

Planning and Implantation of NetFPGA Platform on Network Emulation Testbed

Pang-Wei Tsai¹, Pei-Wen Cheng¹, Mon-Yen Luo², Te-Lung Liu³ and Chu-Sing Yang^{1,*}

1 Institute of Computer and Communication Engineering, Department of Electrical Engineering, National Cheng Kung University / No.1, University Rd., East Dist. Tainan City, Taiwan

2 Department of Computer Science & Information Engineering, National Kaohsiung University of Applied Sciences / 415 Chien Kung Road, Sanmin District, Kaohsiung City, Taiwan

3 Tainan Branch of National Center for High-Performance Computing / No. 28, Nan-Ke 3rd Rd., Hsin-Shi Dist., Tainan City, Taiwan

E-Mails: cache@ee.ncku.edu.tw; peiven.cheng@gmail.com; myluo@cc.kuas.edu.tw; tlliu@nchc.narl.org.tw; csyang@ee.ncku.edu.tw

* 92633R, EE Department Building, Tze-Chiang Campus, National Cheng Kung University, No.1, University Rd., East Dist. Tainan City, Taiwan 701, Taiwan; Tel.: +886-6-2757575-62357; Fax: +886-6-234-5482

http://dx.doi.org/10.7125/APAN.32.1 ISSN 2227-3026

Abstract: The concepts of cloud computing and Internet applications have expanded gradually and have become more and more important. Researchers need a new, high-speed network to build experimental environments for testing new network protocols without affecting existing traffic. In this paper, we describe a way to integrate NetFPGA platform, OpenFlow concept and NetFPGA reference designs into a network testbed to improve the packet processing speed and the dynamic adjustability for network emulation experiments. Furthermore, combined with Tunneling and VPLS, the proposed network testbed can be connected to distributed network, thus providing researchers a traffic-controllable and NIC-programmable experimental networking testbed in intra-communicating part.

Keywords: Network Testbed; Network Emulation; NetFPGA.

1. Introduction

For the next generation network development, the concepts of cloud and future internet application have expanded gradually and have become more and more important. Nowadays, with increasing need for high speed, high availability, and high stability of network, researchers have injected a lot of effort into developing new protocols and new applications. Using network simulators such as ns2 [1] to make an environment for experiments is one way. However, network simulator could not satisfy researches related to complicated topology and large-scaled application experiments. For now, network emulation came out to provide one option for researchers to experience a more real experiment environment.

Deploy a network testbed with NetFPGA Platform

In order to establish emulated network for intra-network communication experiment and investigate the cutting-edge Future Internet technologies issue, in this paper, we describe a way of integrating NetFPGA [2] platform with OpenFlow [3] concept into a network testbed based on Emulab [4], which is a time- and space-shared emulator which fulfills the requirements of being quarantined, closed, controllable and resource sharing of hardware.

Customized NetFPGA nodes and pre-configured OS images

NetFPGA is a high-speed, flexible, and open platform for network research which designed by Stanford University. With improvements of high-speed packet processing and dynamic adjustability, nodes with NetFPGA card will provide researchers a wide range of environments to develop, debug, and evaluate their network systems. Users can load pre-configuration OS images which build-in different reference designs into NetFPGA nodes and testing their ideas in a near-real environment without affecting the production network traffic [5].

Initial work for supporting large-scale experiment

To make an extension with intra-network communication issue, we are planning to combine with both Tunneling and VPLS techniques, and making a virtualized network as LAN- Extension. So as to provide researchers the possibility of creating large-scaled network and security experiments with NetFPGA nodes in different area.

The rest of this paper is organized as follows: Section 2 describes the architecture and our concept of planning and developing the experimental network testbed at NCKU; Section 3 presents the process of our approach; Section 4 makes a conclusion of our network and demonstrates the results.

2. System Concepts

2.1 Testbed at TWISC

Taiwan Information Security Research and Education Center, under the support of NSC of Taiwan, has deployed a security testbed based on Emulab platform which is developed by the Flux research team from the University of Utah. The emulation-ability of Testbed@TWISC (206 nodes and 5 switches for experiment) is quite appropriate for network researchers since it meets the requirements of quarantined, closed, and controllable. In this paper, we deployed another minor branch for experimental development at National Cheng Kung University (Testbed@NCKU) , and integrated NetFPGA platform which brought in more functions with network concepts for building experimental environment.

2.2 NetFPGA

NetFPGA is a high-speed, flexible, and open platform for network research; it contains four Gigabit Ethernet interfaces and a Xilinx Virtex-II FPGA programmed with user-defined logic as figure 1 shows. NetFPGA can support many reference designs[6] released under open source license, e.g., reference router, reference NIC, and OpenFlow switch[7], etc. Therefore, we integrated NetFPGA into Testbed@NCKU to implement network emulation scenarios for users.

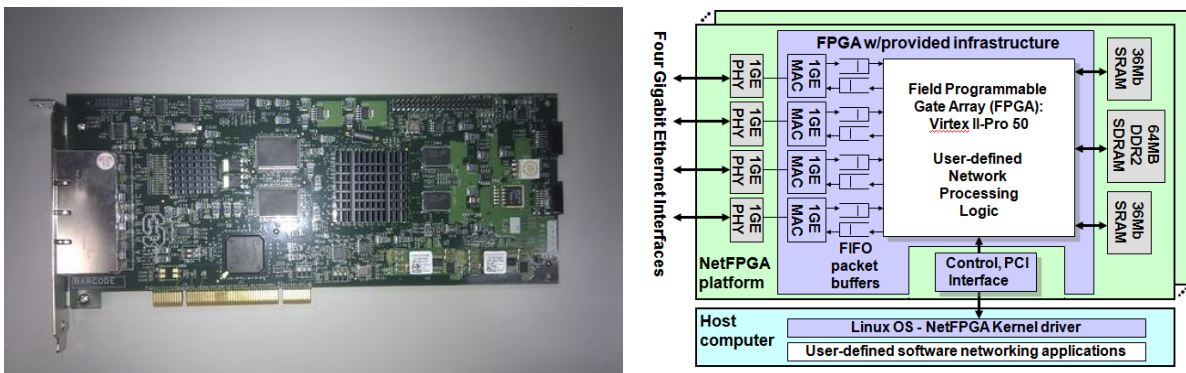


Figure1. NetFPGA Card and Design Architecture [8]

2.3 OpenFlow

OpenFlow is a way to deploy experimental or new protocols in networks that carry production traffic. OpenFlow is proposed by Stanford University and it provides researchers to control the network traffic according to their design. We've also made a customized OS images with OpenFlow software for providing users to create related experiment on nodes.

3. Testbed@NCKU

Testbed built at NCKU is a minor network-emulated testbed based on Emulab. It provides researchers an environment with self-defined topology, multi-function nodes and multi-setup options to construct flexible experiments. With NetFPGA and OpenFlow concepts, more flexibility and expandability are enhanced to support network emulation. Figure 2 shows the infrastructure of the Testbed@NCKU.

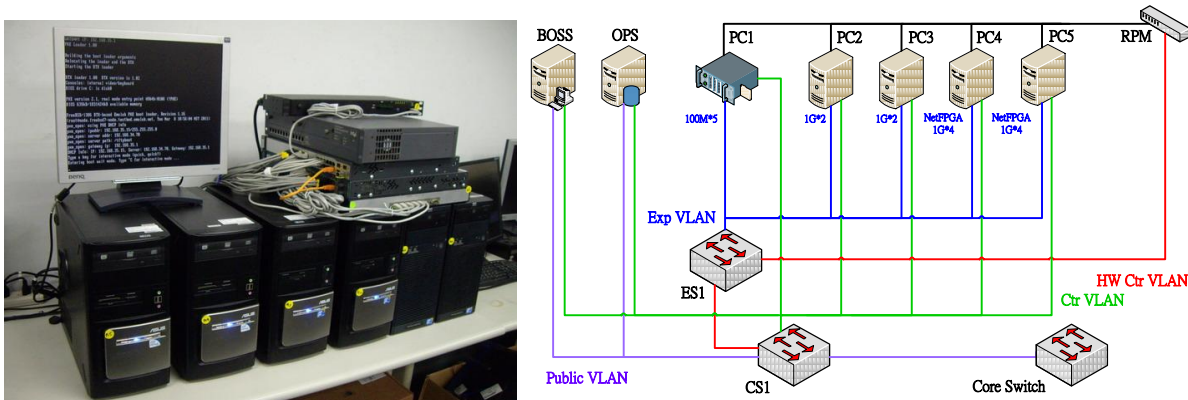


Figure2. Infrastructure of Testbed@NCKUEE

In following segments, we will introduce the operation tools, options, and the flexibility connection setup both in intra-connection and in inter-connections between testbeds.

3.1 Build-in Configuration for NetFPGA Nodes

Although NetFPGA is low-cost hardware platform, but it still take much time to setup for experiment on general PCs. For the purpose of training and education, we try making the setup steps easily and achieving the goal of resource-sharing. Experiment-NICs on these nodes are replaced with NetFPGA card. Figure 3 shows the interface information on Emulab’s management webpage. When the node is booting, a serial configuration script will be executed for CPCPI[9] reprogramming and default reference design loading on NetFPGA card.

Interface	Model; protocols
nf2c0:	nf2c (Digilent EtherExpress Pro; ethernet)
nf2c1:	nf2c (Digilent EtherExpress Pro; ethernet)
nf2c2:	nf2c (Digilent EtherExpress Pro; ethernet)
nf2c3:	nf2c (Digilent EtherExpress Pro; ethernet)

Figure3. NetFPGA Interface Information on Emulab’s Management Webpage

3.2 Customized OS Image for NetFPGA Node

We have the NetFPGA node related installation processes burned into an image file by Frisbee [10], a disk image generation tool, which can allocate operating system into node as user requested. Images that users can load on their nodes on Testbed@NCKU are listed in Table 1.

Table 1. Customized Image List of Testbed@NCKU

Image List of Testbed@NCKU			
Stable		Still Under Testing	
Image Name	Description	Image Name	Description
CentOS56-STD	CentOS 5.6	CentOS56-NF-REFROUTER	CentOS 5.6 with NetFPGA - Reference Router
CentOS56-NF-REFNIC	CentOS 5.6 with NetFPGA - Reference NIC	CentOS56-NF-OF	CentOS 5.6 with NetFPGA - OpenFlow Switch
CentOS56-NF-PKTGEN	CentOS 5.6 with NetFPGA - Packet Generator	CentOS56-NF-Tunnel	CentOS 5.6 with NetFPGA - Tunnel

3.3 Comparison Between Testbed@TWISC and Testbed@NCKUEE

Configuration Aspect

Here we take the packet generator as example. In testbed@TWISC, we need to install tools like Nemesis [11] and IDSwakeup [12] and all the other related packages which cost users lots of time. While in Testbed@NCKU, we can easily download the bit file of NetFPGA packet generator. Packet generator will replicate packets from log file after execution. For more, NetFPGA packet generator provides more functions such as specifying the transmitting rate and delay. All above indicates that experiments with NetFPGA node helps reduce the setup time and increase the resiliency.

Packets Forwarding Efficiency

In Testbed@TWISC, using common node as router could only configure it as a software router. Compared to software-based router which maintains its table in Linux kernel, NetFPGA reference router will write the forwarding table onto NetFPGA card's memory. With forwarding table maintaining in hardware, input packet can quickly be matched with the entry and be forwarded without the need to trap, increasing the packets forwarding ability and efficiency.

3.4 Intra-Network Emulating with NetFPGA gateway

Along with these NetFPGA nodes' functions depicted in above section, the Testbed@NCKU added more flexibility and extendibility of topology in intra-connection. Figure 4 shows a simple topology which can be built in the Testbed@NCKU. It indicates the important role of intermediate node which can perform several functions as researchers' experiments required.

3.5 Planning LAN-Extension for Inter-Network Experiment

Using VPLS technique / L3 Tunnel, Testbed@NCKU will be able to connect to Testbed@TWISC through TWAREN [13] backbone currently. For one, users can setup local experiment inside the testbed. For the other, nodes will be able to emulate inter-domain issue between two Testbeds. We are planning to establish customized gateways for realizing a management mechanism on two emulated networks such as link state routing or distance vector routing experiments. As Figure 5 shows, the scale of experiment would be extended.

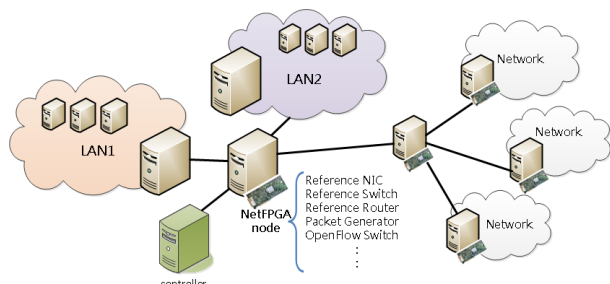


Figure 4. Intra-connection Scenario

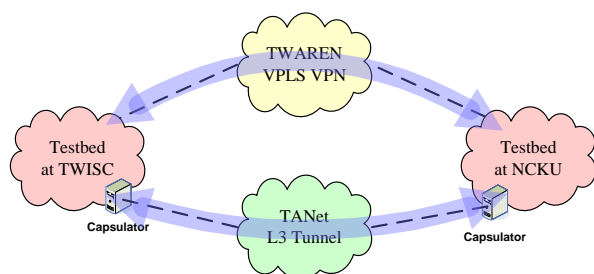


Figure 5. Connection via VPLS or L3 Tunnel

4. Conclusion

We've integrated NetFPGA Platform with OpenFlow concept and NetFPGA reference designs into a network testbed based on Emulab. This provides researcher a traffic-controllable and NIC-programmable experimental networking testbed in intra-communicating part, improving high-speed packet processing and dynamic adjustability for network emulation experiments. So as to provide researchers diverse functions to emulate specific experiment topology and to form virtualized networks for experiments. In addition to new techniques contributed intra-connection in a single network testbed, we are planning to provide researchers the possibility of their experiments with large-scaled topology across different testbeds.

Acknowledgements

This research was financially supported by the National Science Council of Taiwan, under grants No. NSC 99-2219-E-006-001 and NSC 100-2219-E-006-001, for which we are grateful.

References

1. ns2, The Network Simulator. http://nslam.isi.edu/nslam/index.php/User_Information

2. John W. Lockwood; Nick McKeown; Greg Watson; Glen Gibb; Paul Hartke; Jad Naous; Ramanan Raghuraman; Jianying Luo. NetFPGA, An Open Platform for Gigabit-rate Network Switching and Routing. *MSE 2007*, San Diego, June 2007. <http://netfpga.org/>
3. Nick McKeown; Tom Anderson; Hari Balakrishnan; Guru Parulkar; Larry Peterson; Jennifer Rexford; Scott Shenker; Jonathan Turner. OpenFlow: Enabling Innovation in Campus Networks. *ACM 08*, New York, NY, USA, March 14, 2008.
4. Emulab of Utah University. <http://www.emulab.net/>
5. Anderson, D.S.; Hibler, M.; Stoller, L.; Stack, T.; Lepreau, J. Automatic Online Validation of Network Configuration in the Emulab Network Testbed; Autonomic Computing. *ICAC '06. IEEE International Conference*, Dublin, Ireland, June 2006.
6. NetFPGA Projects. <http://netfpga.org/foswiki/bin/view/NetFPGA/OneGig/ProjectTable>
7. J. Naous; D. Erickson; G. A. Covington; G. Appenzeller; N. McKeown. Implementing an OpenFlow switch on the NetFPGA platform. In *Symposium on Architecture for Networking and Communications Systems 2008 (ANCS, 08)*, New York, NY, USA.
8. Major components of NetFPGA platform.
http://netfpga.org/foswiki/bin/view/NetFPGA/OneGig/Guide#Software_Installation
9. CPCI Architecture. Glen Gibb <http://netfpga.org/orig/docs/cpci.pdf>
10. Mike Hibler; Leigh Stoller; Jay Lepreau; Robert Ricci; Chad Barb. Fast, Scalable Disk Imaging with Frisbee. <http://www.cs.utah.edu/flux/papers/frisbee-usenix03/>
11. Nemesis. <http://nemesis.sourceforge.net/>
12. IDSwakeup. <http://www.hsc.fr/ressources/outils/idswakeup/>
13. Taiwan Advanced Research & Education Network. TWAREN <http://www.twaren.net/>
14. Implementation of a Future Internet Testbed on KOREN based on NetFPGA/OpenFlow Switches. http://netfpga.org/DevWorkshop/NetFPGA_DevWorkshop09_Paper_11-KOREN_Testbed.pdf
15. Chi-Sung Lai; Jung-Shian Li; Mao-Jie Lin; ShRiau-Han Chang; Li-Da Chen; Shih-Hsien Tseng; Michael Chang. Development and Operation of Testbed@TWISC. *JWIS 2008*, Seoul, Korea. <http://testbed.ncku.edu.tw/index.php3>
16. Internet2. <http://www.internet2.edu/>

© 2011 by the authors; licensee Asia Pacific Advanced Network. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).