D-DEVSNET and the power of the DEVS approach on parallel and distributed applications, some tests were carried out such as the measurement of throughput capability of different sized networks in a certain period of time, the average end-to-end packet delay and the amount of packet loss, consumption of speed and memory. According to the tests that were conducted, D-DEVSNET has been proven a well-scalable and high performing simulation tool.

(Received in June 2013, accepted in October 2013. This paper was with the authors 2 months for 1 revision.)

Key Words: Distributed Network Simulator, Client / Server Architecture, Large-Scale Network, Simulation Tools, DEVS, D-DEVSNET

1. INTRODUCTION

Modelling and simulation theory is an important tool for testing network design and protocols as well as understanding the dynamic behaviour of the network. Reaching to the applications that have run inside the global network by billions of people, as a part of their daily life regularly, has the network applications further elaborated. The growth of the biggest network, Internet's dazzling speed, requires new simulation techniques.

In the case of parallel and distributed applications, an additional level of complexity has to be addressed. The design of network components such as topological structure, protocols, traffic flow control, connection technology and routing algorithms is complex and difficult process. To reduce this complexity, the designers can benefit from modelling and simulation techniques in a certain level of abstraction [1].

Nowadays, computer networks have been used commonly in a wide range of fields for different purposes. In various sectors, network systems have been used by adopting them as service and activity fields, from public services to simple entertainment and chat networks. The necessity of rapid increases in power of performing operations and network speed has entailed routing and administration operations, to slide from a central structure towards to a distributed structure [2, 3].

In order to run a large-scale network simulation system, using only a single computer is rather expensive or impossible. Instead of this, by using more than one computer, designing a distributed system which runs in line with this objective, after assigning simulation components to different computers, is much cheaper and convenient [4]. Within this framework, with intent to form realistic and complex "virtual worlds" for high level interactive simulation activities, it can be provided to join simulation tools of different type and in different locations, with each other, by using distributed technologies (Client / Server, Java RMI, CORBA, HLA).

Nowadays, in large scale networks, it isn't possible to carry out an experiment so as to design more robust routing algorithms and test them, to determine the sensibility degree against errors and available simulation tools have remained insufficient for solving problems such as to prevent the network's collapsing against errors or to solve all the system behaviour. Therefore, it is obvious that researches must be focused on developing new techniques in order to model large-scale networks [5, 6].

In this paper, we developed a new network simulator which is based-on server/client architecture. This simulator has ability to work on distributed and parallel architecture using DEVS approach which has discrete event based, modular and hierarchical modelling approach. This new simulator is called as 'D-DEVSNET' which is developed on top of DEVS-Suite.

The developed network simulation tool can serve parallel and distributed. Also, it has web-support, open source and platform independent and the simulation results can be easily traceable at every stage of the simulation. Furthermore, this developed network simulation tool has a good visually which leads the simulation tool be usable mainly at distance education studies. In this scope, developed D-DEVSNET simulator is distributed, scalable, adaptive and it forms a template frame for powerful network application modelling / design, also indicated that DEVS (Discrete Event System Specification) approach has wide application area.

The rest of the paper is organized as follows. Section 2 describes DEVS formalism. Section 3 describes the component of D-DEVSNET. Section 4 and 5 show simulation parameters and performance evaluation. Finally, Section 6 provides conclusions.

2. DEVS FORMALISM

DEVS approach firstly has been put forward in the book 'Theory of Modeling and Simulation' in 1976 by Zeigler et al. for modelling and analysing of discrete event systems [6, 7]. DEVS is a simulation approach which is discrete event based, modular and hierarchic, and in recent times, it has come to the forefront more than the other approaches [8, 9].

DEVS approach has provided convenience in modelling complex large-scale systems owing to its supporting distributed study and a hierarchic/modular structure. In the developed parallel and distributed simulation algorithm, parallelism has been provided by using parallel DEVS M&S approach; in case distributed approach, it has been provided with server/client based architecture, and this algorithm has been used to develop D-DEVSNET network simulation tool.

3. THE COMPONENTS OF THE D-DEVSNET

The conceptual model includes all models and algorithms that will be employed to meet the defined software requirements. The conceptual model also includes all limitations that are to be made and how they may impact the ability of the model to satisfy the demands [10, 11].

The conceptual model of D-DEVSNET, together with network components within server and client model, is depicted in Fig. 1. Network components of server model are created by nodes / routers within autonomous systems (AS-1), server, experimental frame and topology generator. Client model network components are composed by nodes / routers within autonomous systems (AS-2), server, experimental frame and topology generator. At the same time, DEVS kernel is located in both client side and server-side.

A simple network model can be characterized / represented by only nodes and data lines. The nodes that have been formed towards to modelling purpose of DEVS based large scale network simulator; and primitive network components which provide communication of these

nodes have been defined as ‘atomic DEVS network model’. Afterwards, by getting together these atomic components, ‘coupled DEVS network model’ has been formed.

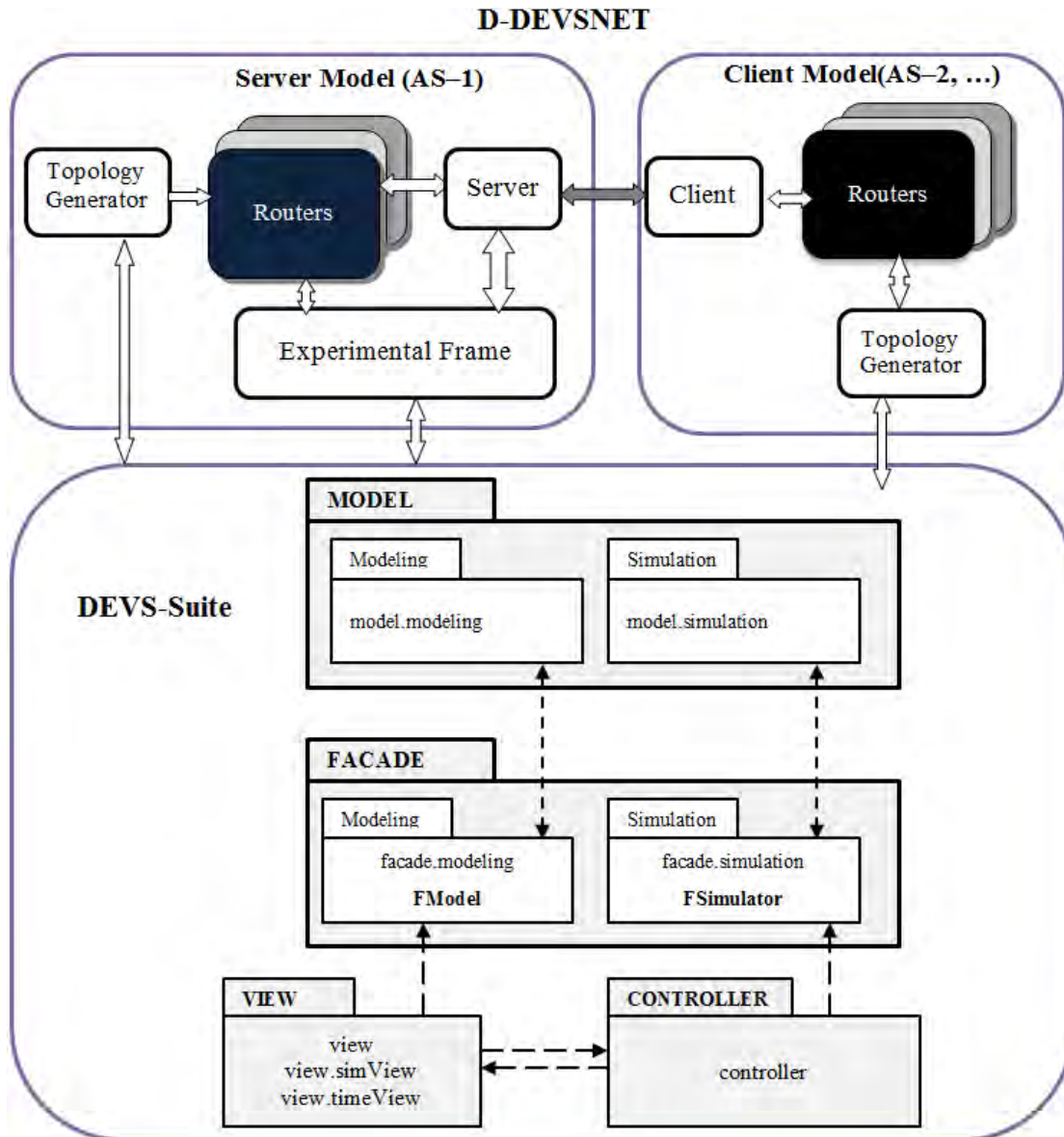


Figure 1: D-DEVSNET conceptual model and components.

The developed network model has three different types of atomic model, a coupled model and experimental frame. While the atomic node models have formed by server node atomic model, client node atomic model and routing atomic model; coupled node models have formed autonomous system. Furthermore, experimental frame model has also generator and transducer atomic models.

Every node in the developed network model has a duplex connection that allows to traffic flows in both directions and processing packets is modelled as a switching component which has the ability to be able to route them to suitable targets. The modelled nodes are atomic DEVS models which are connected to each other with two or more network connections.

Nodes and routers have a buffer memory that has capacity to be able to conduct the traffic and the speed of doing operations, a certain bandwidth in order process traffic. By configuring these properties, network units can be created which have different capacity values and network scenarios can be developed.

3.1 Server atomic node model

Server and client nodes are designed as atomic nodes which will provide opportunity to route traffic between autonomous and in a sense that will give transit passing service. Server atomic node model is designed as a router that has a duty to provide the communication with network client side. In the routing operation between the developed server and client nodes, BGP [12] protocol has been operated.

The only difference between server and client nodes is that it has experimental frame. Consequently, all the traffic results have been kept in a single file. Network Interface Card (NIC), DEVS entities, data and control packets have taken place inside server node. Server node has a socket structure that provides the connection with client, as well. Socket programming is an assembly programming technique and provides bidirectional communication between programs (client-server) which run on a network. In other words, they perform a channel function in order to be able to make application connections between each other, and to make their communications. A screenshot of the server atomic node model in the D-DEVSNET is depicted in Fig. 2.

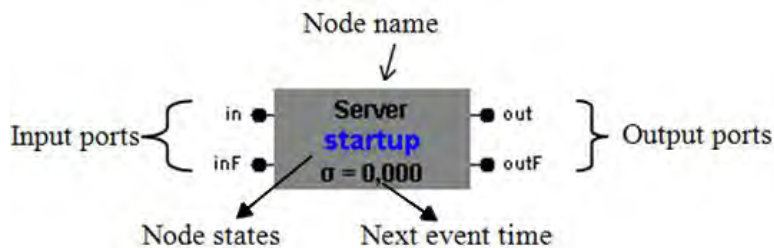


Figure 2: D-DEVSNET screenshot of the server atomic node.

3.2 Client atomic node model

Client node is designed as a router to provide the communication with network server-side. Inside the client node, NIC, DEVS entities, data and control packets have taken place. As expressed in the structure of server node, data transmission between client node and server, Java socket technology is used.

Server and client nodes have boundary node functions that connect autonomous systems with each other. The screenshot of a client atomic node model on the D-DEVSNET is depicted in Fig. 3.

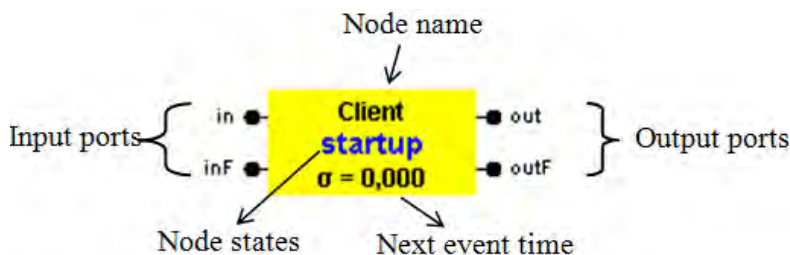


Figure 3: D-DEVSNET screenshot of developed client atomic node.

3.3 Atomic router model

Each atomic router model, routing module, NIC, DEVS entities, communication and control packets are included. Routing module is a unit where the routing has been made and runs OSPF [13] and BGP protocols. Each node uses data inside this routing module while it

transmits packets to target node. Routing module packet carries out a transaction by using routing tables and algorithms. Within the routing module, routing table, neighbour table and topology database are components. These routing tables reflect the present situation of the network. The screenshot of a router atomic model on D-DEVSNET is depicted in Fig. 4.

When more packets have come to router, it mounts the packets in a queue and when the queue gets full, it discards the coming packets. A typically Network Interface Card (NIC) has a very simple MAC protocol behaviour which performs the operation for taking incoming and outgoing messages into a queue. Since our main target is to test the developed model under large scale network conditions, implemented MAC protocol has been kept very simple.

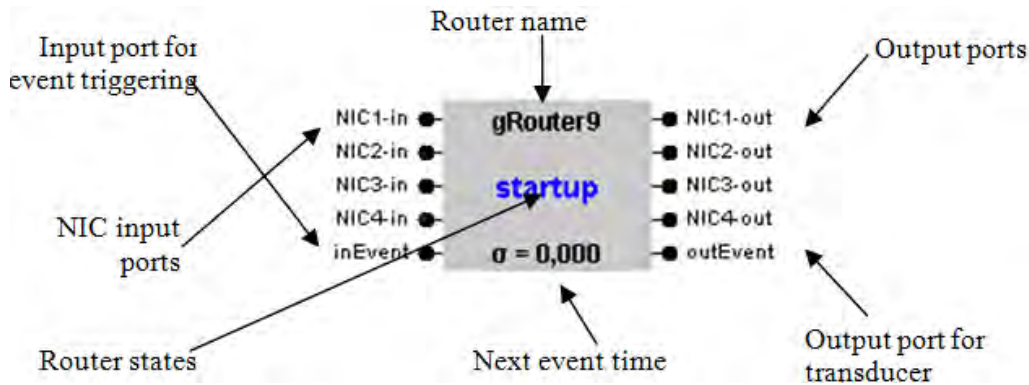


Figure 4: D-DEVSNET screenshot of the router atomic model.

OSPF protocol runs inside autonomous systems and BGP protocol runs among autonomous systems. Therefore, OSPF and BGP protocols, which are within routing module, have been executed. In the routing module, different protocols bundles (heaps) containing protocol administration and their organizations are maintained. For example, 'neighbour table', the information about neighbour table is kept. Routers have a visual Network Interface Card (NIC) for every neighbour. Control messages are formed by the messages which have 'hello', 'LSA' packets and BGP messages [14].

3.4 D-DEVSNET experimental frame

Experimental Frame (EF) defines the work of the system under certain conditions that are observed or experiment. In order to be able to test the system / the model in the simulation environment, it is necessary to form / define DEVS experimental frame concept. In our study, experimental frame is composed with two components.

Event Generator: It is a generator to give a triggering signal to system / model, to entry end of system/model. This generator is represented by 'EventGenr' model (see Fig. 5).

Event Transducer: In order to evaluate the coming results from system/model, it is connected to end points of system/model discharge (output) end. This transducer is represented by 'EventTransd' model. It has an essential function in obtaining significant results which will help in evaluating the network and analysis of obtained/produced outputs (packets) during the work of network.

The data belong to observed / traced network environment by event transducer at D-DEVSNET, both they can be followed instantaneously and they can be saved in a CSV (Comma Separated Values) file. Performance evaluation of the network and various graphics which are produced in its assessment can be obtained / drawn by making use of these saved data (Fig. 5).

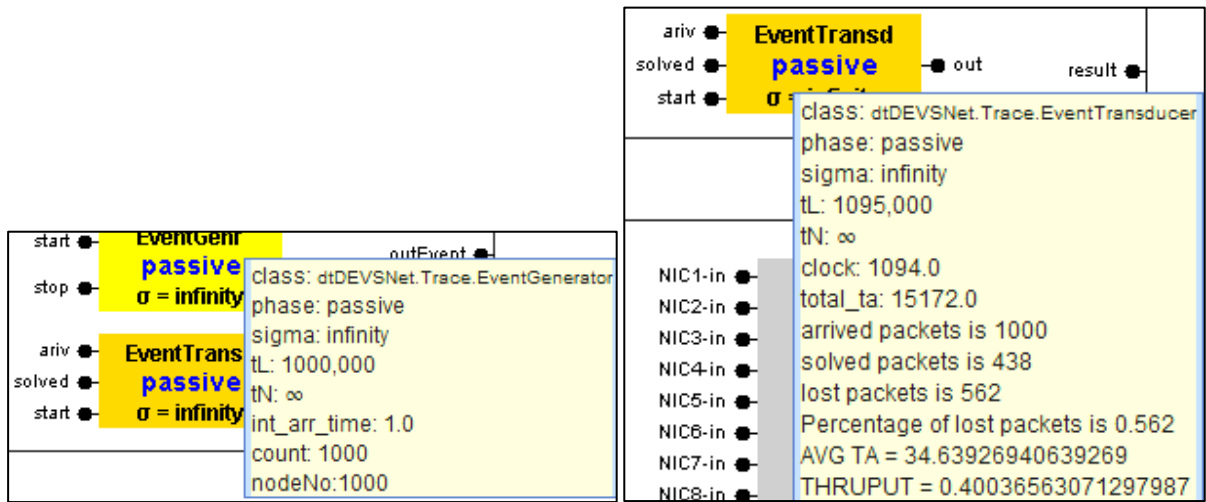


Figure 5: The observed data on experimental frame in a sample network which has 1000 routers.

Experimental frame and its connection with network components are depicted in Fig. 6. As seen in the Fig. 6, experimental frame is run on the server-side. While the autonomous system on the side of server has direct connection with the network components, its connection with the autonomous system on the side of client, becomes really over the server-client nodes indirectly.

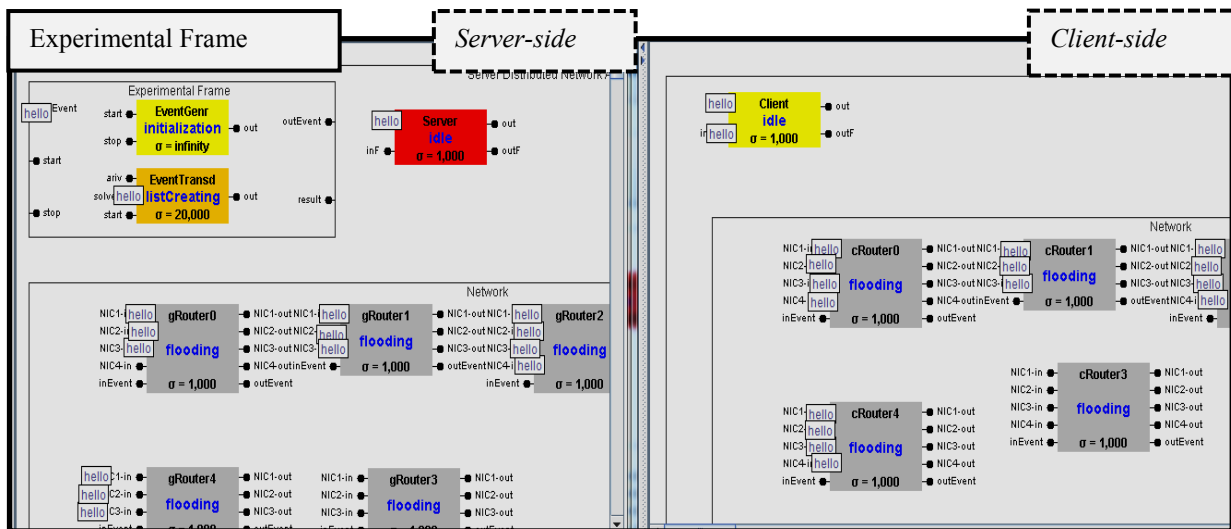


Figure 6: D-DEVSNET and the Experimental Frame.

3.5 Coupled models

Coupled models are made up of union of atomic components. Nodes and routers on the client-side and models on the server-side, are joined as autonomous systems, all network outputs are conducted and traced via experimental frame.

3.6 Autonomous system model

In Fig. 6, D-DEVSNET simulation environment is composed by two autonomous systems, on sides of client and server. But, more than two autonomous systems can be formed easily by

forming more clients when it is required. Autonomous systems are standalone systems in its entirety. Internet traffic infrastructure, while OSPF protocol is used within autonomous system, BGP protocol is used among autonomous systems. BGP is a router protocol which provides making decision on which routers should implement data traffic on the connection among autonomous system with the purpose to connect autonomous systems to each other.

In Fig. 7, D-DEVSNET autonomous system coupled model is depicted. Server and client nodes are boundary nodes which connect autonomous systems to each other. BGP protocol provides the data/packet traffic between these nodes.

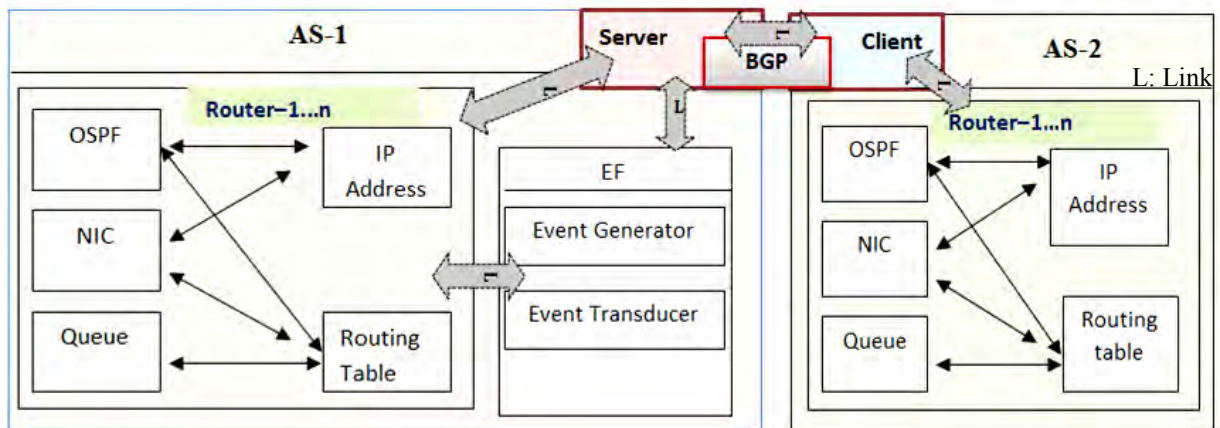


Figure 7: D-DEVSNET autonomous system coupled model.

4. SIMULATION PARAMETERS

In this study, as a result of scenarios that were produced under different scales, different topologies and different network traffics, the obtained data and observed outputs were offered. The results have shown that the D-DEVSNET simulator has ability to model large scale network.

Computers were run as client and the other as server on distributed architecture. By predicating on parameters which are their simulation experiments explained in Table I, simulation results were tested with JRE 1.7, on the computers that have Intel Core i7, 1.73GHz, 4GB RAM, 64 bit Linux and Intel Core i5, 2.0GHz, 2GB RAM, 64 bit Linux.

During the realized simulation process, the parameters that belong to simulation environment, the obtained data (statistics, average network output, packet loss, etc.) and information belonging to transmitted message in the network, were interpreted by event transducer and they were saved in the file (output.csv) that has a file name extension CVS (comma separated values) and were offered with graphics.

In Table I, developed D-DEVSNET simulation parameters and real world parameters were compared. For example, queue length, delay and bandwidth were chosen same as Internet. Especially, it was tried to choose IP address, protocols and data packets as in real world. However, it will not be practical to expect all parameters of simulation environment same with real world parameters. By using these parameters, D-DEVSNET simulator performance evaluations were done.

Table I: The applied simulation parameters in experiments.

Simulation parameters	D-DEVSNET	Real world
Exterior routing protocol	BGP	BGP
Interior routing protocol	OSPF	OSPF, RIP
IP Address model	IPv4	IPv4, IPv6
IP Address size (byte)	4	4
Message type	Modified IP Packet	IP Packet
MTU (byte)	452	576 for IPv4, 1280 for IPv6
Packet title size (byte)	20	20
Max. LSA packet life	Infinite	1 hours
Hop count	15	15
Node process time	4.5 Mb/s	1 Mb/s - 40 Gb/s
Queue length (packet)	362	32-500
Link delay (msec)	3	3-16
AS number	2	~40000
Scale	~5.000 routers	~900.000 nodes
Simulation time (sec)	1000	-

5. PERFORMANCE EVALUATION

Performance analyses of D-DEVSNET were carried out according to various scales having 50, 100, 200, 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000 and 4500 routers.

5.1 Network throughput

In the operation of testing routing algorithms, in unit time, throughput graphic which shows the number of incoming-outgoing packets in a network can be used as an important criterion. Throughput graphic that belongs to a distributed network and has different node/router number (50, 100, 200, 500, 2000, 3500, 4000, 4500), can be seen in Fig. 8. In the figure, it can be observed that while the router number increases, throughput increases gradually. Particularly, while the throughput is less in networks having up to 3000 nodes, the throughput increases for equal to or greater than 3500 nodes. The reason is that besides used traffic in different topologies become distinct, the capacity of memory and processor remains same. Naturally, increasing in operation capacity of network leads to increase of throughput, as well.

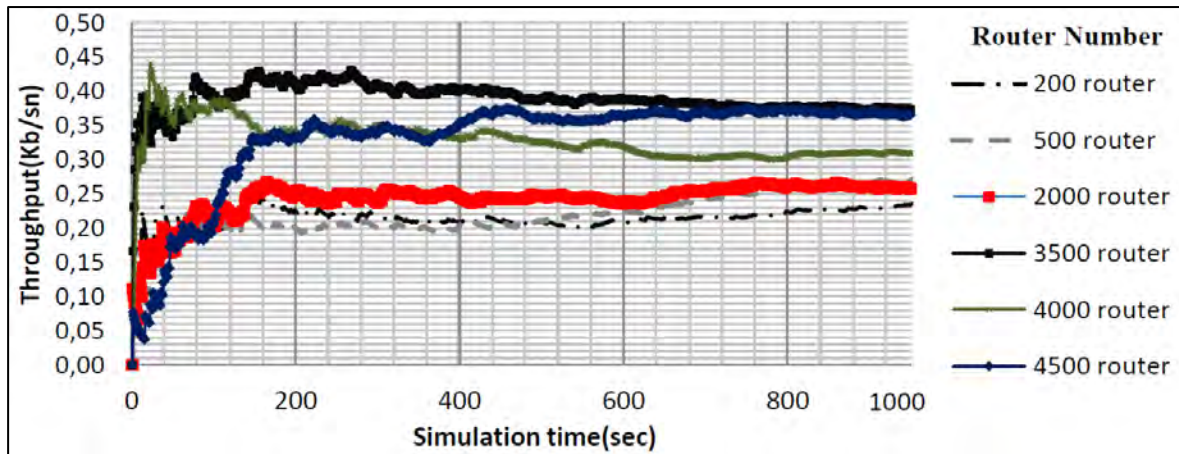


Figure 8: D-DEVSNET throughput in different scales.

5.2 End-to-end average packet delay

End-to-end average packet delay is the average travel-time of data packets outgoing from source node to target nodes. The time difference is determined for data packets which send from every node in network, as difference between outgoing time from source and incoming time to target node. By taking the average of determined time differences, end-to-end packet delay is calculated.

In Fig. 9, end-to-end average packet delay of D-DEVSNET simulation in different scales networks at 1000 seconds time period was given. Accordingly, it can be seen that while network scale is getting bigger, a remarkable loss of performance has not been observed.

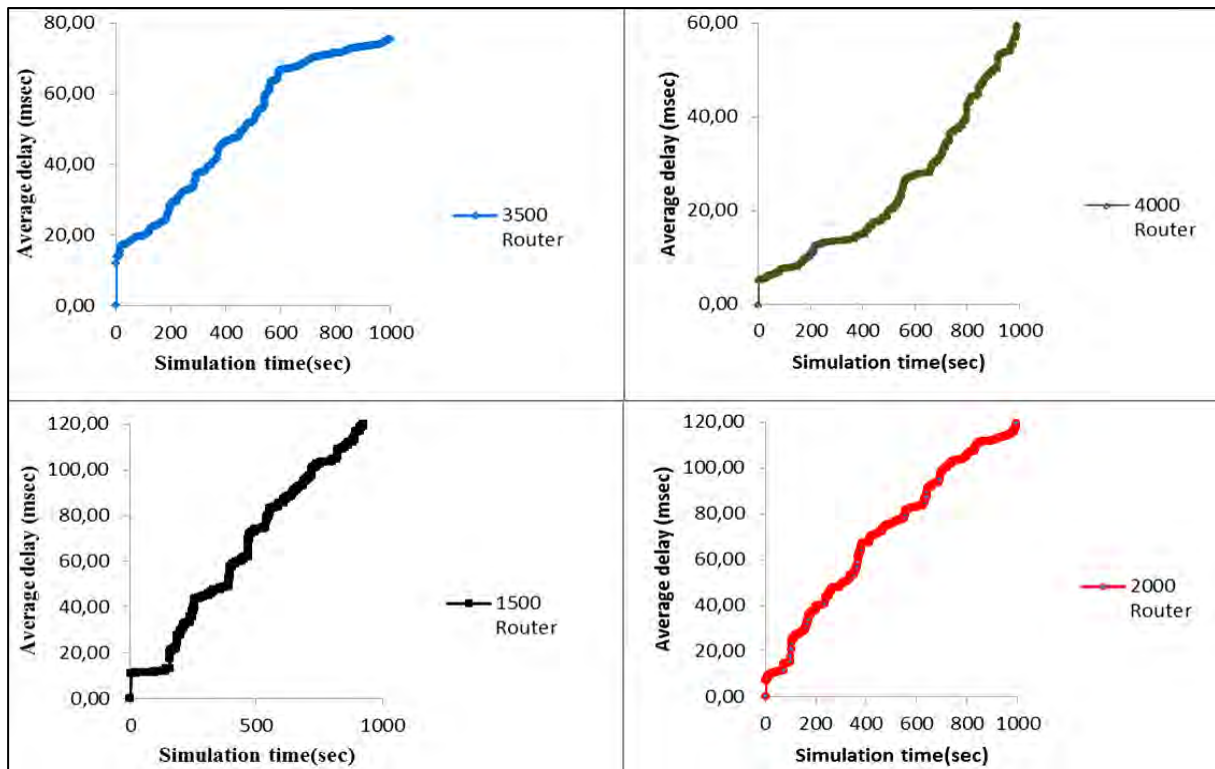


Figure 9: In different scales D-DEVSNET end-to-end average delay.

5.3 Comparison of network simulation tools

There are lots of studies directed to the comparison of speed and memory consumption of network simulation tools. In the literature, it has been done comparisons of speed and memory consumption that belong to distinct network simulation tools. Nicol [15] had compared ns-2 [16], JavaSim (J-Sim) [17] and SSFNet [18] simulation tools considering the connection number (10-10000) in duration of 1000 seconds. In Table II, memory consumption per connection was given. Nicol had determined that ns-2 simulation tool has quicker speed but further memory consumption.

Table II: Comparison of speed and memory consumption of network simulation tools.

Simulator	Memory consumption per connection
ns-2	93,3 KB
J-Sim	21,7 KB
SSFNet	53,3 KB
D-DEVSNET	42,8 KB

By Weingartner et al. [19], ns-2, ns-3 [20], OMNeT++ [21], SimPy [22] and JIST [23] simulation tools had been compared by considering nodes numbers (4-3025 nodes) in 600 second duration. At the end this study, it was emphasized that ns-2, OMNeT++ and SimPy tools have similar memory consumption; JIST has quicker but more memory consumption.

By Albeseder et al. [24], OMNeT++, ns-2, QualNet [25] and SimPy tools were compared by considering their working time durations. Also, in this study, the similar results were obtained and it was particularly emphasized that the performance of OMNeT from the point of speed was not good.

By Fujimoto et al. [7], ns-2, PDNS [26] and GTNeTS [27] simulation tools were compared and it was indicated that ns-2 works quicker.

From all these studies, we can say that simulation tools which work quicker need more memory capacity. As seen from Table II, memory consumption amount per connection while D-DEVSNET simulator (while 'hello' packets were circulated in network), is better than ns-2 and SSFNet. Nevertheless, in tests we have performed, in one computer while it was scaled at most 1500 nodes, with D-DEVSNET, it can be scaled up to about 3000-3500 nodes. This shows that D-DEVSNET has a better scalability.

A comparison was made between D-DEVSNET and other network simulation tools which are tested. Summary of the network simulation tools and their strengths and weaknesses are presented in Table III. In line with different properties / purposes, it can be said that there is not a perfect simulator tool, when the simulators in the table are examined. Furthermore, each simulator tool developed or was developed in the direction of a different purpose. For instance, it was seen that in SSFNet, large scale Internet traffic; ns-2, wireless network simulation studies have become prominent. D-DEVSNET, large scale network applications are preferable for education and research purpose.

Table III: Comparison of network simulation tools.

Aspect	ns-2	J-Sim	SSFNet	D-DEVSNET
Purpose	Education, research	Education	Commercial	Education, research
Network Topology	Meso-Scale	Meso-Scale	Large-Scale	Large-Scale
Object-orientation	M	VS	VS	VS
Network Models Library	S	M	W	W
Analysis of the results	M	W	W	VS
Extendibility	M	VS	VS	VS
Expertise need	VS	W	S	M
Deployment	W	VS	S	VS
Documentation	M	W	W	M
Availability	VS	VS	VS	VS
Visualization	W	M	VS	S
Ease of use	W	W	W	S
Scalability	W	S	VS	VS
Performance	S	W	VS	VS
Web access	-	S	-	S

M – medium, S – strong, VS – very strong, W – weak.

6. CONCLUSIONS

In this study, a new network simulation tool having discrete event base that will be able to work in large scale networks has developed. The name 'D-DEVSNET' was given to this developed simulation tool and its validity was tested by the other network simulation tools.

In the study of Begg et al. [28], it is emphasized that simulation tools which were developed for special-purpose, have to meet user demands. Also, by customizing DEVS-Suite

tool in the direction of certain targets for network simulations, it can be transformed to a tool that can meet user requirements.

Developed distributed simulator, its web-supported ability to work, platform independence in every stage easy traceability of simulation results, its visuality, its easy setting up were targeted for educational usage. Within this scope, developed D-DEVSNET simulation tool, has formed a sample frame for modelling of large scale network applications, by providing opportunity to be made healthy and easy simulation studies for large scale networks that have distributed, scalable, adaptive, different topologies, and it was showed that DEVS approach has a large field of application.

To conclude, it was seen that developed D-DEVSNET network simulator provides opportunity to make healthy and easy simulation studies on a large scale network which has classic heterogeneous structure. In addition, it was seen that the performance of the simulation tool was not much affected by the network node / router number (increase or decrease of) and the hardware capacity.

REFERENCES

- [1] Rahman, M. A.; Pakstas, A.; Wang, F. Z. (2009). Network modelling and simulation tools, *Simulation Modelling Practice and Theory*, Vol. 17, No. 6, 1011-1031, [doi:10.1016/j.simpat.2009.02.005](https://doi.org/10.1016/j.simpat.2009.02.005)
- [2] Cobanoglu, B.; Zengin, A.; Tuncel, S.; Ekiz, H. (2010). A Survey of Network Modeling and Simulation Tools: DEVS Comparison, *ISSD 2010, Second International Symposium on Sustainable Development*, Sarajevo, 418-424
- [3] Fujimoto, R. M.; Perumalla, K. S.; Park, A.; Wu, H.; Ammar, M. H.; Riley, G. F. (2003). Large scale network simulation: how big? how fast?, *11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems (MASCOTS 2003)*, 116-123, [doi:10.1109/MASCOT.2003.1240649](https://doi.org/10.1109/MASCOT.2003.1240649)
- [4] Vickovic, L.; Celar, S.; Mudnic, E. (2011). Disk array simulation model development, *International Journal of Simulation Modelling*, Vol. 10, No. 1, 27-37, [doi:10.2507/IJSIMM10\(1\)3.174](https://doi.org/10.2507/IJSIMM10(1)3.174)
- [5] Zeigler, B. P.; Mittal, S. (2001). Modeling and Simulation of Ultra Large Networks: A Framework for New Research Directions, *Workshop on Modeling and Simulation of Ultra Large Networks: Challenges and New Research Directions*, Tucson, Arizona, 1-34
- [6] Zeigler, B. P.; Praehofer, H.; Kim, T. G. (2000). *Theory of Modeling and Simulation*, Second Edition, Academic Press, San Diego
- [7] Fujimoto, R. M.; Perumalla, K. S.; Riley, G. F. (2007). Network simulation, *Synthesis Lectures on Communication Networks*, Morgan & Claypool Publishers, Vol. 1, No. 1, 1-72, [doi:10.2200/S00046ED1V01Y200609CNT001](https://doi.org/10.2200/S00046ED1V01Y200609CNT001)
- [8] Zengin, A.; Sarjoughian, H.; Ekiz, H. (2013). Discrete event modeling of swarm intelligence based routing in network systems, *Information Sciences*, Vol. 222, 81-98, [doi:10.1016/j.ins.2011.06.012](https://doi.org/10.1016/j.ins.2011.06.012)
- [9] Raczynski, S. (2012). Semi-discrete events and models in categorical language, *International Journal of Simulation Modelling*, Vol. 11, No. 2, 89-96, [doi:10.2507/IJSIMM11\(2\)3.203](https://doi.org/10.2507/IJSIMM11(2)3.203)
- [10] Taylor, S. J. E.; Balci, O.; Cai, W.; Loper, M. L.; Nicol, D. M.; Riley, G. (2013). Grand challenges in modeling and simulation: expanding our horizons. *Proceedings of the 2013 ACM SIGSIM conference on Principles of advanced discrete simulation*, Montreal, 403-408, [doi:10.1145/2486092.2486151](https://doi.org/10.1145/2486092.2486151)
- [11] Kehris, E. (2009). Web-based simulation of manufacturing systems, *International Journal of Simulation Modelling*, Vol. 8, No. 2, 102-113, [doi:10.2507/IJSIMM08\(2\)4.126](https://doi.org/10.2507/IJSIMM08(2)4.126)
- [12] McPherson, D., BGP-4 Protocol (RFC 4277), from <http://tools.ietf.org/rfc/rfc4277.txt>, accessed on 05-06-2013
- [13] Moy, J., OSPF Version 2 (RFC 2328), from <http://www.ietf.org/rfc/rfc2328.txt>, accessed on 05-05-2013

- [14] Zengin, A.; Sarjoughian, H. (2010). DEVS-Suite simulator: A tool teaching network protocols, *Proceedings of the 2010 Winter Simulation Conference (WSC)*, Baltimore, 2947-2957, doi:10.1109/WSC.2010.5678989
- [15] Nicol, D. M. (2003). Scalability of network simulators revisited, *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS'03)*, Orlando, 8 pages
- [16] The Network Simulator – ns-2, from <http://www.isi.edu/nsnam/ns/>, accessed on 10-09-2013
- [17] J-SIM Official, from <https://sites.google.com/site/jsimofficial/>, accessed on 03-06-2013
- [18] SSFNet. Scalable Simulation Framework, from <http://www.ssfnet.org/homePage.html>, accessed on 05-05-2013
- [19] Weingartner, E.; vom Lehn, H.; Wehrle, K. (2009). A performance comparison of recent network simulators, *IEEE International Conference on Communications (ICC'09)*, Dresden, 1-5, doi:10.1109/ICC.2009.5198657
- [20] Ns-3. The ns-3 network simulator, from <http://www.nsnam.org/>, accessed on 10-09-2013
- [21] OMNeT++, OMNeT++ Network Simulation Framework, from <http://www.omnetpp.org/>, accessed on 10-09-2013
- [22] SimPy documentation, from <http://simpy.readthedocs.org/en/latest/>, accessed on 10-09-2013
- [23] Barr, R.; Haas, Z. J.; van Renesse, R. (2005). JiST: An efficient approach to simulation using virtual machines, *Software: Practice and Experience*, Vol. 35, No. 6, 539-576, doi:10.1002/spe.647
- [24] Albeseder, D.; Fugger, M.; Breitenecker, F.; Loscher, T.; Taubock, S. (2005). Small PC-network simulation – A comprehensive performance case study, *Simulation News Europe SNE 44/45*, Vol. 15, No. 2-3, 26-32
- [25] QualNet (Scalable Network Technologies), from <http://web.scalable-networks.com/content/qualnet>, accessed on 05-05-2013
- [26] PDNS – Parallel/Distributed NS, from <http://www.cc.gatech.edu/computing/compass/pdns/>, accessed on 05-05-2013
- [27] Riley, G. F. (2003). Large-scale network simulations with GTNetS, *Proceedings of the 2003 Winter Simulation Conference*, New Orleans, Vol. 1, 676-684
- [28] Begg, L.; Liu, W.; Pawlikowski, K.; Perera, S.; Sirisena, H. (2006). *Survey of simulators of next generation networks for studying service availability and resilience*, Technical Report TR-COSC 05/06, University of Canterbury, Christchurch