Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2013) 21: 1720 – 1736

doi:10.3906/elk-1112-68

Discrete event simulation-based performance evaluation of Internet routing protocols

Fatih ÇELİK, Ahmet ZENGİN,* Bülent ÇOBANOĞLU
Department of Computer Engineering, Faculty of Technology, Sakarya University, Sakarya, Turkey

Received: 20.12.2011 • **Accepted:** 24.05.2012 • **Published Online:** 02.10.2013 • **Printed:** 28.10.2013

Abstract: This paper presents a discrete event system specification (DEVS)-based comparative performance analysis between the open shortest path first (OSPF) protocol and the routing information protocol (RIP), together with the border gateway protocol (BGP), using DEVS-Suite. In order to evaluate the OSPF and RIP's scalability performance, several network models are designed and configured with the OSPF and RIP, in combination with the BGP.

Evaluations of the proposed routing protocols are performed based on the metrics, such as the execution time, convergence time, turnaround time, throughput, and efficiency across an increasing size and complexity through the simulated network models. The evaluation results show that the OSPF routing protocol provides a better scalability performance than the RIP routing protocol for Internet applications.

Key words: DEVS, DEVS-Suite, OSPF, RIP, BGP, Internet

1. Introduction

The primary aim of routing is to transmit data packages in a network [1]. The main task in implementing routing belongs to the routing protocols. Although there are several types of routing protocols, their basic functions (such as network monitoring, routing, the shortest path calculation, etc.) are the same. The fundamental property that separates these protocols from each other is how they are informed about the changes in the network and/or how they calculate the shortest path for each direction. The routing protocol is a set of rules that prompts the network traffic and finds the most suitable path. Routing protocols are either interior gateway protocols (IGPs) or exterior gateway protocols. The routing information protocol (RIP), open shortest path first (OSPF) protocol, and enhanced interior gateway routing protocol (EIGRP) are examples of IGPs, while the border gateway protocol (BGP) and border gateway protocol version 4 are examples of BGPs.

Routing is one of the very crucial functions of large-scale network systems such as the Internet [2]. The BGP [3], RIP [4], and OSPF [5] protocols are the most commonly used protocols in the Internet today. In the traffic substructure of today's Internet, while the OSPF and RIP protocols are used within autonomous systems (ASs), the BGP protocol is used between the ASs. It is very obvious that the Internet will continue to be improved, even if it is developed on a complex and scalable basis. Because it is impossible to do operational work on the Internet system in the real world, modeling and simulation enable us to benefit in such a way that it can be studied on a single computer, and thus not require the Internet as a testbed [6]. The main goal of this study is to investigate the consequences of deploying the RIP, OSPF, and BGP on an Internet scale network.

*Correspondence: azengin@sakarya.edu.tr

In the literature [1,7–11], performance analyses related to routing protocols are generally focused on small-scale networks. Our study aims to test routing protocols on a large-scale network, and to reveal the strong and weak aspects of these protocols. In our study, it is shown that the OSPF is quite successful in comparison with the RIP in terms of learning path information expeditiously (rapid convergence), the ability to work better on large and complex networks, and reliability.

Scalability is a very important property for Internet systems. There are rising demands for studies on scalable routing approaches in current Internet systems [12]. However, there are a lack of formal approaches and efforts to survey the scalability aspects of the existing routing protocols to improve their performance. Formal approaches improve the correctness of the routing protocols and modeling tools [13]. Nonformalized approaches result in low performance, bad scalability, difficulty with validation and experiments, and nonreusable products [14]. Among the discrete event methodologies, discrete event system specification (DEVS) formalism [15] enables the modeler to build highly formalized, system theoretic, and scalable network systems [14,16].

In this paper, we developed DEVS-based protocol models for the RIP, OSPF, and BGP for addressing the main challenges in the scalability of these protocols on the Internet. During the experiments, DEVS-based simulator software, known as DEVS-Suite [17], is successfully manipulated as domain-neutral simulation software. Apart from that, it offers an easy graphic and trackable interface, and additionally provides an extensive library of components for more reliably and efficiently designing a simulation model. The developed DEVS-Suite OSPF and RIP frameworks provide visualization, advanced tracking capability, reusability, and a component-based model design. The research also investigates how well the OSPF and RIP protocols respond to various Internet performance metrics, such as execution time, efficiency, convergence, end to end delay, and throughput. Such analysis is important since it facilitates the determination of the most suitable and robust Internet protocols in a proposal to optimize the traffic goals on the Internet. The research also examines and compares the routing performance of the protocols under a variety of network conditions and scales. A number of important system parameters, such as the network size, number of nodes per AS, number of links per node, and topology of the model, are taken into consideration. Changes in such parameters are made to realize the different realistic Internet scenarios, as well as to evaluate the extent of their impact on the network and routing protocols' performances, such as scalability.

The rest of the paper is organized as follows: in Section 2, we briefly review the background of the DEVS formalism and routing protocols. In Section 3, the network model, protocol models, and auxiliary tools are briefly discussed. Section 4 presents the experimental models and experiments. An analysis of the simulation experiments and an assessment of the approach are presented in Section 5. The paper concludes with future directions and conclusions in Section 6.

2. Background

2.1. DEVS formalism

DEVS formalism is a novel approach for modeling dynamic systems and was developed by Zeigler [15]. In DEVS, time passes only when a new event occurs. This approach is well suited for formally describing concurrent processing and the event-driven nature of the arbitrary configuration of networked components. This modeling approach supports the system's theoretic, formalized, hierarchical, and modular model construction, and distributed execution, and therefore characterizes complex, large-scale systems with atomic and coupled models. The following properties are the main reasons to make use of DEVS formalism in our network modeling effort:

- An event-based efficient approach: The DEVS approach is developed as an extension of the Moore machine formalism [18]. DEVS formalism is a kind of finite state automaton approach in which the states and outputs are created by the current state and the inputs. Contrary to other finite state approaches, DEVS has brought new concepts to the simulation community, such as the state lifespan and hierarchy. These innovations have rendered it possible to create very efficient simulation software.

- Strong couplings among the components: In the DEVS component connection architecture, a basic entity is a component called an atomic model. Each atomic model has one or more communication ports. Moreover, 2 atomic models are connected by "wiring" their ports together. DEVS formalism relies on network models to facilitate the entity or message movement and decision making.

- Modular and hierarchical design: The DEVS autonomous component architecture yields to the design of highly modular and hierarchical systems. This scheme provides a manageable system across an increasing size and complexity. In particular, for large-scale network applications, this property is needed to design scalable methods. In DEVS implementations, models, their simulators, and their experimental frames are all distinct entities, and their software representations are also modular.

- Object-orientation: Object-oriented design is today's most popular programming paradigm. Object-orientation yields simpler, concrete, robust, flexible, and modular software implementations. DEVS exploits these properties to create a successful and manageable simulation application.

- Easy to abstract: DEVS provides abstraction and simplification mechanisms, as well as model objectives. When modeling ultimately complex and dynamic systems, simplification mechanisms become important for the sake of building scalable systems.

- Automated parallel/real-time execution: DEVS can manage parallelism [19]. It has a specific function for parallel-occurring events, which orders the events according to some criteria. DEVS also supports real-time execution, where DEVS-based simulators can perform single-host, distributed, and real-time executions.

- Interoperation and reuse: DEVS simulators can run over various middlewares, such as message passing interface, high-level architecture, and common object request broker architecture [20]. In a typical DEVS design, a variety of model components can be reused. This may reduce the development times and enable the focus on higher levels. Model repositories and experimental frames can be created and maintained, and they are ready to reuse.

The developed DEVS network models can be executed using the DEVS-Suite simulation engine [21]. As an extension of DEVSJAVA, DEVS-Suite is an object-oriented realization of parallel DEVS and its associated simulators. The main components of the simulation package are simview, DEVS tracking environment, and timeview [17].

2.2. Routing protocols for Internet

Routing protocols in the network systems can be split into 2 main categories: link state routing and distance vector routing [2]. Currently, in particular for the Internet, link state protocols are used for intranets, while distance vector protocols are used for intergateway interactions [3].

2.2.1. Routing information protocol

The RIP is a widely deployed routing protocol for the Internet that is based on the distance vector routing algorithm [4]. Having an open framework and easy to deploy architecture, the RIP has become popular in today's network devices; however, it does not meet the requirements of the current large-scale network systems [2].

In the RIP, the router sends its entire routing table (which lists all of the other hosts that it knows about) to its closest neighbors, every 30 s. From neighbor to neighbor, finally the routing information messages reach every router in the network and all of the routers within the network have the same knowledge of the routing paths. This process is called network convergence. The RIP uses a hop count as a way to determine the network's distance, but it can use other metrics such as link bandwidth and delay. Consequently, each router in the network uses the routing table information to determine the next hop to route a packet to for a specified destination [4].

The RIP can be seen as an effective solution for small homogeneous networks. For large-scale and more complex networks, computing the overhead of the RIP can generate a heavy amount of extra traffic in the network. Therefore, in that case, an alternative to the RIP is the OSPF protocol [5].

2.2.2. Open shortest path first protocol

The OSPF, as one of the famous link state routing protocols, is an open standard routing protocol and a particularly efficient IGP that is faster than the RIP, which is one of the most well-known kinds of distance vector protocols [5]. It uses the Dijkstra algorithm when estimating the shortest paths [22].

The OSPF routing protocol was developed to provide an alternative to the RIP, based on shortest path first algorithms instead of the Bellman–Ford algorithm, which is the basis for the RIP [23]. It uses a tree that describes the network topology to define the shortest path from each router to each destination address. The OSPF addresses all of the deficiencies of the RIP, without affecting the connectivity to the RIP-based networks. Fast growing and large-scale networks must be designed properly if the capabilities of the OSPF are to be fully exploited. Because of its ability to handle variable networking masks, the OSPF also helps to reduce the waste of today's precious internet protocol (IP) addresses. The OSPF will enable networks to scale to very large topologies, while maintaining high levels of availability and performance. The main difference between the OSPF and the RIP is that the RIP only keeps track of the closest router for each destination address, while the OSPF keeps track of a complete topological database of all of the connections in the local network.

The OSPF algorithm logic is based on 3 phases [16]. They are important for describing how the algorithm behaves in a discrete event fashion. In the startup phase, as soon as a router connects to the network, it sends Hello packets to all of its neighbors, receives their Hello packets in return, and establishes routing connections by synchronizing databases with neighboring routers that agree to synchronize. In the update phase, each router sends an update message at regular intervals. This message is called 'link state', describing its routing database to all the other routers, so that all of the routers have the same description for the local network topology. In the final phase, each router estimates a mathematical data structure called a 'shortest path tree' that describes the shortest path to every destination address, indicating the closest router for communication.

2.2.3. Border gateway protocol

The BGP is an interAS routing protocol that provides exchange network reachability information to other ASs [2]. This exchanged information contains the list of ASs that the BGP packet traverses. Using this information, a graph of the AS's connectivity is constructed, by which routing decisions can be made according to the distance vector algorithm.

2.3. Simulation model

In this section, simulation model components, including the DEVS network definition, protocol models, traffic generators, and topology generator, are detailed. On developing these components, a large-scale network simulation framework is established to analyze the large-scale characteristics of the protocols.

2.4. Network model

In order to evaluate the performance of the routing protocols, a network model is developed on top of the DEVS-Suite simulation viewer. The developed simulator has a hierarchical and modular design inherited from its underlying DEVS modeling formalism. Being an object-oriented implementation of the DEVS formalism, DEVS-Suite has a high-level user interface that is built using Java [17]. The hierarchical structure of the developed simulator is divided into 3 levels. These are the network application level, node level, and event process level.

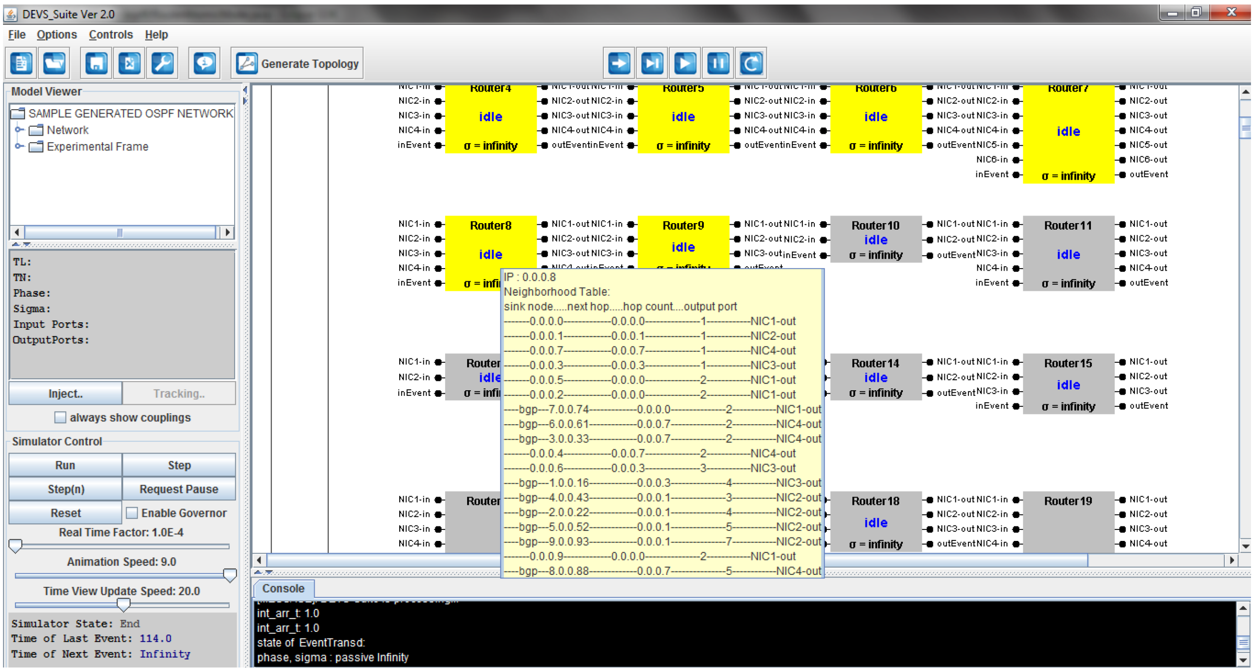


Figure 1. RIP-DEVS network model in the DEVS-Suite simulation viewer.

2.4.1. Network application level

The network model is generalized to model a wide range of network technologies in a single model. First of all, devices belonging to the first layer of the 7-segment open system interconnection layers are modeled. These hardware devices are links, queues, network interface cards, and the nodes. DEVS formalism and its associated tools are selected for modeling such a distributed system since DEVS enables the modeler to specify systems in a system theoretic manner and yields hierarchical and modular developments, facilitating the model distributed systems with intelligent components and overcoming the complexity. The DEVS definitions of the network model can be found in [16] and [24].

In this level, coupling between the nodes/routers, interconnection, and configuration, which can be done via the experimental frame and topology generator components, can be included in the network model. The complete network model represents the overall system to be simulated. In Figure 1, a network model of the RIP-DEVS model is shown with a RIP/BGP routing table visualization in the DEVS-Suite simulation viewer.

2.4.2. Node level

The node level is an internal infrastructure of the network level AS. Since a typical networked system can only be characterized as nodes and links, we begin to develop the network model by specifying these basic components using a parallel DEVS atomic model. Thanks to the DEVS component structure flexibility, a node can be the routers, workstations, satellite, and so on. A block view of the developed OSPF and RIP network system, together with its experimental frame, is presented in Figure 2, where it can be seen that a node includes several databases, including the link state advertisement (LSA) history and tables to be used by the protocols and it supports different routing protocols such as the RIP, OSPF, BGP, and BEE [25]. These stacks support the protocol management and organization. The OSPF-DEVS model details can be found in [24] and its large-scale characteristics can be found in [16]. The node has an underlying operating mechanism called a DEVS kernel, through which discrete event logic can be implemented. The routing module is the most important part of the node and it is also the most detailed part of the node model, since the main concern is to test the routing protocols under large-scale experimental conditions. Nodes also have network interface cards, in which a simple MAC protocol is implemented. A typical interface has a queue for the incoming and outgoing packets, which is simply a drop-tail queue. Nodes can originate from control packets such as hello, LSA or RIP, and acknowledgment packets.

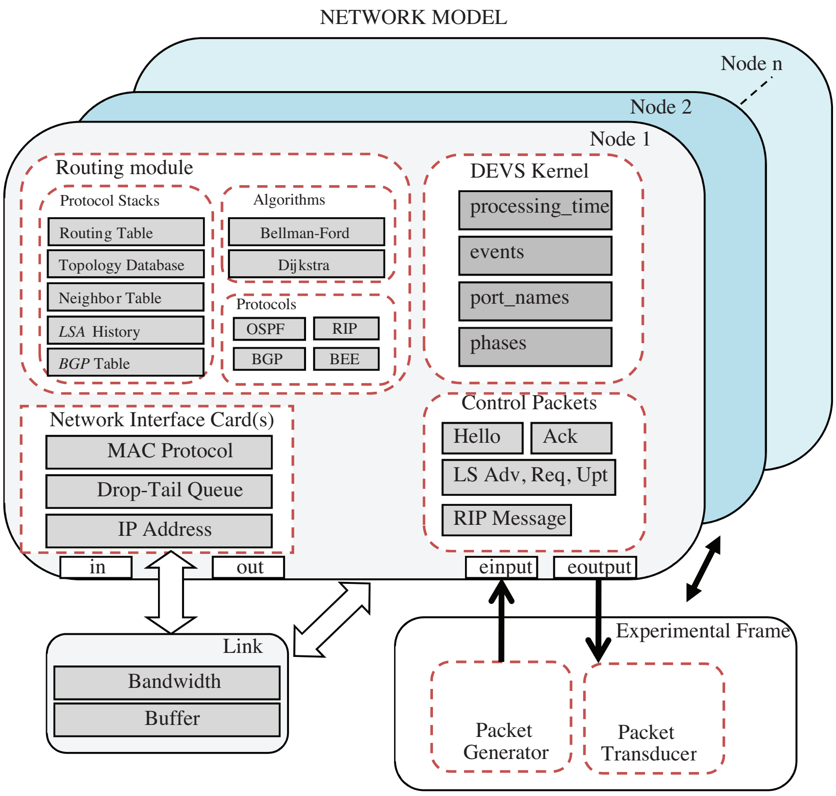


Figure 2. Node level abstraction.

2.4.3. Process level

The process level is used to specify the attributes of the node model using the DEVS model specifications and source code Java that is inside of the node models. Figure 3 depicts a node's state chart, in which the states are only chanced via the internal and external transitions. As seen in Figure 3, a node's OSPF behavior is abstracted to 10 states to mimic the OSPF protocol behavior. For example, if a node is in an 'idle' state, it receives a packet and its queue is full, and then the state changes to 'congested'.

2.5. Protocol models

After modeling the basic components of the network system, such as the nodes and links, the OSPF, RIP, and BGP behaviors are implemented into the instrument network with routing capability and intelligence. As already mentioned in Section 2, the OSPF and RIP protocols are selected since the current Internet systems work with them due to their scalability properties.

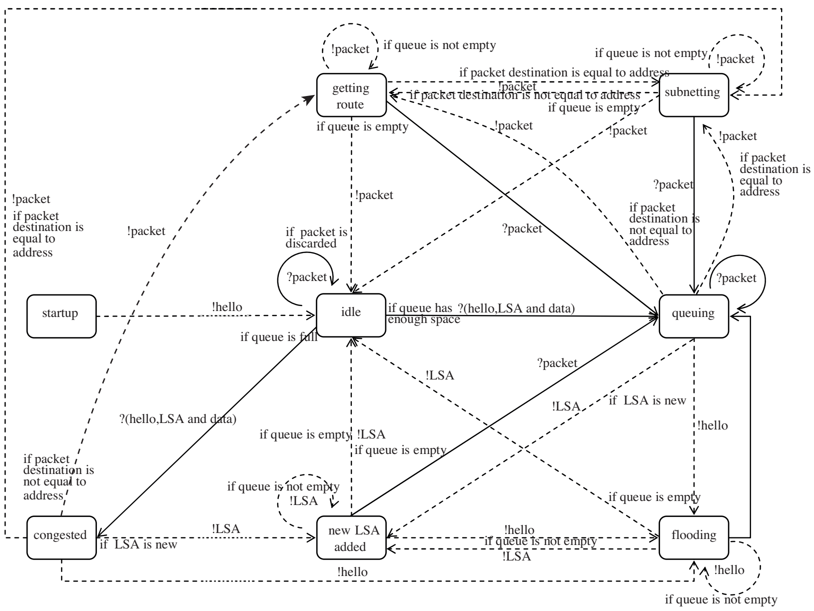


Figure 3. Finite state diagram of a node running the OSPF protocol.

2.5.1. Formalization of the OSPF-DEVS model

In this section, the OSPF-DEVS model is detailed [24]. The OSPF is a link state routing protocol and an open standard routing protocol. It runs with the Dijkstra [22] algorithm when estimating the shortest paths. The OSPF-DEVS model runs based on the following several steps. These steps are the discovery, flooding, shortest path calculation, and message forwarding phases. Theses phases are also modeled separately in the DEVS-Suite package.

1. Discovery phase: This phase is also called the hello protocol phase. The OSPF hello protocol is run by OSPF routers to get information about the neighbors and to maintain them. The routers that run the OSPF protocol are considered to be neighbors in the case of having interfaces to a common network. The hello protocol maintains the adjacencies by multicasting the hello packets. The nodes send hello packets to their neighbors at regular intervals. Next, the routers receive these hello packets sent by their neighbors. This exchange maintains the neighbor relationships on each network.

2. Flooding phase: This phase is about exchanging the link state advertisement packets between the routers. When receiving hello messages according to the hello protocol in the discovery phase, a router updates its neighbor table and sends the link state advertisement packets to all of the neighbors. Routers have a mechanism and logic for flooding link state advertisements. These mechanisms are modeled in the atomic node models. If a topological change occurs in the network (due to changes in the link status or routes), an immediate update is sent from that neighbor, alerting other OSPF-speaking routers to the change.

3. The database of the AS topology in the developed OSPF model is represented by both the neighbor table and the topology database (see Figure 1).

4. Shortest path calculation phase: The most important phase and process are to calculate the routes from the protocol stacks, such as the neighbor table and topology database. The Dijkstra [22] algorithm is used to overcome this calculation. According to the Dijkstra logic, a router calculates the shortest-path tree, considering itself on top of the hierarchy.

5. Message forwarding phase: As already mentioned, packets that are exchanged among the components in the form of DEVS messages can be distinguished as data and control packets. Data packets are basic IP packets that carry information such as ID and precedence. Control packets allow the node to obtain a whole network view and to measure the traffic.

2.5.2. Formalization of the RIP-DEVS model

Aside from the OSPF model definition, the RIP protocol is also analyzed and modeled using DEVS and implemented in the DEVS-Suite simulation viewer. First of all, as in the OSPF case above, the whole RIP logic is split into the subfunctions and procedures. The following are phases through which the complete RIP behavior can be implemented:

- Route request phase: In this phase, routers periodically advertise their distance to the destination to their neighbors, similar to the hello protocol in the OSPF case. After the initialization process, a typical RIP router uses a request message to get the neighbor's routing information. On receiving a request, the RIP router floods response messages periodically. In Figure 4a, the route request messages that are initiated in the RIP-DEVS simulator between the nodes are shown.

- Routing table update phase: Every RIP router maintains a routing table to route the data packets. In our routing table implementation, there is only one routing entry for each destination. This is the information to the destination and the cost value. On receiving an advertisement message, the router estimates whether the received routes can be used or not. If the received routes contain new information or routes, then the router updates its routing table. In Figure 4b, a typical routing table is shown where the route information is listed, including the BGP routes.

- Flooding phase: Whenever a router updates its routing table, it goes through the advertising neighbor with this newly obtained information. In Figure 4c, the flooding phase is shown in the developed simulator. RIP response messages contain the whole routing table and are exchanged among the routers.

- Message forwarding phase: When a new route is established, a router can route or forward a message containing this new information to its neighbors (see Figure 4d).

2.5.3. BGP-DEVS protocol abstraction

In our modeling study, very simple BGP protocol implementation is applied in order to simplify the model into a manageable size. Many aspects are abstracted and simplified (see Table 1). Instead of the full functionality of the BGP protocol, important properties of the dynamic behavior are considered. Our BGP-DEVS model includes receiving messages, forwarding LSAs, and applying protocol rules. In our model, a BGP message contains only the information about the sender and its associated AS. The information about the AS is obtained from the sender's IP address, since the hierarchy is implemented in the level of addressing.

Table 1. Protocol model properties.

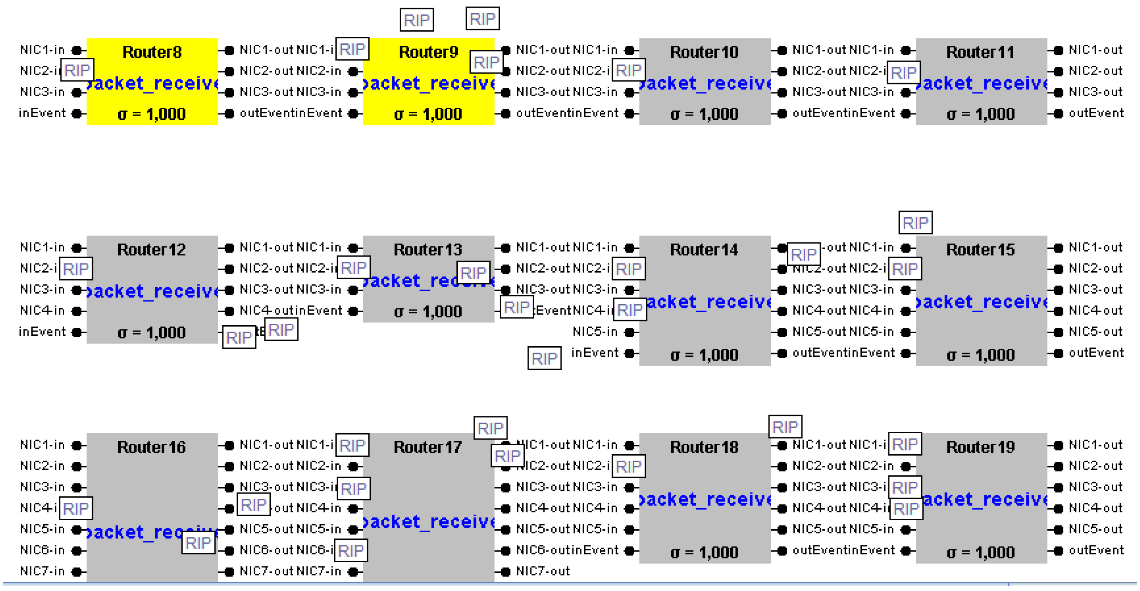
Modeled protocol properties										
Protocol	Algorithm	Protocol stacks and extended data structures	Control packets	Update of the routing information	Structure of the routing mechanisms	Hierarchy implement ation	Level of hierarchy	Metric	Routing method	Loop free routing
OSPF	Dijkstra	neighbor_table Vector<Route>	Hello	Event-driven	Hierarchical	IP address level	2 (router +AS)	Hop count	Flooding	Yes
		topology_database Vector<NETPacket>	LSA							
		RoutingTable Vector<Route>								
		LSA_history String[]								
RIP	Bellman-Ford	routingTable Vector<Route>	Hello	Periodically						
		RIP_history String[]	RIP message							
BGP	Bellman-Ford	BGP_table Hashtable<Integer, IPAddress>	BGP message	Event-driven						

Modeling ASs is one of the most important parts in a BGP protocol implementation. DEVS coupled model specification [15] is used to represent the ASs, which means that each AS is modeled as a simple digraph model, without taking its complexity and geographical span into account in the developed BGP-DEVS model. By coupling the ASs, the modeling of the intraAS network topology is performed. Consequently, each AS is modeled as a separate and independent model that is capable of communicating with other ASs by exchanging BGP messages and making decisions according to its logic for large-scale network modeling. Closure under coupling property of the DEVS [15] enables us to consider each AS as one router and model, and as such, to facilitate the building and managing of large models in a hierarchical manner.

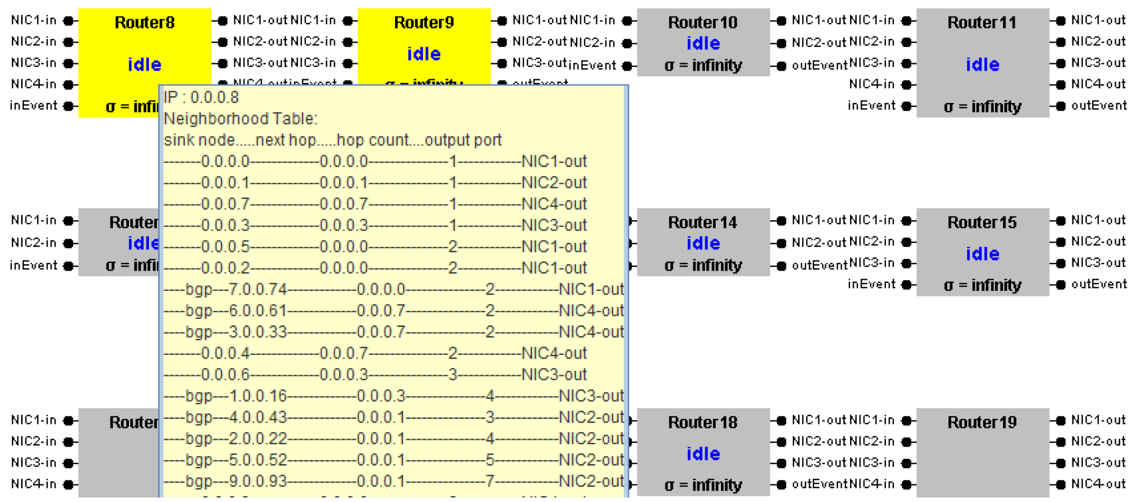
2.6. Traffic model and experimental frame

DEVS implementation has 2 options to model user traffic in the network: 1) distributed and 2) central traffic models. As a result of having a compact, easy to maintain, and deployable traffic model, an independent experimental frame model is developed and implemented (see Figure 5). It is not integrated into the network model, but it can be connected to any simulation model via couplings. The relation between the network and the experimental frame models has a resemblance to the relation between an oscilloscope and an electronic device or component to be measured.

Therefore, the management of the trace and track process in a simulation study is done formally. Although the experimental frame is a separate central authority, it does not cause a bottleneck for the simulation experiment since it does not take part in the simulation. In other words, there is no event generation, the experimental frame is a passive entity, thus, and its overhead during the experiments is negligible. Modeling some environmental events, such as the node congestion and link downs, effectively controls the execution of the whole model, and interpreting the simulation outcomes is the main function of our experimental model. In our implementation, a typical experimental frame consists of an event generator and an event transducer (see Figure 5). The DEVS experimental frame concept makes simulation management easier and takes the management tasks away from the compute nodes. The main responsibilities of the experimental frames are listed as follows:

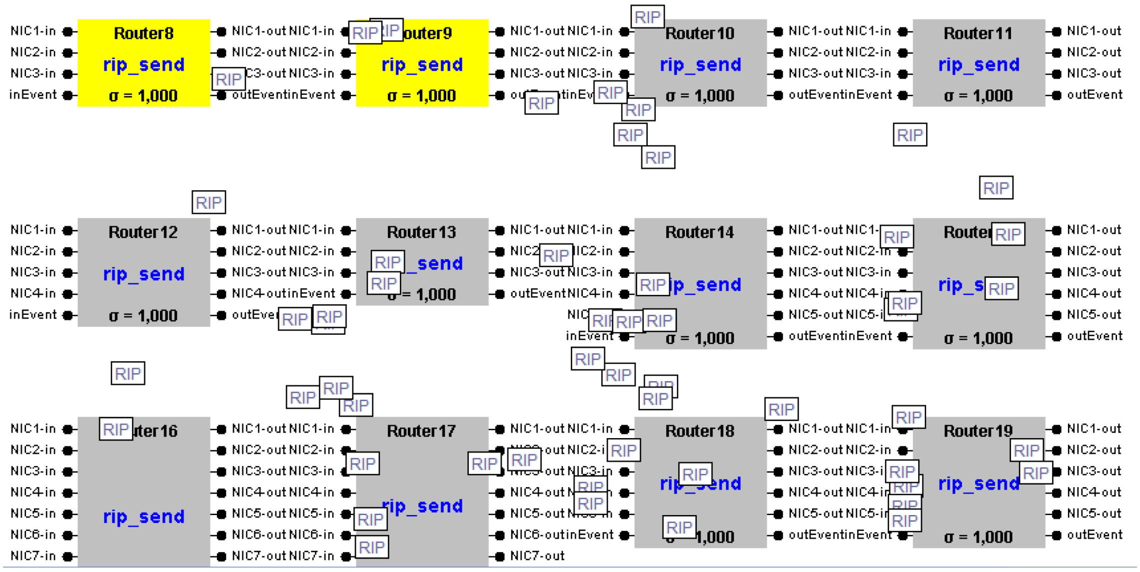


(a) Route request phase.

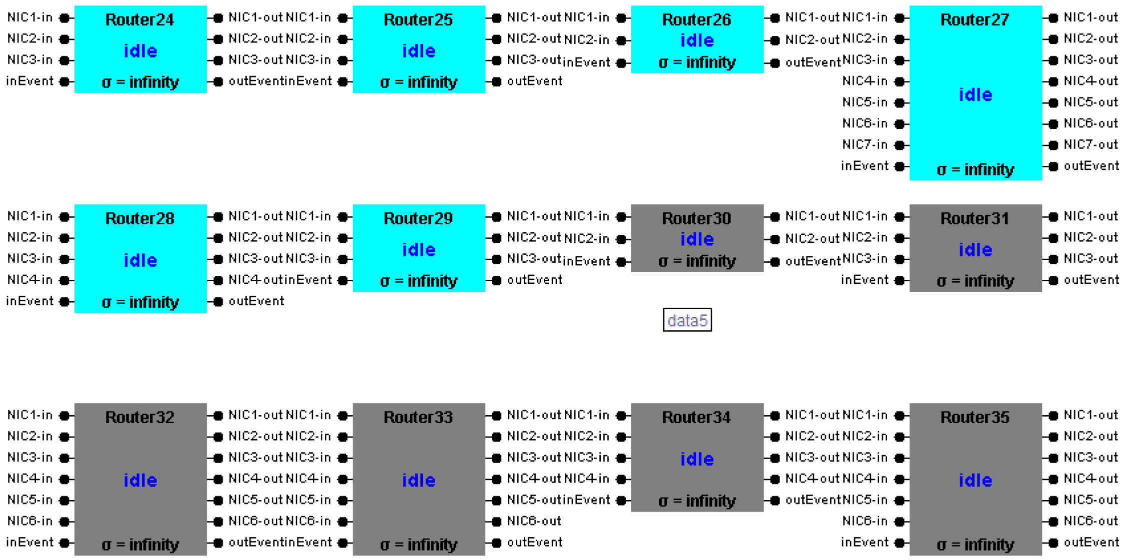


(b) Routing table update phase.

Figure 4. RIP logic phases on the DEVS-Suite RIP model visualization.



(c) Flooding phase.



(d) Message forwarding phase.

Figure 4. Continued.

- Triggering the events: In DEVS logic, simulation is already a sequence of the events. However, in some cases, there may be a need for special events, such as node and link downs, to show the performance of the protocol against a highly changing topology. In our implementation, an experimental frame can trigger every kind of event, including error models. The user traffic is simulated by generating the packet events. The generator can generate packets with fixed time intervals by randomly choosing the source and destination addresses (see Figure 5).
- Monitoring the simulation: Some statistical data can be obtained from the simulation execution (e.g., process load, memory usage, or the wallclock execution time of the simulation runs). This information can be periodically kept in an experimental frame using Java technology.

- Process results: The transducer observes and analyzes the network outputs and stores these results in trace files. The transducer simply converts the data into information that is meaningful for the modeler. Together with the DEVS-Suite tracking capability, an exact trace is applied to the developed model.

- Starting and finishing the simulation: Simulation is a process having start and end points. Simulation experiments are started and terminated by messages from generators. The first data packet starts the experiment, while last one terminates it. On termination, the transducer reports the results in human-readable form.

2.7. Topology generator

Network generators are essential as they can generate the Internet's large-scale topology for the development of efficient routing protocols. Large-scale models of more than 1000 nodes cannot be built manually. Though it is difficult to shape the Internet's router and an AS level topology, there are several network topology generators [26]. From the network topology generators, the Boston University Representative Internet Topology Generator (BRITE) [27] is an open-source, now unsupported, both C++ and Java-based, and object-oriented network generator that allows modelers to import from and export to specific topology files for those simulators such as ns-2, OMNeT++, JavaSim, and SSFNet. It is extended to support the DEVS-Suite and the integrated BRITE provides for the creation of multiple generation models, including flat AS, flat router, and hierarchical topologies.

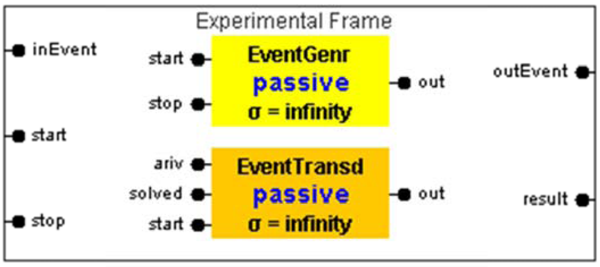


Figure 5. An experimental frame with the generator and transducer models.

3. Simulation experiments

In this paper, a network simulator on the DEVS-Suite simulation viewer has been developed and used as a protocol evaluation framework. DEVS-Suite is a simulator built on top of the DEVS, and it simulates the system's behavior by modeling each event in the system and then processes it through user-defined processes [17]. In order to investigate the performance of the simulator in large-scale models and the Internet, a series of simulation experiments are done using the RIP and OSPF, as well as the BGP models as described in previous sections. The experiments are conducted in a Core 2 Duo machine running at 2.1 GHz with a 4 GB RAM and an Ubuntu 9.10 64-bit operating system. Large-scale DEVS coupled models of up to 1000 nodes are generated using an integrated BRITE topology generator and are then measured, and the simulator outcomes are reported. In the following table and graphs, these results are given in detail to show the developed models and the simulator's performance. Over 10,000 nodes, the Java virtual machine reports an out of memory error for the OSPF-DEVS simulator, while over approximately 9000 nodes, the RIP-DEVS model reports the same message. Consequently, experiments were done within these scales.

Table 2. Selected simulation parameters and the Internet.

Parameters	Simulation	Internet (real-world)
Exterior routing protocol	BGP	BGP
Interior routing protocol	OSPF & RIP	OSPF & RIP
IP address model	IPv4	IPv4 & IPv6
IP address size	4 bytes	4 bytes
Message type	Modified IP Packetpacket	IP packet
MTU	552 bytes	576 (fragmented)–65.548 byte (unfragmented)
Header size	20 bytes	20 bytes
LSA max age	Infinity	1 h
Number of hops	15	15
Node processing time	1	1 Mb/s–40 Gb/s
Queue length	200 KB (362 packets)	32–500 packets
Link bandwidth	25 Mb/s	25 Mb/s–10 Gb/s
Link delay	3 ms	3–16 ms
Layer	2	2 (Internet core and ASs)
Scale	10,000 nodes/hosts	Hundreds of millions of hosts and users
Model	Waxman	N/A
Observation time	1000	N/A
Interarrival time	1	N/A

In Table 2, the simulation and real-world parameters are listed to show the parameter's verification of the developed models. The parameters are selected as almost the same as the real-world ones (e.g., the packet sizes are selected as 552 bytes, whereas on the Internet, the size is approximately 576 bytes).

4. Results and analysis

The RIP and OSPF models are compared to reveal their scalability characteristics on increasing size and complexity. Figure 6 depicts the wallclock execution times of the modeled protocols across a network size. With network scales of up to 10,000 nodes, the execution times increase exponentially for both protocols. Though the RIP has lower execution times for small networks (up to 3500 nodes), for very large networks the OSPF shows its strength and consistency against the RIP (for example, for a 7000-node network, the OSPF yields 340 s, while the RIP yields a 580 s wallclock execution time); the results show that the OSPF has better scalability when compared to the RIP.

Table 3 shows the OSPF and RIP performance results for all of the synthetic topology scenarios across the varying topology parameters. Models composed of a varying number of nodes and ASs are generated using the BRITE topology generator. All of the networks are modeled similar to the Waxman topology model [28] and the connectivity parameters (links per nodes - m) are set to 2, but are selected as 1 in some models, as seen in Table 3.

First, the convergence values of the models are compared. The convergence time is a simulation setup time that all of the routers install in their routing databases. Low convergence means a speedy routing process and it is desired in large-scale cases. For all of the models, the convergence time varies linearly with the number of nodes. It also has a linear relation with the connectivity (m value). As shown in Table 3, the RIP has a significantly better convergence time than the OSPF. In particular, for larger models, the RIP is approximately 3 times faster than the OSPF (e.g., for 5000 nodes, the RIP convergence value is 786 simulation steps, while the OSPF yields 2445 steps).

Aside from the convergence value, the efficiency is measured for all of the network models. The efficiency can be formulated as a ratio of the packets delivered successfully. For the OSPF, it is obvious that the simulator runs with an extremely high degree of efficiency, which is estimated as the packet delivery ratio, and the lowest efficiency is 99.547%, which means that 99.547% of the total packets are delivered to their destinations safely. As opposed to the convergence, the RIP protocol shows less performance, where its efficiency is 6.5% for the 7000-node model, while for the OSPF it is 99.9%.

The throughput and turnaround time as main network performance criteria are also observed from the experiments for the OSPF and RIP models. These values allow us to evaluate the network performance. The throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second or data packets per time slot. In large-scale models, as the average bandwidth consumption increases, the throughput inherently gives a small value; for example, 2.3 Kbps for a 10,000-node network size in the OSPF. From Table 3, the throughput values are almost the same up to the 5000-node network scale. After that size, the RIP becomes worse (e.g., in the 7000-node size, the OSPF yields 3.51 Kbps and the RIP yields 0.29 Kbps).

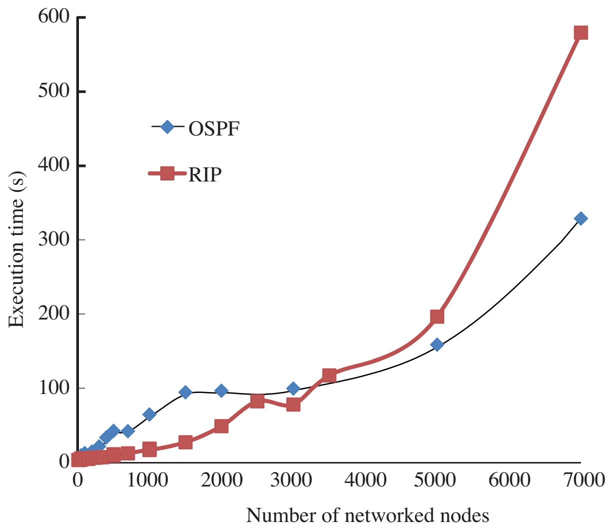


Figure 6. Wall clock execution times of the RIP and OSPF models.

The average delay was measured in seconds for large networks in all of the model cases. The average delays of both the RIP and OSPF models are almost the same for up to 1000 nodes. In larger models of more than 1000 nodes, the gap between the delay values increases as opposed to the OSPF (for example, in the 7000-node network size, the average delays were 90.47 s for the OSPF and 25.26 s for the RIP).

5. Conclusion and future work

Interior routing protocols such as the RIP and OSPF are widely used in computer network systems such as the Internet. In this paper, we have presented a comparative analysis of the selected routing protocols using a high performance modeling strategy DEVS and its associated technologies. Comparative analysis has been done in the same network configurations with different protocols for real-time applications. Performance has been measured on the basis of some parameters that aimed to figure out the effects of the scalability on the routing protocols. Network scalability can be enhanced by reducing the network convergence time and decreasing the overhead of the routers. In our paper, the implementation of the RIP shows that the network convergence time is much faster than that of the OSPF networks because the RIP network learns the topology information and

Table 3. Comparison of the OSPF and RIP protocol scalability characteristics.

Number of nodes	Number of AS	Number of nodes per AS	Topology model	Logical convergence time(steps)		Efficiency (%)		Average throughput (Kbps)		Average turnaround time (sec)		Simulation logical time (steps)	
				OSPF	RIP	OSPF	RIP	OSPF	RIP	OSPF	RIP	OSPF	RIP
10	1	10	Waxman (m = 2)	46	50	100.00	100	4.40	4.39	5.076	5.213	1051	1007
20	1	20	Waxman (m = 2)	109	98	100.00	100	4.39	4.36	6.317	7.346	1116	1012
30	1	30	Waxman (m = 2)	191	128	100.00	100	4.39	4.38	6.818	7.652	1199	1008
50	1	50	Waxman (m = 2)	392	183	99.97	93.5	4.28	4.08	7.64	9.024	1401	1012
50	5	10	Waxman (m = 2)	104	131	100.00	100	4.36	4.35	10.231	10.16	1118	1015
70	1	70	Waxman (m = 2)	647	213	99.91	98.4	4.01	4.30	8.23	10.5	1599	1010
100	1	100	Waxman (m = 2)	747	228	99.92	96.7	4.05	4.20	8.62	11.35	1755	1016
100	5	20	Waxman (m = 2)	207	130	99.99	92.9	4.34	4.03	12.88	13.11	1225	1017
100	10	10	Waxman (m = 2)	132	131	100.00	100	4.36	4.34	12.347	13.29	1146	1018
150	5	30	Waxman (m = 2)	355	167	99.89	99.1	3.90	4.28	13.69	17.15	1374	1023
200	10	20	Waxman (m = 2)	259	171	99.99	97.7	4.30	4.19	15.17	17.14	1281	1030
250	5	50	Waxman (m = 2)	517	235	99.96	92.2	4.19	3.94	15.19	18.41	1535	1033
300	10	30	Waxman (m = 2)	450	244	99.94	93.9	4.05	4.05	18.2	19.56	1484	1024
350	5	70	Waxman (m = 2)	739	268	99.96	97.6	4.13	4.17	18.99	20.67	1774	1033
500	5	100	Waxman (m = 2)	1304	477	99.82	94.5	3.57	4.06	18.63	22.7	2330	1028
500	10	50	Waxman (m = 2)	706	285	99.85	91	3.65	3.88	21.9	22.84	1735	1036
500	50	10	Waxman (m = 2)	654	273	100.00	97.3	4.27	4.19	21.091	21.95	1688	1026
700	10	70	Waxman (m = 2)	1061	320	99.88	93.9	3.82	4.03	23.86	24.12	2103	1028
1000	10	100	Waxman (m = 2)	1385	469	99.85	87	3.65	3.69	24.11	25.52	2419	1041
1000	50	20	Waxman (m = 2)	935	412	99.96	92.2	4.13	3.94	26.17	24.87	1971	1034
1000	100	10	Waxman (m = 1)	246	134	100.00	29.6	4.11	1.27	45.243	27.59	1321	1027
1000	100	10	Waxman (m = 2)	1154	321	99.99	96.9	4.24	4.15	23.95	23.82	2186	1031
1500	50	30	Waxman (m = 2)	1092	505	99.87	75.9	3.71	3.23	31.78	28.96	2140	1037
2000	100	20	Waxman (m = 2)	1638	810	99.94	83.8	4.00	3.58	30.095	27.92	2680	1033
2500	50	50	Waxman (m = 2)	238	780	99.81	69.6	3.44	2.96	39.46	31.84	1289	1038
3000	100	30	Waxman (m = 2)	2184	901	99.86	69.2	3.63	2.96	34.807	30.07	3230	1034
3500	50	70	Waxman (m = 2)	245	743	99.86	45.9	3.67	2.28	35.24	31.04	2707	1047
5000	50	100	Waxman (m = 2)	2496	1024	99.67	51	2.86	2.08	37.06	32.67	3544	1041
5000	100	50	Waxman (m = 2)	2445	786	100.00	50.5	3.74	2.12	95.431	32.1	1523	1053
7000	100	70	Waxman (m = 1)	407	281	99.90	6.5	3.51	0.29	90.47	25.26	1542	1001
8000	100	80	Waxman (m = 2)	-	944	-	68.8	-	1.34	-	32.88	-	1030
9000	100	90	Waxman (m = 2)	-	1060	-	63.2	-	1.57	-	33.11	-	1035
10,000	100	100	Waxman (m = 2)	3321	-	99.54	-	2.3	-	39.906	-	4372	-

updates faster than the OSPF. However, the RIP's efficiency is significantly low in larger networks. In the context of efficiency, we have found that the efficiency in the RIP network is less than that of the OSPF. In comparison, the simulation results have shown that the throughput in the OSPF network is higher than that of the RIP network.

The simulation results have shown that the end-to-end delay of the RIP network is relatively less than that of the OSPF network. As a result, data packets in the RIP network reach their destination faster. Another performance metric for routing protocol evaluation is the packet delay variation, which measures the differences between the delays of the packets. The performance of the packet delay variation for the RIP is better than that of the OSPF for large-scale networks. In small networks, these values are almost the same.

Consequently, in this work, the comparative performance among the RIP and OSPF routing protocols for large-scale applications has been analyzed. By comparing these protocols' performances, we have discovered that the implementation of the OSPF routing protocol in the network performs better than the RIP (i.e. the OSPF can scale up to 10,000 nodes, while the RIP is about 9000). In future, a research work can be done on the explicit features of both the OSPF and RIP in a parallel and distributed environment.

In the performed tests, it has been seen that on small-scale networks, the RIP is more productive, while on large-scale and complex networks, the OSPF protocol is more productive. As seen in Table 3, the productivity of the RIP protocol with respect to the OSPF has been decreased on a medium-scale (greater than 1000 routers) network. The performance of the RIP on large-scale networks (3500 and more routers) is seen from the running time graphic in Figure 6. These results have shown that because of the fact that the required productivity on extra-large scales cannot be taken from the RIP protocol, it is necessary to use OSPF protocols.

Acknowledgments

This work has been funded by Sakarya University Scientific Research Projects Agency under the contract 2011-50-02-022. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Sakarya University.

References

[1] H. Mahini, R. Berangi, A. Mahini, "MLET: a power efficient approach for TCAM based, IP lookup engines in Internet routers", International Journal of Computer Networks and Communications, Vol. 2, pp. 13–26, 2010.

[2] M. Steenstrup, Routing in Communications Network, New Jersey Prentice-Hall, 1995.

[3] A.S. Tanenbaum, Computer Networks, 4th edition, New Jersey, Prentice Hall, 2003.

[4] C. Hendrik, "Routing information protocol", http://tools.ietf.org/html/rfc1058, accessed 30 May, 2012.

[5] J. Moy, "OSPF version 2", http://www.ietf.org/rfc/rfc2328.txt, accessed 30 May, 2012.

[6] L. Chen, M.A. Babar, "A survey of scalability aspects of variability modeling approaches", Workshop on Scalable Modeling Techniques for Software Product Lines at the 13th International Software Product Line Conference, 2009.

[7] M.A. Yehia, M.S. Aziz, H.A. Elsayed, "Analysis of IGP routing protocols for real time applications: a comparative study", International Journal of Computer Applications, Vol. 26, pp. 11–17, 2011.

[8] I. Kaur, M. Sharma, "Performance evaluation of hybrid network using EIGRP & OSPF for different applications", International Journal of Engineering Science and Technology, Vol. 3, pp. 3950–3960, 2011.

[9] S.G. Thorenoor, "Dynamic routing protocol implementation decision between EIGRP, OSPF and RIP based on technical background using OPNET modeler", 2nd International Conference on Computer and Network Technology, Vol. 1, pp. 191–195, 2010.

[10] R. Randhawa, J.S. Sohal, "Comparison and performance of routing protocols in SONET based networks", Optik - International Journal for Light and Electron Optics, Vol. 121, pp. 997–1002, 2010.

[11] C.C. Chiang, C. Chen, D. Jeng, S. Wang, Y. Ho, "The performance and security evaluations of internet routing protocols", Journal of Informatics and Electronics, Vol. 2, pp. 21–27, 2008.

[12] H. Pun, "Convergence behavior of RIP and OSPF network protocols", Master's thesis, University of British Columbia, 1998.

[13] D. Pei, D. Massey, L. Zhang. "A formal specification for RIP protocol", Technical Report TR040046, UCLA CSD, 2004.

[14] A. Zengin, "Modeling discrete event scalable network systems", Information Sciences, Vol. 181, pp. 1028–1043, 2011.

[15] B.P. Zeigler, H. Praehofer, T.G. Kim, Theory of Modeling and Simulation, New York, Academic Press, 2000.

[16] A. Zengin, "Large-scale integrated network system simulation with DEVS-Suite", KSII Transactions on Internet and Information Systems, Vol. 4, pp. 452–474, 2010.

[17] S. Kim, H. Sarjoughian, V. Elamvazhuthi, "DEVS-Suite: a simulator supporting visual experimentation design and behavior monitoring", Proceedings of the Spring Simulation Multiconference, pp. 29–36, March 2009.

[18] B. Zeigler, Object Oriented Simulation with Hierarchical Modular Models, Massachusetts, Academic Press, 1990.

[19] A. Chow, "Parallel DEVS: a parallel, hierarchical, modular modeling formalism and its distributed simulator", Transactions of the Society for Computer Simulation International, Vol. 13, pp. 55–67, 1996.

[20] H.S. Sarjoughian, B.P. Zeigler, "DEVS and HLA: complementary paradigms for modeling and simulation?", Transactions of the Society for Modeling and Simulation International - Ethical Issues in Modeling and Simulation, Vol. 17, pp. 187–197, 2000.

[21] H.S. Sarjoughian, "DEVS-Suite Simulator", http://devs-suitesim.sourceforge.net/, accessed 30 May, 2012.

[22] E. Dijkstra, "A note on two problems in connexion with graphs", Numerische Mathematik, Vol. 1, pp. 269–271, 1959.

[23] R. Bellman, "On a routing problem", Quarterly of Applied Mathematics, Vol. 16, pp. 87–90, 1958.

[24] A. Zengin, H.S. Sarjoughian, "OSPF discrete event simulator for computer network education and research", Submitted to ACM Tomacs Journal, 2012.

[25] A. Zengin, H.S. Sarjoughian, H. Ekiz, "Discrete event modeling of swarm intelligence based routing in network systems", Information Sciences Journal Vol. 222, pp. 81–98, 2013.

[26] K. Wehrle, M. Güneş, J. Gross, Modeling and Tools for Network Simulation, Heidelberg, Springer, pp. 35–59, 2010.

[27] A. Medina, A. Lakhina, I. Matta, J. Byers, "BRITE: an approach to universal topology generation", International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, pp. 346–353, 2001.

[28] B.M. Waxman, "Routing of multipoint connections", IEEE Journal on Selected Areas in Communications, Vol. 6, pp. 1617–1622, 2002.