

A Real-Time Testbed for Routing Network

Kang Yao¹, Weiqing Sun², Mansoor Alam¹, Mingzhe Xu² and Vijay Devabhaktuni¹

¹Department of EECS, The University of Toledo, Ohio, USA
{kang.yao, mansoor.alam2, vijay.devabhaktuni}@utoledo.edu

²Department of ET, The University of Toledo, Ohio, USA
{weiqing.sun, mingzhe.xu}@utoledo.edu

Abstract. Existing network testbeds can enable developers to evaluate the performance of different routing protocols in a network and help students to enhance their hands-on experiences and understand complex and abstract concepts of routing protocols by allowing them carrying out projects on it, but they are either limited in features or expensive to establish and manage. To address this problem, this paper presents ARTNet - A Real-Time Testbed for Routing Network – which supports almost all the popular routing protocols for typical applications in a cost-effective manner. ARTNet has been implemented on a multiprocessor server for users to create and manage their routing networks. Performance and functionality evaluations on the ARTNet platform show that it is a promising approach.

Keywords: ARTNet, Network Testbed, Routing Protocols, Virtualization, Network Service.

1 Introduction

Network researchers and designers need to test different routing protocols under various conditions to determine whether they are robust or reliable. Students need to learn how to deploy routing protocols and manipulate their parameters in order to understand how these protocols actually work. However, it is not always possible to setup an entire live routing network testbed due to its high cost.

Two types of techniques, emulation and simulation, have addressed the problem. The main difference between the two techniques is that a simulator is a program focusing on modeling components of a network, while an emulator is the hardware, software, or both focusing on duplicating the behavior of a live network. For emulation, a computer system running software such as Zebra [1], Quagga [2], and XORP [3] functions as a part of emulated testbed. For simulation, only software is used to simulate a whole network. There are a few well-known simulators in the market, such as OMNeT++ [4], Ns-2 [5], Cisco Packet Tracer [6], and OPNET IT Guru [7].

An ideal testbed for routing protocols should comply with the following basic requirements:

1. It should provide a real-time, cost-effective, and realistic environment. Although the approach of emulation provides a real-time environment, it is expensive in terms of hardware cost. The approach of simulation, on the other hand, does not provide the real-time ability and the necessary realism.
2. It should enable users to study and evaluate the performance of the network under specified loads. The load could consist of HTTP, FTP, TFTP, SMTP, POP3, DNS, and so on.
3. It should be scalable and extensible. The testbed should scale well with increasing number of routers, links and be able to inject real-world application traffic to the routing network. The approach of emulation allows users to inject real-world traffic, which is extensible. However, the scalability is constrained by its limited resources, management, and physical setup. Although the approach of simulation is used to perform large-scale routing experiments, it is hard to add non-built-in traffic.
4. It should support routing protocols used widely in campus or industry networks, including Intermediate System to Intermediate System (IS-IS), OSPF, Routing Information Protocol (RIP), EIGRP, Border Gateway Protocol (BGP), and so on. Among these, different Interior Gateway Protocols (IGP), OSPF and EIGRP have been considered as the pre-eminent routing protocols. However, because EIGRP is a Cisco proprietary protocol [8], most emulators or simulators do not support it.
5. It should provide a user-friendly environment for conducting routing experiments and performing analyses.

In this paper, we design a testbed ARTNet in an effort to meet all the requirements by providing all the above features. The rest of the paper is organized as follows: Section 2 presents the architecture of ARTNet. Section 3 describes the implementation and evaluation of ARTNet. Section 4 presents related work. Finally, in Section 5, conclusions are drawn, and future work is proposed.

2 ARTNet Architecture

Based on the requirements, we design the ARTNet architecture as shown in Fig. 1. ARTNet consists of a Web-based Interface (WBI) for remote access and control, Initialization Module (IM), Real-time Configuration Module (RCM), ARTNet Database, Virtual Network (VN), Virtual Hosts (VHs), and Results Visualization Module (RVM).

The WBI is a part of the GUI developed for users to login and access the virtual networks. It also enables users to specify configuration options for a network including the topology, protocols along with their parameters, and various loads on VHs. Once IM receives the users' input through WBI, it is responsible for initializing the VN and VHs of their networks. Thus, these two components provide the configuration initialization capability of ARTNet. WBI, IM, and RCM can read and write data into the ARTNet Database, which contains all the data related to the routing networks in the testbed. Users can also interact with ARTNet and their networks during the process of the experimentation, since RCM provides the real-time

control to the VN and VHs. The RVM component provides the functionality of capturing and analyzing the experimentation data. It is also equipped with a user-friendly interface for displaying the network statistics, deriving performance metrics, and makes it easier for users to analyze results and develop insights from the results.

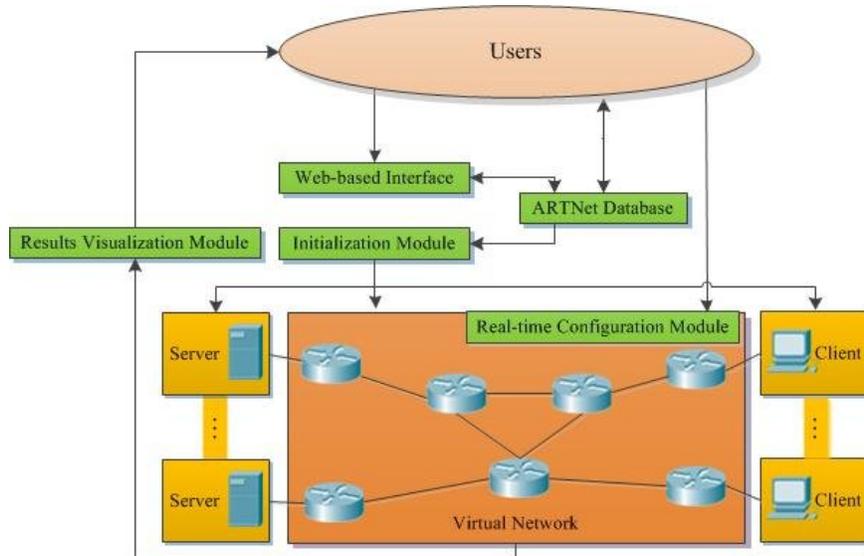


Fig. 1. Architecture of ARTNet.

2.1 Virtual Network

The VN consists of virtual routers, switches, and links. Our infrastructure is built on top of the real-time network simulator GNS3 [9]. It is a free and open-source Graphical Network Simulator that allows emulations of complex networks using Cisco Internetwork Operating Systems (Cisco IOSs) running on Windows, Linux and Mac OS X based computers. GNS3 was chosen due to the following features:

- Users can interact with it during experimentations. If an unusual event occurs in a network, they can see the impact immediately and make prompt adjustments. It provides the basic real-time control ability needed by ARTNet.
- Cisco IOS software supports most of the routing protocols including ODR, RIP, IGRP, OSPF, EIGRP, IS-IS, and BGP [10]. These routing protocols will make a good set supported by ARTNet.
- GNS3 supports real-time interactions with the simulated network. It also allows a connection of the simulated network to real routers or other computers. These features play a key role in the extensibility of ARTNet.
- It supports a realistic environment by using real Cisco IOSs and CLI, which is a powerful method for expert administrators to configure Cisco's devices.

There are two sides to every sword. Due to licensing restrictions, users have to buy Cisco IOSs from Cisco or its partners [11]. In addition, The IOSs can not to be modified. In other words, ARTNet is not programmable and new routing protocols will be supported only Cisco IOSs update to have this new feature.

Two sets of configuration files in GNS3 are supported and manipulated in ARTNet, one for routers' configurations, and the second file is used to store GNS3 topology information such as name, location, and the IOS for each virtual router. When users start a virtual network, GNS3 will load these files for instantiating the topology and routers.

2.2 Virtual Host

A VH is a completely isolated duplicate of a real computer. In ARTNet, it provides the ability of generating real-world application traffic. VHs are connected to the virtual network through the Cloud device in GNS3. Users can manage how VHs connect to VN using WBI in ARTNet. Application servers are installed and run in VHs to provide the capability of traffic generation based on users' specifications. Hence, the network performance can be analyzed by measuring specific traffic on a particular network.

VMware Workstation is chosen to provide virtual hosts because it provides seamless support for different guest OSs, manageability, and operational flexibility. VMware Workstation allows one physical PC to run multiple operating systems at the same time without rebooting or hard-drive partitioning. Physical hardware resources are mapped to the virtual machine's resources, so that each virtual machine has its own CPU, memory, disks, I/O devices and more [12].

There are some common application services used in today's networks, including HTTP, FTP, TFTP, Email, and DNS services. Table 1 lists the software that is commonly used to provide these services. User can also choose to incorporate other applications that produce various types of traffic into the network. Examples include Voice over IP (VoIP), and streaming video.

Table 1. Application Servers

Application Services	Software
HTTP	Apache HTTP Server
FTP	Vsftpd
TFTP	Tftpd
Email	Postfix
DNS	BIND

Currently, in ARTNet, Ubuntu is used for the guest operating system running on each VH. A few open-source application servers have been pre-installed in the guest OS, such as BIND and Postfix. With these services installed, VHs function as application servers. The desired application services can be easily specified using ARTNet interface without users worrying about the installation or configuration of such services.

2.3 Web-based Interface and Initialization Module

ARTNET consists of two GUI components, WBI and IM for researchers and students to create and manage their routing networks in ARTNet.

WBI provides an interface to specify configuration options. Only the authenticated users can login and manage their networks remotely. Users can configure a VN or VNs in a visual way. For example, possible specifications for a router include which routing protocol to deploy, the IP address for each interface on the router, and so on. After configurations are made, IM generates the corresponding configuration files. ARTNet will then power on the specified network automatically by using the respective configuration files. Fig. 2 shows a screenshot of the WBI for configuring a virtual EIGRP network.

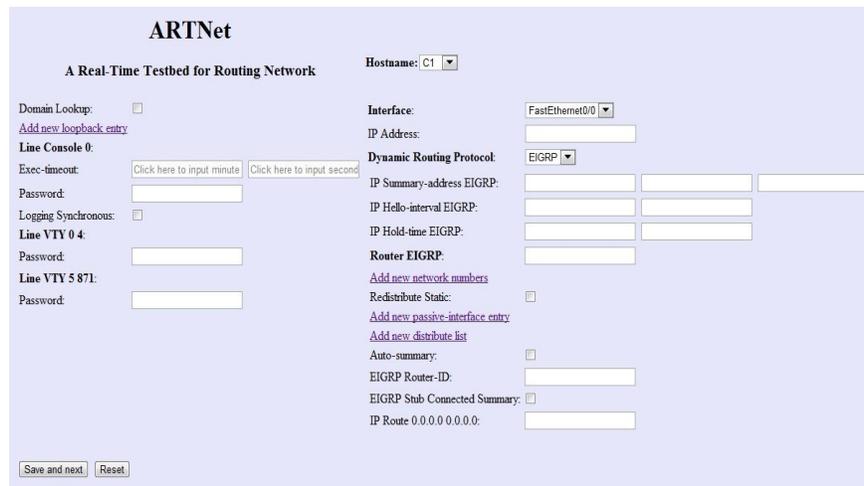


Fig. 2. Screenshot of WBI for a sample network.

For example, if a user wants to configure a virtual router C1 in an existing virtual network, he can choose C1 from the drop-down menu and then configuration changes may be made to this router. As you can see, users do not need to fill in the complete information, and default values will be used for the rest. After pressing the save button, the configuration changes will be sent to IM, which will populate the configuration file named C1.cfg for router C1.

By using the GUI interfaces, it is more efficient and easier for users to configure their networks and specify routing options when compared to text or command-line based methods. And it also helps to avoid making configuration errors.

2.4 ARTNet Database

ARTNet stores all of the information which is needed to run the routing networks in a central database. The stored data consists of three parts: User & Login (U & L) data,

General Configuration Data (GCD), and Protocol Dependent Configuration Data (PDCD). Each type of data will be handled by its corresponding module.

The U & L data consists of users' names, passwords, and activity logs. A user can access ARTNet from any computer connected to the Internet using WBI. User name and password are checked against the authentication information stored in the database. The administrator can pre-register the network for each user. And users may also register by themselves.

GCD and PDCD are used to store data regarding routers' configurations in a routing network. GCD stores some common configurations such as router name, static route, loopback interface or access-list. PDCD stores data about routing protocols, such as router ID, redistribution, Hello/Dead interval, and more.

WBI locates the network configuration data in ARTNet Database upon a successful user login. Then IM will extract the data and generate different configuration files to instantiate the network.

2.5 Real-time Configuration Module

Networks hosted by ARTNet can also be configured and controlled dynamically using RCM. For instance, they can dynamically adjust the routing parameters in real-time for VN and VHs, when the whole network is powered on.

RCM enables users to start, stop, restart and login to a virtual router. For instance, users are able to interrupt the primary link for a subnet during the experimentation by shutting down the network interface of the router.

RCM can also be used to control a specific VH or a group of VHs. It is capable of performing the following operations: start and stop a VH, run an executable program within a VH, copy a file from a VH to the host computer or vice versa, and assign an IP address to a VH. For example, the packet file stored in a VH can be copied from the guest VH operating system to the host operating system for further analysis.

2.6 Results Visualization Module

RVM is responsible for capturing packets, monitoring networks, filtering and visualizing packets, and displaying traffic statistics. Because the raw packet capture files usually include a large amount of data, it is important for users to be able to display and analyze only those packets of their interest. In addition, RVM provides key performance metrics such as convergence duration, throughput, end-to-end delay, delta time, number of retransmission packets, and service response time, which are used to measure the performance of routing protocols.

The RVM consists of two different components. The first part is a packet analyzer, which is used to capture all the packets over a network interface. The second is a GUI which will display the statistics about the performance metrics. As VMware Workstation or GNS3 has no built-in packet capture functionality, we use the open-source Wireshark [13] as our basic packet capturer/analyzer. After an experiment is completed, RVM will import the files captured by Wireshark, and then generate the

statistics about these metrics from the packet files. Users can then compare them across different routing protocols in a visual way with just a few clicks.

3 Implementation and Evaluation of ARTNet

This section describes the hardware and software components used to implement the ARTNet platform. In addition, performance and functionality evaluations are performed on ARTNet.

3.1 Implementation of ARTNet

ARTNet is currently hosted on a multiprocessor server with 16×Intel® Xeon® E5620 @2.40GHz processor, 48 GB memory, 500 GB hard drive, and six gigabit Ethernet interfaces.

The open-source software bundle, LAMP (Linux, Apache, MySQL, and PHP5), is used to implement the ARTNet architecture as described in Section 2. The operating system is Ubuntu 11.10 64-bit desktop version which has the built-in GUI support. The ARTNet Database is created and managed by MySQL 5.5. The Apache Web Server is used to host WBI and IM, and PHP 5.4.0 is used to implement the server side functionality as required by the WBI and IM components of the architecture. In particular, two PHP scripts are written to realize the function that users can initialize routing networks in ARTNet. The first one is used to interact with users to provide authentication and writing users' configuration options into the database. The second script is responsible for reading from the database and generating corresponding configuration files.

ARTNet uses the stable version GNS3 v0.7.4, which supports 1700, 2600, 3600, and 3700 serials routers for constructing routing networks. In our current setup, we use router model 3640 and the corresponding IOS is c3640-ik9o3s-mz.124-25b.bin.

VMware Workstation 8 is used to create VHs. The OSes for guests are the same as the OS for the workstation. Several virtual network interfaces (vmnets) are created and set up as host-only networks.

3.2 Evaluation of ARTNet

The ARTNet can run multiple instances of the routing networks for different users at the same time. This is especially useful in a class setting, when each student (or group of students) is assigned a dedicated routing network for experimentations. The number of instances supported is dependent on the size of routing network and the available resources on the physical server. The performance evaluation will shed light on this aspect. In addition, a comparative routing protocol study is performed on ARTNet to show its usability. Both the evaluations are conducted based on a campus routing network.

The campus network segregates the functions of the network into three layers: Access, Distribution, and Core Layers [14]. According to the Cisco recommended

hierarchical campus design [15], routers at different layers play distinct roles. Routers at the Access Layer are intended to forward traffic only to and from the locally connected subnets. They ensure that the packets are delivered to end-user devices. The Distribution Layer acts as an aggregation point for all Access Layer devices. The Core Layer provides the high capacity transport between the attached Distribution Layer routers. Routers at this layer implements policies, such as route filter or summarization. The Core Layer provides the high capacity transport between the attached Distribution Layer routers. Routers at this layer focus on speed and ensure reliable delivery of packets.

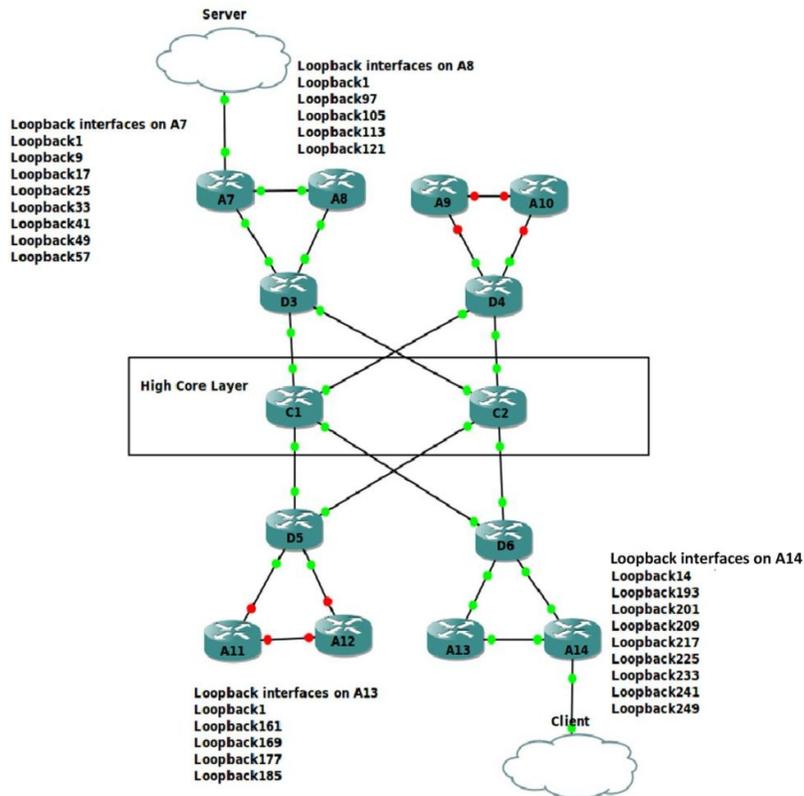


Fig. 3. A small campus network-EIGRP scenario.

3.2.1 Performance Evaluation

First, two small campus network scenarios including 14 routers and 2 Cloud devices (in the form of VHs) as shown in Fig. 3 and Fig. 4 are created in ARTNet. D3, D4, D5, and D6 belong to the Distribution Layer. C1 and C2 belong to the Core Layer. Other end-user devices in a LAN are represented by the loopback interfaces on the Access Layer routers. The performance evaluation is done by running an increasingly

number of instances of the two scenarios until the workstation runs out of CPU or RAM resources.

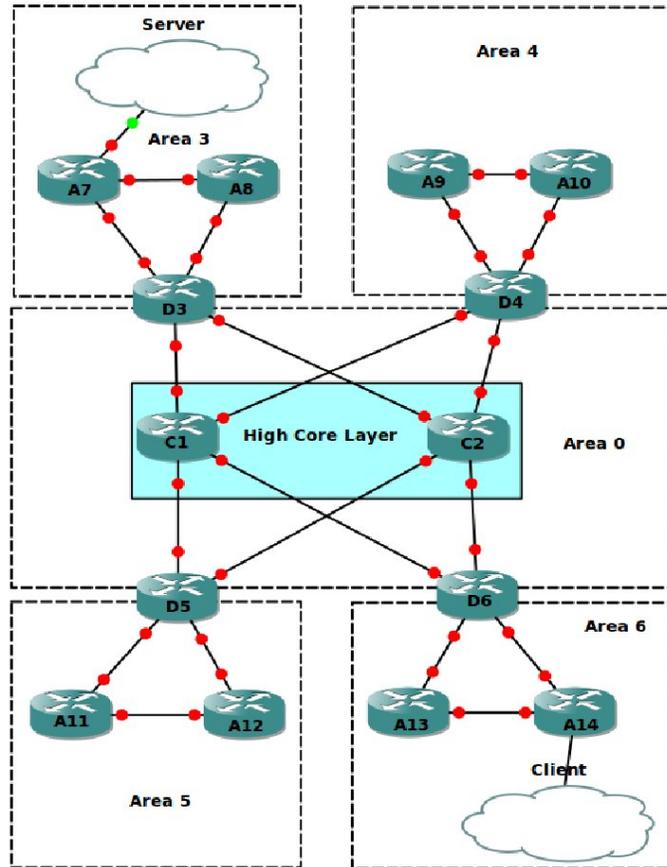


Fig. 4. A small campus network-OSPF scenario

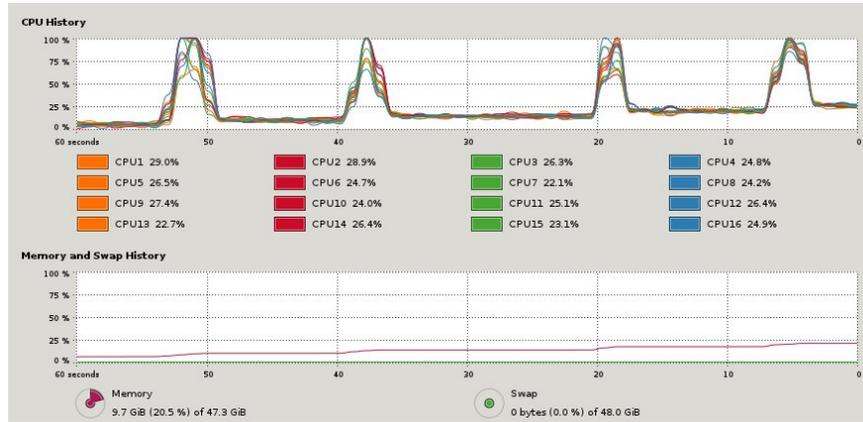


Fig. 5. CPU and RAM usage with 4 running instances . The average CPU usage is about 25.4% and the memory usage is 20.5%.

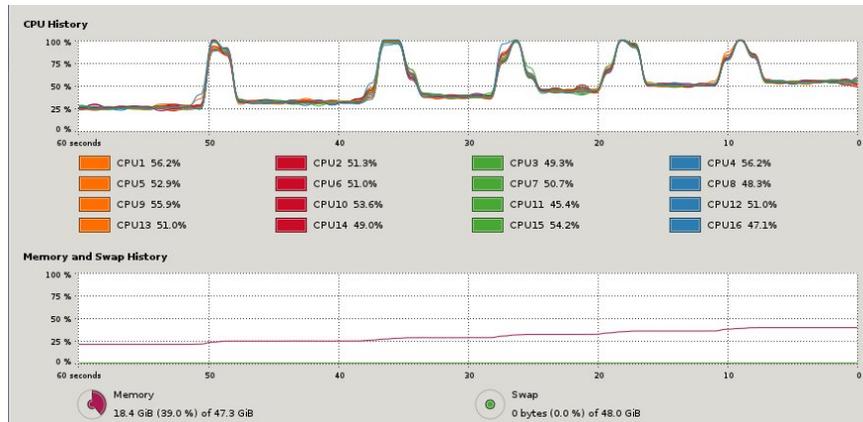


Fig. 6. CPU and RAM usage with 9 running instances. The average CPU usage is about 51.4% and the memory usage is 39.0%.

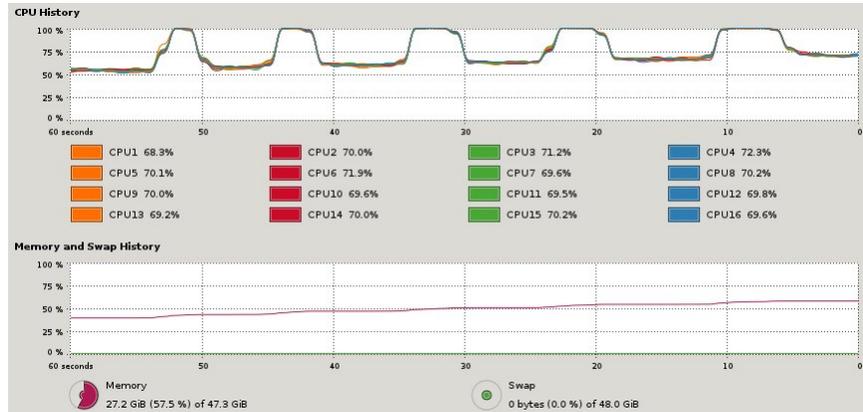


Fig. 7. CPU and RAM usage with 14 running instances. The average CPU usage is about 70.4% and the memory usage is 57.5%.

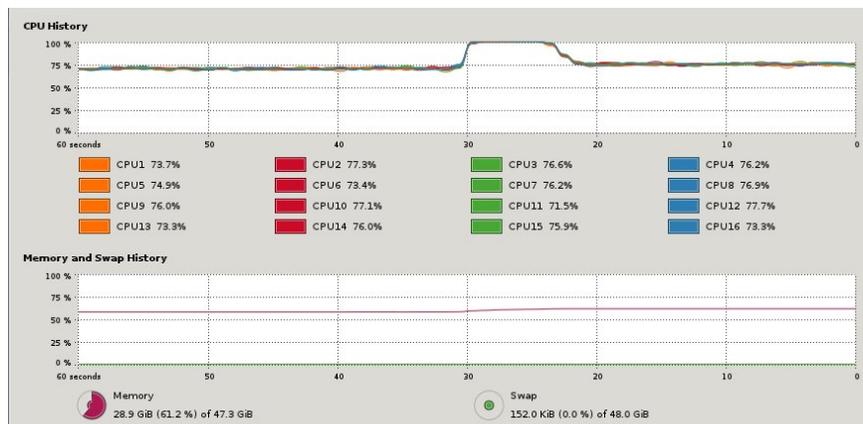


Fig. 8. CPU and RAM usage with 15 running instances. The average CPU usage is about 75.4% and the memory usage is 61.2%.

As shown in Fig. 5, Fig. 6, Fig. 7, and Fig. 8, the CPU runs at 100% temporarily when a new routing network instance starts. Then the CPU usage drops after 3-7 seconds. This happens after the idlepc values have been set. It sleeps the virtual router occasionally while virtual routers are idle. This significantly reduces the CPU consumption on the server. 25% of the CPU resources are reserved for the virtual routers and VNs to perform their tasks. For our specific setup, ARTNet supports about 15 routing network instances concurrently.

3.2.2 Functionality Evaluation

The two routing network instances as shown in Fig. 3 and Fig. 4 are used to accomplish the task of comparing the performance of EIGRP and OSPF protocols for HTTP and FTP applications. The comparison and evaluation of these two protocols are done based on performance metrics, packet loss and delta time.

Packet loss occurs when packets of data travelling across the network fail to reach the destination. It can be caused by a number of factors, so it is normal in a stable network. HTTP and FTP sessions were performed from the Server to the Client to transfer a large video file whose size is 27,580,668 bytes. Table 2 lists the number of packet loss generated by RVM. Recovery means that while the primary link is shut down (by using RCM) for the subnet between the virtual server and A7 at about 45 second marks, the routings will converge again after certain time periods.

Table 2. Packet Loss

	HTTP/Recovery	FTP/Recovery
EIGRP	65 packets / 90 packets	58 packets / 76 packets
OSPF	87 packets /164 packets	83 packets /137 packets

The following formula is used to compare the rate for the increasing packet loss:

$$Rate = \frac{Recovery\ Packet\ Loss}{Packets\ Loss}$$

For HTTP in EIGRP network,

$$Rate_{HE} = \frac{90}{65} = 1.385$$

For HTTP in OSPF network,

$$Rate_{HO} = \frac{164}{87} = 1.885$$

This indicates that the rate for the increasing packet loss is smaller in EIGRP network than in OSPF network under HTTP traffic.

For FTP in EIGRP network,

$$Rate_{HE} = \frac{76}{58} = 1.310$$

For FTP in OSPF network,

$$Rate_{HO} = \frac{137}{83} = 1.651$$

Similarly, this indicates that the rate for the increasing packet loss is smaller in EIGRP network than in OSPF network under FTP traffic or HTTP traffic.

Delta time is the time between the captured packets. Recovery delta time means the time between the last packet, which is captured before the primary link is interrupted, and the first packet, which is captured after the backup link becomes active. ARTNet RVM generates graphs like Fig. 9 and Fig. 10.

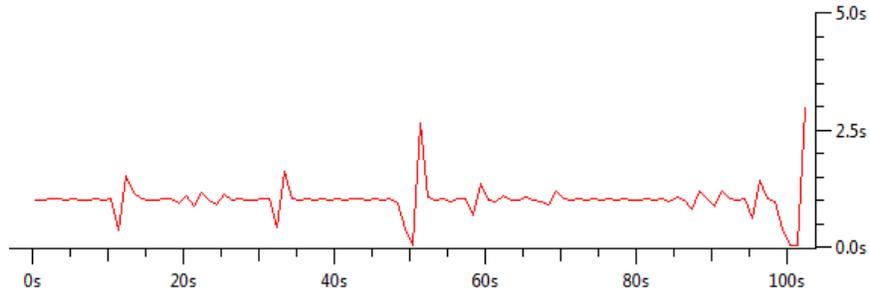


Fig. 9. Recovery delta time for HTTP in EIGRP scenario.

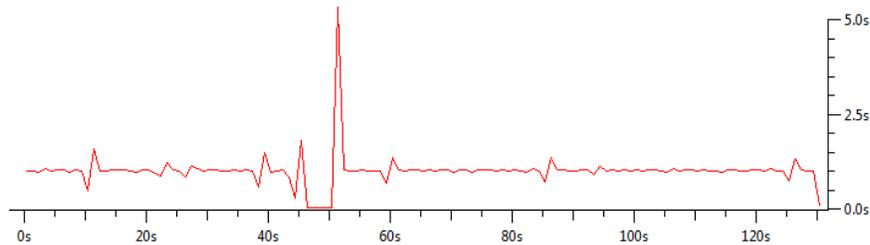


Fig. 10. Recovery delta time for HTTP in OSPF scenario.

By comparing recovery delta time, most of the recovery delta times for both scenarios are the same. The spike in OSPF scenario (5.0 second) is larger than the spike in EIGRP scenario (2.5 second). In other words, EIGRP converges faster than OSPF.

These experiments demonstrate that ARTNet can be used to effectively perform the necessary quantitative comparisons between different routing protocols.

4 Related Work

A number of previous researches have been done to evaluate the performance of interior protocols with application data flows by using either a live network testbed or a simulator. The way of employing a live testbed consists of real routers, switches and computers provides the most realistic results, but is costly to establish and manage. On the other hand, the pure simulator based approach reduces cost, but the results obtained are less realistic.

Some studies have used simulator OPNET to compare EIGRP, OSPF or other protocols for different applications. Kaur and Sharma [16] evaluated the performance of a hybrid network implementing EIGRP, OSPF and BGP routing protocols for different applications in a campus network using the OPNET IT GURU Academic Edition simulator. The applications included Email, FTP, Remote Login, and HTTP & Print Server. Yehia, Aziz, and Elsayed [17] compare EIGRP and OSPF

performances for voice, HTTP, and FTP using OPNET modeler. They measured convergence duration, voice packets delay, and throughput. Lucio, Paredes-Farrera, Jammeh, Fleury and Reed [18] presented a comparative study of two simulators OPNET Modeler and NS-2. Simulator outputs are compared to the output from a live network testbed in a sample network. They used CBR-type traffic and FTP sessions. OPNET is a powerful tool for administrators and researchers to design, secure, analyze, troubleshoot and improve large-scale networks. However, both OPNET and NS-2 lack the real-time control functionality as provided in the ARTNet testbed presented in this paper.

Knežević, Schubert and Kostić [19] proposed a testbed named MX to emulate programmable routers running over a realistic topology on multi-core servers. MX used XORP for the control plane and one physical computer was allowed to behave as multiple routers. Li, Liu and Rangaswami [20] integrated PRIME and XORP. This design used a forwarding plane offloading approach to reduce I/O overload. A case study using OSPF to demonstrate its capability to conduct elaborate routing test was also provided. Petac and Musat [21] built a platform which includes Cisco routers and computers running GNU Zebra and Quagga. They studied the convergence time by interrupting various links in a hybrid network. Nguyen, Roughan, Knight, and Falkner [22] described AutoNetkit to auto-configure the complex emulation network. Quagga was used to provide implementation of routing protocols. Platforms using XORP, Zebra, and Quagga software, provide programmable routers and allow rapid introduction of new protocols, but they do not include the popular proprietary routing protocols which ARTNet supports.

5 Conclusions and Future Work

In this paper, we present ARTNet, a cost-effective, realistic testbed for routing networks. It is scalable, extensible and user-friendly. It supports popular routing protocols and allows users to easily configure networks, control networks during real time and analyze the results after experimentations in a visual way.

We use ARTNet to evaluate the performance of two popular routing protocols, EIGRP and OSPF based on quantitative metrics packet loss for standard application services. The comparative analysis demonstrates that EIGRP converges faster for HTTP and FTP applications, when the primary link for a subnet is interrupted.

For future work, we plan to integrate IM with RCM to make ARTNet more user-friendly and develop an intelligent results analysis module to help users to develop insights from their experimentations.

References

1. GNU Zebra, <http://www.gnu.org/software/zebra/>
2. Quagga Software Routing Suite, <http://www.quagga.net/>
3. XORP Routing Platform, <http://www.xorp.org/>
4. OMNeT++, <http://www.omnetpp.org/>

5. Ns-2, <http://www.isi.edu/nsnam/ns/>
6. Cisco Packet Tracer, http://www.cisco.com/web/learning/netacad/course_catalog/PacketTracer.html
7. OPNET IT Guru, http://www.opnet.com/university_program/itguru_academic_edition/
8. Introduction to EIGRP, http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080093f07.shtml
9. Graphic Network Simulator-GNS3, <http://www.gns3.net/>
10. IP Overview, http://www.cisco.com/en/US/docs/ios/12_0/np1/configuration/guide/1covervw.html#wp4585
11. Software License Agreement, <http://www.cisco.com/public/sw-license-agreement.html>
12. VMware Workstation Datasheet, <http://www.vmware.com/files/pdf/VMware-Workstation-Datasheet.pdf>
13. Wireshark, <http://www.wireshark.org/>
14. Hucaby, D.: CCNP SIWTCH 642-813 Official Certification Guide. Cisco Press. Indianapolis, IN (2010)
15. Campus Network for High Availability Design Guide, http://www.cisco.com/en/US/docs/solutions/Enterprise/Campus/HA_campus_DG/hacampusdg.html
16. Kaur, I., Sharma M.: Performance Evaluation of Hybrid Network Using EIGRP & OSPF for Different Applications. International Journal of Engineering Science and Technology (IJEST), vol.3 no.5, pp. 3950--3960 (2011)
17. Yehia, M.A., Aziz, M.S., Elsayed, H.A.: Analysis of IGP Routing Protocols for Real Time Applications: A Comparative Study. International Journal of Computer Applications, vol.26 no.3, pp. 11--17 (2011)
18. Lucio, G.F., Paredes-Farrera, M., Jammeh, E., Fleury, M., Reed, M.J.: OPNET Modeler and Ns-2: Comparing the Accuracy of Network Simulators for Packet-Level Analysis using a Network Testbed. In 3rd WEAS International Conference on Simulation, Modeling and Optimization, vol.2, pp. 700--707 (2003)
19. Knežević, N., Schubert, N., Kostić, D.: Towards a Cost-Effective Networking Testbed. SIGOPS Operating Systems Review, vol.43 No.4, pp. 66--71 (2009)
20. Li, Y., Liu, J., Rangaswami, R.: Toward Scalable Routing Experiments with Real-Time Network Simulation, Proceedings of the 22nd Workshop on Principles of Advanced and Distributed Simulation, pp. 23--30 (2008)
21. Experimental results about Multiprotocol Routing and Route Redistribution, 6th RoEdu Net International Conference, pp. 83-88, Craiova, Romania, November 2007
22. Nguyen, H.X., Roughan, M., Knight, S., Falkner, N., Maennel, O., and Bush, R.: How to Build Complex, Large-Scale Emulated Networks. Proc. 6th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities, pp. 3--18 (2010)