Journal of Applied Sciences

ISSN 1812-5654

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 13 (22): 5003-5011, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.5003.5011

RNET: A Reconfigurable Network Emulation Testbed Based on Click Modular Router

Zhang Dong and Chen Xi
College of Mathematics and Computer Science, Fuzhou University,
350108, Fuzhou University, China

Abstract: Some innovation network architectures, such as Software Defined Network (SDN) and Information-Centric Networking (ICN) attract more and more interest in recently years. Network virtualization and reconfigurable are the basis of these new network technologies. Besides theoretical study, a large amount of research works should be done to test and verify the assumptions on the testbeds. Some large-scale research network test platforms are designed. But they are not suitable for researchers to carry out small-scale network virtualization experiment, because they are quite complexity or still at primary stages. However, the software platforms are unable to simulate the real network traffic and test the efficiency of application software. Therefore, it would be a better choice to design a small-scale virtualization testbed that cost less and can be easily deployed. In this study, we design a small-scale reconfigurable network emulation testbed called RNET, based on Click Modular Router, especially for the new network paradigms research work. Through a lot of independent experiments which using multiple protocols and different kinds of applications, we conclude that RNET would be an efficient testbed for future network researches.

Key words: Network virtualization, network testbed, virtual network, reconfigurable network, click modular router

INTRODUCTION

After several decades of development, Internet has demonstrated its strong vitality and broad prospects. With the development of network technologies and applications, it is difficult for the traditional architecture of Internet to adapt to variety of network requirements and Quality of Service. Opening the way to perform transformative research into the future Internet has become the direction of network research. Some innovation network architectures, such as Software Defined Network (SDN) and Information-Centric Networking (ICN) attract more and more interest in recently years. Some projects and institutes around the world have carried out relevant work in promoting future Internet architecture and next generation network testbeds such as Future Internet Design (FIND) (http://find.isi.edu), Global Environment for Network Innovations (GENI) (Evans and Ackers, 2007), Planet-lab (Peterson, 2004) in USA, New Generation Network (AKARI) (http://akari-project.nict.go.jp) in Japan and China Next Generation Internet (CNGI,http://www.edu.cn/html/cngi/index.html)

The European Commission has launched the Future Internet Research and Experiment (FIRE) (http://cordis.europa.eu/fp7/ict/fire) initiative in its 7th Framework Program (FP7)Error! Reference source not found. These projects of future Internet researches have different characteristics. CNGI in China provides platform for advanced network technology and applications and its backbone is native IPv6. Planet-lab, a global research network that supports the development of new network services, is the first community testbed that takes virtualization as a main principle. However, it may not be used for research on lower-layer network protocols and architectures. GENI and FIRE allow the creation of experiments that create a version of the network that could have its own management protocols and incompatibilities with the Internet (Chowdhury and Boutaba, 2009).

Network virtualization has become an important evolution direction of future Internet architecture research. Virtual network and network device virtualization are inseparable. This study proposes the combination of network device virtualization and virtual network programming, as well as the construction of future Internet architecture testbed. Meanwhile, new proposals for Internet architecture, protocols and services should not only be defined by studywork. Early experiments and tests in large-scale environments are also needed (Pavlou, 2007; Szegedi, *et al*., 2009). However, the deployment of larger-scale network environment

Corresponding Author: Zhang Dong, College of Mathematics and Computer Science, Fuzhou University, 350108, Fuzhou University, China

supporting virtualization is capital-intensive investment. This study focuses on the problem to construct a simulation testbed based on Click Modular Router for future Internet architecture and services. Although routers based on software architecture can not play a role in the core routers, they can provide good services to small-scale border routing and setting up campus network experiment environment for their good technical expand, low cost and ease of design (Wang *et al*., 2010).

RELATED WORK

In order to be satisfied with the need of future network research, some Internet testbeds based on network virtualization are proposed, these testbeds have different characteristics and advantages.

VINI provides network researchers a realistic and controlled network environment to evaluate their innovation protocols and new types of network services, by using the virtualization technology on the operating system-level and a variety of open source virtualization tools (Bavier *et al*., 2006). VINI allows researchers to deploy their ideas with real routing software, realistic network conditions, real traffic loads and network events, in addition, it supports multiple experiments run in parallel and the real connection to the Internet (Bavier *et al*., 2006). ShadowNet (Chen *et al*., 2009) is a sharable, programmable and composable infrastructure, consisting of carrier-grade equipment. Compare to VINI that is based on software router, ShadowNet takes a different virtualization mechanism, it allows multiple logical routers to be executed in parallel in a real physical router through resource virtualization technology to support higher throughput and forwarding rate.

OpenFlow (McKeown *et al*., 2008) is added as a feature to commercial Ethernet switches and routers. The advantage of the OpenFlow is that it achieves a pragmatic compromise: on one hand, it allows researchers modify the network control logic to conduct innovative experiments. While on the other hand, equipment manufacturers do not need to expose the internal implementation details of network devices. OpenFlow (McKeown *et al*., 2008) could serve as a useful campus component in proposed large-scale testbeds like GENI. The FlowVisor (Sherwood *et al*., 2010) is a special purpose OpenFlow controller. It slices the OpenFlow network by adding a layer between the control plane and the data plane, which allows multiple control planes to share the same data plane with strong isolation. By using multiple and isolated slices in the same physical network, it solves the problem that OpenFlow network can only carry out an innovation experiment at a time.

Onix (Koponen *et al*., 2010) and Nicira (Casado *et al*., 2010) provide the virtualization capabilities on the network control layer and the network-forwarding layer respectively. Onix is a platform implemented between the control logic and the underlying network infrastructure. Onix provides a general API for network researchers to focus on the development of control plane, which resolve some tedious problems such as state distribution, element discovery and failure recovery mechanisms. Nicira achieve the virtualization of network forwarding plane, the main idea is to use a network hypervisor, which is similar to the hypervisor implement in host. Nicira provides a standard forwarding layer service model for the control logic on the upper layer and realize the desired functionality on the hardware layer below.

Compare to these testbeds mentioned previous, RNET, proposed on this study, has its unique characteristic: Software-based and modular, which add the flexibility to innovate experiment.

NETWORK VIRTUALIZATION

A networking environment supports network virtualization if it allows coexistence of multiple virtual networks on the same physical substrate (Chowdhury and Boutaba, 2010). Network virtualization is composed of two main aspects: network device virtualization and virtual network. In the network virtualization environment, router devices are classified into core router nodes supporting virtualization and edge routers that don’t support virtualization. Logical virtual router technology divides a physical router into different logical routers, which run one or more virtual routers to support new business models and new applications (Yang, 2009).

Network virtualization environment, as shown in Fig. 1, consists of edge routers and core routers supporting virtualization, which are connected by virtual links. Different application communication services are aggregated through the edge routers into the network virtualization environment. Different business and users are running in isolated virtual networks. These virtual networks share the same physical infrastructure, such as router switch devices and physical links. In this virtualization environment, users can design their virtual networks, according to business characteristics and application requirements, through running defined virtual routers. Flexibility is a key requirement of any virtual networking substrate (Unnikrishnan *et al*., 2011). In the virtual low-layer network, users can reconfigure future Internet architecture, design new network policy, deploy particular route and communication protocols and

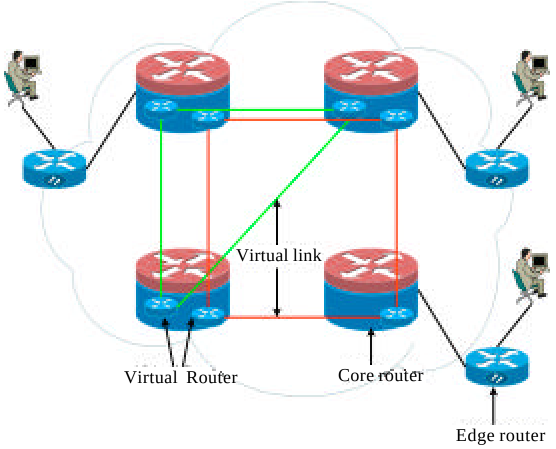

Fig. 1: Network virtualization environment

Table 1:Virtual network testbed characteristics

Network virtualization components objects	Attribute
Logical network of users' requirements	Node, bandwidth, protocol, delay, start time and end time, hops
Virtual network	Virtual network ID, virtual link, node, bandwidth, protocol, start time and end time, route real-time traffic
Physical node	Physical node ID, cache, port number and protocol
Virtual node	Virtual node ID, cache, virtual port/protocol

program feasible network topology. From “low-layer” to “high-layer”, all the things can be reconfigured and designed. Different virtual network are independent segregation. Route protocols and network architecture don’t affect each other, but share the same substrate.

As a future Internet testbed, its objective is to construct business-oriented virtual network and supply a simulation environment and experiment tools to researchers. A testbed service management system, according to business characteristics and user requirement, creates a virtual router in core router node and designs the appropriate network topology, allocates the links’ bandwidth and establishes the virtual links. Meanwhile, users can set route protocol, monitor node and measure data.

RECONFIGURABLE NETWORK EMULATION TESTBED ARCHITECTURE

In future, Internet testbed based on network virtualization environment management system is expected to build virtual network topology and allocate resources, namely link bandwidth and node computational capability. The list of parameters that characterizes virtual networks, nodes and virtual links is shown in Table 1.

Four main components and their attributes, based on network virtualization testbed, are listed in Table 1. Logical network of users’ requirement explains the requirements for constructing network on the testbed, include node, bandwidth, protocol, delay, start time and end time and hops. Some QoS related attributes are optional. Physical node contains the parameters supporting virtualization. These attribute parameters support scalability.

In the process of designing the testbed architecture, we focused on achieving the following goals:

- **Reconfigurable:** The system must support multiple protocols reconfiguration. A variety of different protocols, applications will run simultaneously on the underlying physical network. It is necessary to provide a reconfiguration support so that different protocols and applications would not interfere with each other. By using protocols reconfiguration, the testbed can support protocols online switch at the same time. Furthermore, the reconfiguration of protocols provides a pretreatment mechanism for network virtualization
- **Flexibility:** Users can conduct the experiments related to future network architecture. The application and service face to the future network architecture defined by the users may not be tested in the existing stable network environment. Due to the virtualization of network testbed, these experiments can be deployed and its performance can be verified, specifically, the users can receive maximum control over their network experiment. The testbed should provide the flexibility to researchers to select their routing protocols (including any modification) in a virtual network environment
- **Scalability:** The testbed should be structured to allow modification and scalability, which mainly reflect on the construction and demolition about the

virtual network. On the basis of the original testbed, some users may put forward the demand for embedding new virtual networks, while some virtual network should be remove from the underlying network when the service provided is end. The experiment testbed should provide a mechanism to support adding and removing virtual network. Scalability means that the testbed should be able to simultaneously support a greater number of programmable virtual networks

The simulation testbed environment is composed of four major components: Core software routers based on Click Modular Router, management system, mesh links and users client. The core routers support multi-protocols reconfiguration and bandwidth reservation. The management system can manage all network resources, such as bandwidth, router state, etc. Users submit network requirements or monitor the state of virtual network through end-user clients. Management system decides whether virtual network requirement should be accepted or rejected.

Figure 2 explains IPv4/IPv6 protocols reconfiguration element which provides multi-protocols reconfiguration and bandwidth reservation element that is used to construct independent virtual network. Virtual network route element supports each virtual network running its route protocol and communicating in its inner network. Management system consists of three main functions: resource awareness, network construction and state detection.

SIMULATION BASED ON CLICK MODULAR ROUTERS

In the spirit of Click's fine-grained modular approach, this study presents VN-Click, a platform designed to provide network virtualization service based on the Click Modular Router. By using this virtual network resource we can build multiple virtual network on a single hardware system. Originally, the VN-Click software platform is designed to run on with these following properties mentioned on Section III: (1) Reconfigurable, (2) Flexibility, (3) Scalability.

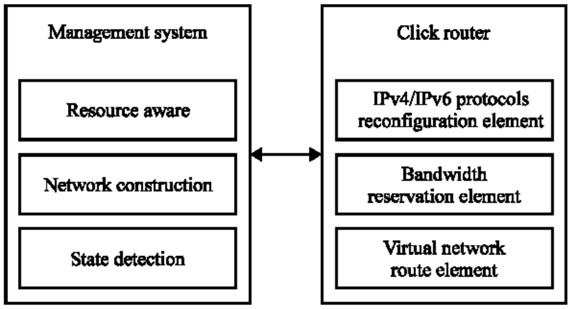
Management system
Resource aware
Network construction
State detection
Click router
IPv4/IPv6 protocols reconfiguration element
Bandwidth reservation element
Virtual network route element

Fig. 2: Management system click router components

In addition, the software system is designed to meet some extra goals:

- **Speed:** Packets should be forwarded at high speed in the virtual network, it can achieve a high performance on small-scale network experiment environment setting up on campus
- **Low cost:** Because it can run on a normal personal computer, which would decrease the cost of hosting virtual networks

Click modular router: Click modular router is new software architecture for building flexible and configurable routers (Kohler *et al*., 2000). A Click router is assembled from packet processing modules called elements. Each element completes a routing function such as packet classification, queuing, scheduling and interfacing with network device. A router configuration is a directed graph with elements at the vertices; packets flow along the edges of the graph. This is the so-called flow-based router context. The flow-based context allows an element to easily locate other elements, on which it depends, extract important information about them (Ye and MacGregor, 2007).

Click is implemented based on Linux, which has already contained over 200 modules (Kohler *et al*., 2000). The behavior of individual elements can be described in C++, each component is a C++ object. These components implement reading data packets from devices, writing data to devices, routing table querying and other functions. To build a router configuration, users can choose a collection of different modules according to their own needs and connects them into a directed graph. To extend a configuration, users can also use C++ to compile a new module or compose existing elements in new ways, in order to achieve user's specific functions.

Figure 3 shows how Click diagram these properties for a sample element, Tee(2). "Tee" represent the element class; a Tee copies each packet received on its single input port, sending one copy to each output port. Click draw input ports as triangles and output ports as rectangles. Configuration strings are enclosed in parentheses: The "2" in "Tee(2)" is interpreted by Tee as a request for two outputs. Method interfaces and handlers are not shown explicitly. They are specified implicitly by the element class.

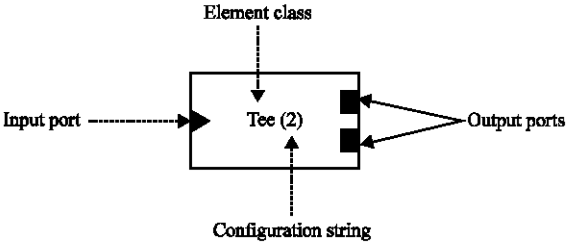

Fig. 3: A sample element

Click supports two kinds of connections, push and pull. On a push connection, packets start at the source element and are passed downstream to the destination element. On a pull connection, in contrast, the destination element initiates packet transfer: it asks the source element to return a packet, or a null pointer if no packet is available.

Ipv4 router: Several elements can be interconnected to form a router. For example, Figure 4 shows a simple IPv4 router instance that will be used on the network virtualization testbed. The software router accepts data packets from a network device via the (FromDevice) element, then checks each packet's IP header (CheckIPHeader), it checks that the packet's length is reasonable and that the IP version, header length and checksum fields are valid, filters non-IP packets, invalid packets are discarded (Discard) as well as the broadcast packets (DropBroadcasts), the correct packets are forwarded to the next component (LookupIPRoute), which looks up each packet's destination annotation address in the routing table and emits the packet on an output that depends on the interface device mentioned in the routing table entry, then (DecIPTTL) decrements the TTL field in the packet, before the packets are forwarded to the physical interface device via the (ToDevice) element. They may also pass through some components that used for error control (ICMPError), which are not shown on the figure below, it generates ICMP error packets for the types such as "Destination Unreachable", "Redirect Message", "Time Exceeded", "Parameter Problem" and so on. A software router that supports IPv6 has a similar architecture: By replacing the elements CheckIPHeader, LookupIPRoute, DecIPTTL in Fig. 4 to CheckIP6Header, LookupIP6Route and DecIP6HLIM.

Configurations can be formulated by Click style description. An example of the Click style description of the sample router in Fig. 4 is shown in Table 2.

We have extended Click's range of functions by implementing some elements that handle reconfiguration. The testbed supports multiple execution management and handles communication with protocol reconfiguration.

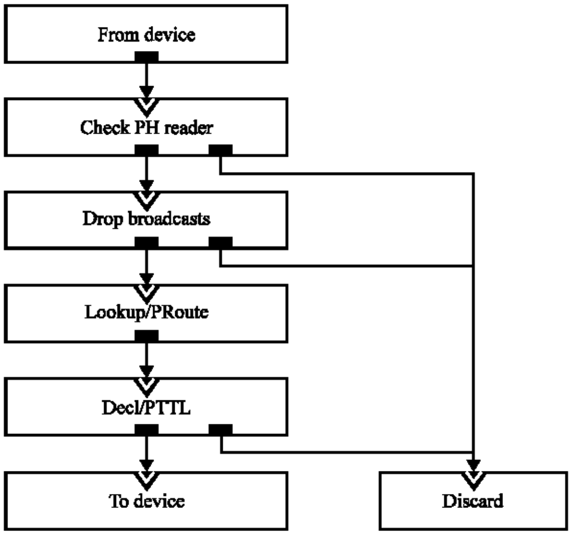

Fig. 4: A sample router

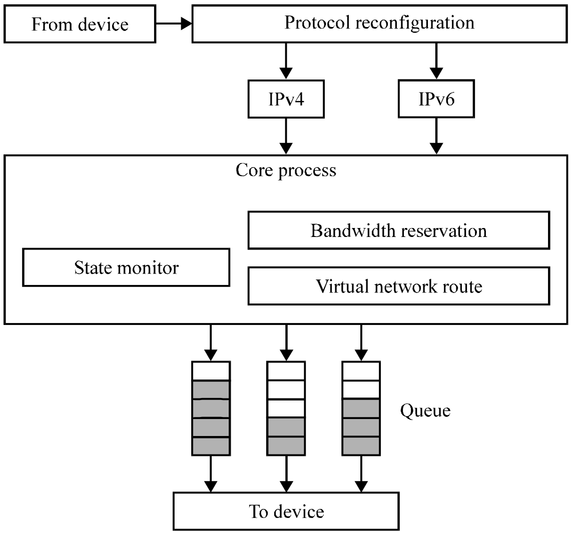

Fig. 5: VN-click router components

Table 2:Click description of the sample router in Fig. 4

Instantiate components	Src:: FromDevice ();
	Check:: CheckIPHeader ();
	Broad:: DropBroadcasts ();
	Route:: LookupIPRoute ();
	DecIP:: DecIPTTL ();
	Dst:: ToDevice ();
	Dis:: Discard ();

Instantiate components instances	Src->Check->Broad->Route->DecIP->Dst;
	Check [1]->Dis;
	Broad [1]->Dis;
	DecIP [1]->Dis;

The testbed provides active processing service for virtual network experiment. Its structure is shown in Fig. 5. Protocols should be reconfigurable and prevented from accessing data held by other protocols. This is achieved simply based the fact that each virtual network runs as a separate process with its own address space and hence is subject to memory boundary limitations. In order to provide protocols separation on the data plane but share the same physical network device, in the testbed we used the element (Classifier) to reconfigure different protocols.

The VN-Click router receives network data packets through a physical network interface, all the packets mainly include ARP requests, ARP replies, Neighbor Solicitation Message, Neighbor Advertisement Message, IPv4 and IPv6 packets that are useful for the virtualization experimental environment. As the program shows the ARP requests and ARP replies used to map IP address and Ethernet address are sent to output 0 and output 1 of the Classifier respectively. (ARPResponder) accept ARP request packets and generates responses to ARP queries if it knows the requested IP address, which outputs an reply giving the corresponding Ethernet address. (ARPQuerier) handle most of the arriving ARP replies, the mapping of the IP address and Ethernet address that router need is recorded. (IP6NDAdvertiser) and (IP6NDSolicitor) play a similar role on the Neighbor Discovery protocol designed for IPv6, which process Neighbor Solicitation Message and Neighbor Advertisement Message that come from port 3 and port 4 of the Classifier. Neighbor Solicitation Message includes the Ethernet header, IPv6 header as well as ARP requests and the Neighbor Advertisement Message provides the Ethernet address mapped to the IPv6 address that router require. The rest of the IP packets are emitted through export 2 or export 5 by matching their protocol type.

The configurations of the elements that support protocols reconfiguration based on IPv4 and IPv6 can be described as follow in Table 3.

By subdividing the packets, the VN-Click can individually enabled traditional IP version 4 protocol or later version 6 protocol and more provides a mechanism of coexistence between IPv4 and IPv6 which may not exist on the traditional hardware routers.

After protocols reconfiguration by the particular component, the packets will be delivered to a more complex and core processing, which contain bandwidth reservation, virtual network routing and status monitoring, which will be described in a later section combine with the experiment.

Table 3: Click description of the protocols reconfigure in Fig. 5

Instantiate components	Class:: Classifier (12/0806 20/0001, 12/0806 20/0002, 12/0800, 12/86dd 20/3aff 54/87, 12/86dd 20/3aff 54/88, 12/86dd);
	ARPRe:: ARPResponder (IP address Ethernet address);
	ARPQu:: ARPQuerier (IP address, Ethernet address);
	IPNDA:: IP6NDAdvertiser (IP6 address Ethernet address);
	IPNDS:: IP6NDSolicitor (IP6 address, Ethernet address);

Instantiate components instances	Class [0]->ARPRe;
	Class [1]->[1] ARPQu;
	Class [2]->IPv4 Process…;
	Class [3]->IPNDA;
	Class [4]->[1] IPNDS;
	Class [5]->IPv6 Process…;

Experiment topology: To measure performance of the reconfigurable testbed based on Click Modular Router, we have established a following experimental network topology shown in Fig. 6, with which running four software routers and six personal computers. The Click routers are running on a Linux box which includes a 2.6 GHz Intel Pentium G620 processor, 3 GB of RAM and 4 USB 2.0 Fast Ethernet Adapter with the rate of 100 Mbps.

Upon the basis experiment topology, we construct three virtual networks, which named virtual network A, B and C described in Fig. 6. Initially, there are two virtual routers in every Node 1, Node 2 , Node 3 and Node 4 and these virtual routers belong to different virtual network: Node 1 provides services for virtual network A and B, while Node 2 provide routers for virtual network B and C, Node 3 and Node 4 are similar to the case. On network virtualization environment, The virtual routers are independent of each other and can run different routing, addressing and forwarding schemes. A physical link can be virtualized same as the physical node, which is multiplexed into virtual links and connect the relevant virtual routers into virtual networks (Unnikrishnan *et al*., 2010). It can be seen from the figure that the physical link between Node 1 and Node 2 offer the connection for virtual network A and virtual network C.

Virtual node: Virtual nodes are built inside the VN-Click node. It allows researchers to host multiple virtual networks on shared hardware (Cherukuri, *et al*., 2011). When a virtual network is embedded in the underlying physical network, it will consist of a series of virtual nodes. At the same time, each virtual node is colored by the virtual network. The VN-Click allows researchers to program their virtual topology based on their requirement. It also allows the researchers to select routing protocols and allows them to define their own forwarding tables.

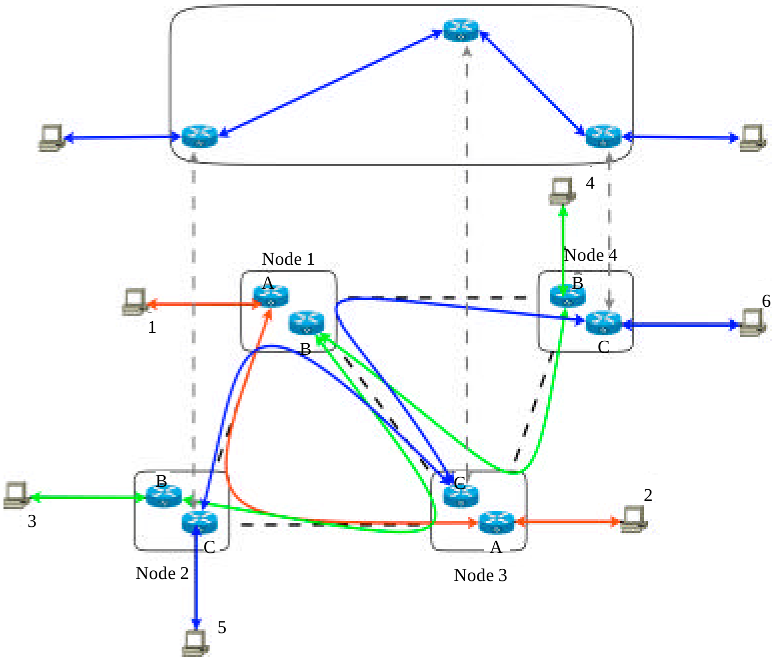

Fig. 6: VN-click experiment topology

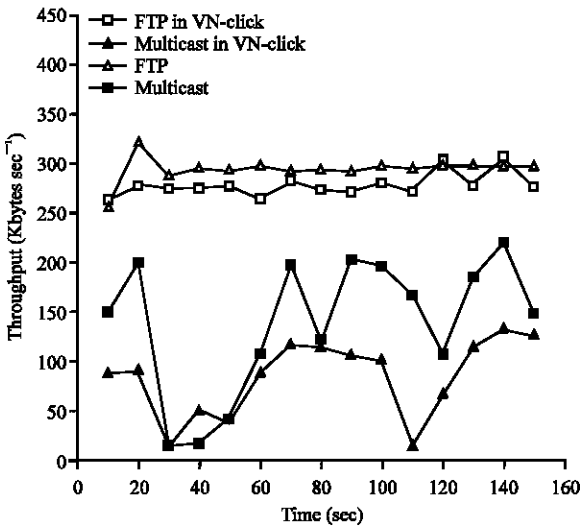

Fig. 7: Throughput of FTP and multicast in VN-click

Virtual link: Virtual links give an illusion of a direct physical link between two virtual nodes, although they may be situated at multiple hops away (Cherukuri *et al*., 2011). In VN-Click, virtual links will be created between virtual nodes when the virtual network is constructed. Once a virtual link is created, it should provide a particular bandwidth specialized by the user. By using these virtual links, different packets belong to different virtual network will be correctly transmission between the virtual nodes. In our implementation of virtual network on the testbed based on Click Modular Router, Differential Services Code Point (DSCP) in the IP packets plays an important role on identifying the virtual network. First, Packets flow on the edge virtual node of a particular virtual network and then these packets are set a unique identifier by component (SetIPDSCP). Other virtual nodes of the same virtual network can recognize this mark. When the packets leave a virtual network, the identification will be cleared. The types of applications or service running on the different virtual network may be not the same, which require different computing capacity of the virtual nodes as well as bandwidth limit of the virtual links. Element (BandwidthShaper) provide the ability to limit the bandwidth, it should be placed at the end of the virtual router and before packets enter a virtual link.

EVALUATION

We have conducted some tests to evaluate RNET performance as a whole in multi-protocols network. We experiment with FTP applications and multicast applications.

Figure 7 describes the throughput of instantiating, respectively the FTP and multicast virtual networks,

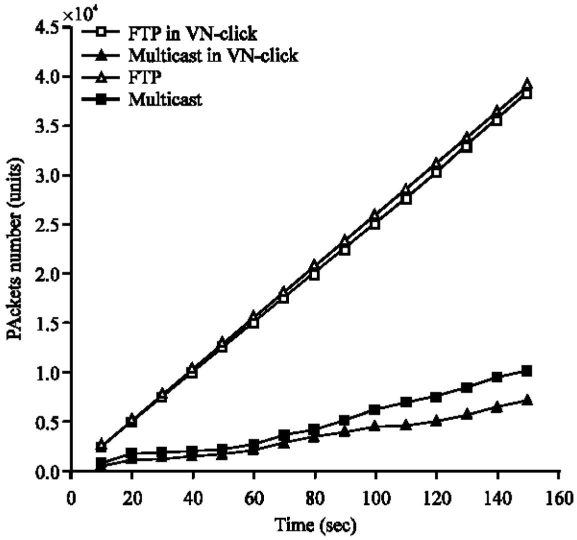

Fig. 8: Packets number of FTP and multicast received in VN-click

which run on the testbed concurrently. Since the different characteristic of the FTP application and multicast application, FTP service is a relatively stable transport business; by contrast, multicast service is affected largely by the quality of video source, thus showing a greater volatility.

In terms of the bandwidth resource, FTP takes up more weight than multicast; also, the throughputs in VN-Click are lower than normal, which represent bandwidth reservation of the system. According to different needs for bandwidth of these two businesses, the throughput of FTP in VN-Click is limited at 300Kbytes/s to support fast transmission. Meanwhile, multicast application is limited to 120Kbytes/s to provide a smooth video playback.

Figure 8 shows packets number received by FTP and multicast in a period of time, from which we can intuitively find that FTP packets increase close to a straight line, while multicast packets behaved tortuous. In VN-Click, the number of packets received is slightly less than that received in physical network, which is due to bandwidth limitation of the system. Our results reveal that VN-Click realizes a balance of isolation, coexistence and controllability between virtual networks.

CONCLUSION

Network Virtualization is the hotspot of research on future Internet architecture. Network device virtualization and virtual network are inseparable. Researches on virtual network and physical device virtualization are at the forefront phase. But there are some shortcomings and challenges in developing network device. It's difficult to conduct early experiments and test in large-scale environments. The main purpose of this study is to provide efficient support for future Internet architecture testbed. We have proposed architecture of future network testbed environment based on network virtualization. RNET model using Click Modular Router has been introduced in this study. The numerical results of running different protocol services indicate that the model proposed support efficient experimental environment.

In the future, we plan to focus on network virtualization and virtual network programming. Another open question that we plan to investigate is how to provide more tools for researchers to manage the virtual network in the testbed.

ACKNOWLEDGMENT

This study is supported by the Natural Science Foundation of Fujian Province, China under (2011J05150, 2013J01231).

REFERENCES

Bavier, A., N. Feamster, M. Huang, L. Peterson and J. Rexfordetc, 2006. In VINI veritas: Realistic and controlled network experimentation. Proceedings of the Conference on Applications, Technologies, Architectures and Protocols for Computer Communications, October 2006, New York, USA., pp: 3-14.

Casado, M., T. Koponen, R. Ramanathan and S. Shenker, 2010. Virtualizing the network forwarding plane. Proceedings of the Workshop on Programmable Routers for Extensible Services of Tomorrow, March 8-15, 2010 New York, USA.

Chen, X., Z.M. Mao and J.V.D. Merwe, 2009. ShadowNet: A platform for rapid and safe network evolution. Proceedings of the International Conference on USENIX Annual Technical, June 14-19, 2009, Berkeley, CA., USA., pp: 3-3.

Cherukuri, R., X. Liu, A. Bavier, J.P.G. Sterbenz and D. Medhi, 2011. Network virtualization in GpENI: Framework, implementation and integration experience. Proceedings of the International Symposium on Integrated Network Management, May 23-27, 2011, Dublin, pp: 1216-1223.

Chowdhury, N.M.K. and R. Boutaba, 2009. Network virtualization: State of the art and research challenges. Commun. Mag., 47: 20-26.

Chowdhury, N.M.M.K. and R. Boutaba, 2010. A survey of network virtualization. Comput. Networks, 54: 862-876.

Evans, J.B. and D.E. Ackers, 2007. Overview of GENI and future internet in the US. GENI, Los Angeles, California, USA., May 22, 2007. http://www.geni.net.

Kohler, E., R. Morris, B. Chen, J. Jannotti and M.F. Kaashoek, 2000. The click modular router. ACM Trans. Comput. Syst., 18: 263-297.

Koponen, T., M. Casado, N. Gude, J. Stribling and L. Poutievski et al., 2010. Onix: A distributed control platform for large-scale production networks. Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation, October 4-6, 2010, Vancouver, BC, Canada, pp: 1-6.

McKeown, N., T. Anderson, H. Balakrishnan, G. Parulkar and L. Peterson et al., 2008. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Comput. Commun. Rev., 38: 69-74.

Pavlou, G., 2007. On the evolution of management approaches, frameworks and protocols: A historical perspective. J. Network Syst. Manage., 15: 425-445.

Peterson, L., 2004. Planet Lab: A blueprint for introducing disruptive technology into the internet. http://www.planet-lab.org/node/263

Sherwood, R., G. Gibb, K.K. Yap, G. Appenzeller, M. Casado, N. McKeown and G. Parulkary, 2010. Can the production network be the testbed? Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation, October 4-6, 2010, Vancouver, BC., Canada, pp: 1-14.

Szegedi, P., S. Figuerola, M. Campanella, V. Maglaris and C. Cervello-Pastor, 2009. With evolution for revolution: Managing FEDERICA for future Internet research. Commun. Mag., 47: 34-39.

Unnikrishnan, D., J. Lu, L. Gao and R. Tessier, 2011. Reclick-a modular dataplane design framework for fpga-based network virtualization. Proceedings of the 7th International Symposium on Architectures for Networking and Communications Systems, October 3-4, 2011, Brooklyn, New York, pp: 145-155.

Unnikrishnan, D., R. Vadlamani, Y. Liao, A. Dwaraki, J. Crenne, L. Gao and R. Tessier, 2010. Scalable network virtualization using FPGAs. Proceedings of the 18th Annual ACM/SIGDA International Symposium on Field Programmable Gate Arrays, February 21-23, 2010, ACM, New York, USA., pp: 219-228.

Wang, Z.k., F. Yang, J.Y. Chen and Y.J. Liu, 2010. Load-balancing high-performance router testbed based on openflow and click. Proceedings of the International Conference on Computational Intelligence and Software Engineering, December 10-12, 2010, Wuhan, pp: 1-4.

Yang, Y., 2009. Some key technology of reconfigurable network routing platform. Proc. J. Inform. Eng. Univ., 10: 34-36.

Ye, Q.H. and M.H. MacGregor, 2007. Click on a cluster: A viable approach to scale software-based routers. Proceedings of the International Conference on Communications, June 24-28, 2007, Glasgow, pp: 6392-6399.