Proceedings of the Asia-Pacific Advanced Network ***2012*** *v. 34, p. 11-16.*
http://dx.doi.org/10.7125/APAN.34.2
ISSN 2227-3026

Connecting NetOpen Nodes for NetOpen Resource Aggregate

Namgon Kim and JongWon Kim
Networked Computing Systems Lab., School of Information and Communications,
Gwangju Institute of Science and Technology (GIST), Gwangju, 500-712, Korea

E-Mails: {ngkim, jongwon}@nm.gist.ac.kr

* Tel.: +82-(0)62-715-2219; Fax: +82-(0)62-715-3155

Abstract: NetOpen RA (Resource Aggregate) is the programmable network substrate containing resources supporting OpenFlow-based programmable networking. A number of experiments can be performed by sharing resources built by connecting NetOpen nodes. As today's Internet does not support layer-2 (L2) data-plane connectivity required for OpenFlow-based programmable networking, tunneling techniques are important in building NetOpen RA with NetOpen nodes. In this paper, we present several tunneling techniques and evaluate the performance of the connections according to tunneling techniques. We also present demonstration experiences with the NetOpen RA.

Keywords: NetOpen; Resource Aggregate; OpenFlow; Networking service; Tunneling.

1. Introduction

With the introduction of Software-Defined Networking (SDN) [1], network innovations are enabled by separating the control plane from the data-path processing. OpenFlow [2], an open source tool for SDN, presents a remotely programmable interface that allows network switches to be controlled from a logically centralized location. It is now easy to set up a programmable network switch with OpenFlow. However, as today's Internet is not compatible with OpenFlow-based programmable networking, it requires tunneling techniques to build programmable network substrates with geographically distributed OpenFlow-enabled nodes.

In this paper, we present how we build NetOpen RA (Resource Aggregate) by connecting NetOpen nodes supporting OpenFlow-based programmable networking. We first present NetOpen node, a container for computing/networking resources with OpenFlow-based programmability. To build NetOpen RA with NetOpen nodes, we present several tunneling techniques to keep layer-2 (L2) data-plane connectivity among NetOpen nodes. We verify the NetOpen RA by evaluating the performance of tunneled connections and presenting our demonstration experience with it.

2. NetOpen Node

We envision the needs for programmable nodes containing various computing/networking (albeit with media processing capabilities) resources. NetOpen nodes are networking-centric milestones towards the envisioned nodes. The NetOpen node serves as a network switch which is programmable (and partially virtualized via flow space) through the OpenFlow protocol [2].

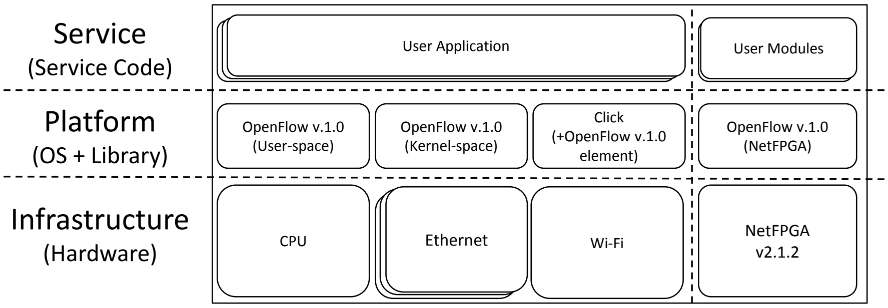

Figure 1. NetOpen node supporting data-plane programmability.

Basically, a NetOpen node is a container for computing/networking resources such as general-purpose CPU and NICs (Network Interface Cards) including Ethernet, Wi-Fi and NetFPGA supporting hardware-accelerated programmable networking. With the OpenFlow-enabled NetOpen node, we can deeply program its networking resources. By linking Click [3] with OpenFlow, a software-based extensible modular router, we can support data-plane programmability and customized processing on each packet via the combination of Click elements.

3. NetOpen RA

NetOpen RA (Resource Aggregate) is the programmable substrate containing computing/networking resources built by connecting a set of NetOpen nodes. There are two requirements for building a NetOpen RA. First, to enable OpenFlow-based network programmability, participating NetOpen nodes should have L2 data-plane connectivity with each other. Next, to be shared among multiple experimenters, a NetOpen RA should support flowspace-level network virtualization.

For keeping L2 data-plane connectivity among NetOpen nodes connected through layer-3 (L3) networks, we selectively use three tunneling techniques. We can use 1) software-based or 2) hardware-accelerated EoIP (Ethernet-over-IP) tunneling solutions to create EoIP tunnel connecting them. Also we can setup 3) VLANs along the path between NetOpen nodes. To support multiple VLANs with a single VLAN, we use 802.1Q tunneling that expands VLAN space by using a VLAN-in-VLAN hierarchy and tagging the tagged packets. The software-based EoIP tunneling solution is the easiest one to use, but we cannot achieve good networking performance. On the other hand, with the hardware-accelerated EoIP tunneling solution, we can get better networking performance. However, both parties should be equipped with the hardware-accelerated EoIP tunneling solution and it could cost much. Although there is an open source implementation on NetFPGA, it has limited port scalability and its stability has not yet been confirmed. Thus configuring VLAN is the best solution in stability and performance perspective. However, it takes time to configure as it needs to be done manually by the network operators managing the networks between NetOpen RA and BEN. For connecting NetOpen nodes, we selectively use one of the three methods according to their network configurations.

To support flowspace-level network virtualization, we use FlowVisor [4], a tool that provides flowspace-level network virtualization in which the same hardware forwarding plane can be shared among multiple experimenters. A flowspace is defined by a subspace of the entire geometric space of possible packet headers. Based on flowspaces allocated to experimenters, FlowVisor performs relay of OpenFlow messages between control nodes of the experimenters and NetOpen nodes.

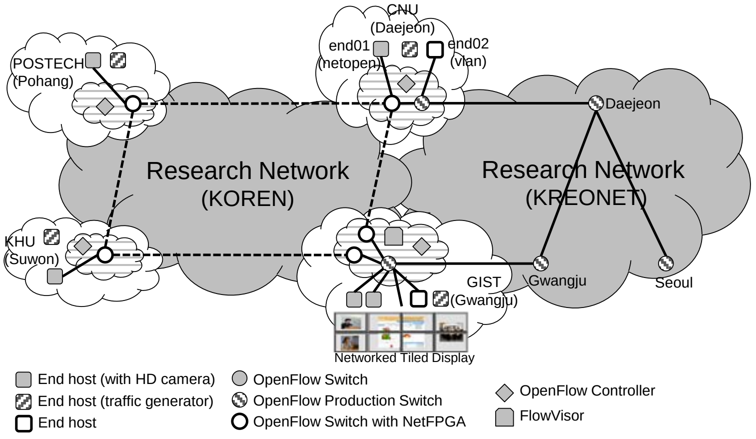

Figure 2. NetOpen RA.

We configure a NetOpen RA by connecting four participating sites (CNU, POSTECH, KHU, and GIST) equipped with NetOpen nodes as depicted in Figure 2. As participating sites basically have L3 network connections via research networks such as KOREN and KREONET, we use NetFPGA-based hardware-accelerated EoIP (Ethernet-over-IP) tunneling solutions (called NetOpen tunneling, hereafter) to enable OpenFlow-based network connectivity among NetOpen nodes. By supporting NetOpen tunneling solutions, we can dynamically configure links (depicted as dotted line) between sites according to networking needs. We also enable L2-VLAN based static connections between GIST and CNU (depicted as solid line) through KREONET. It also supports flowspace-level network virtualization by connecting all the NetOpen nodes in the NetOpen RA with FlowVisor.

4. Verification

4.1. Throughput Result

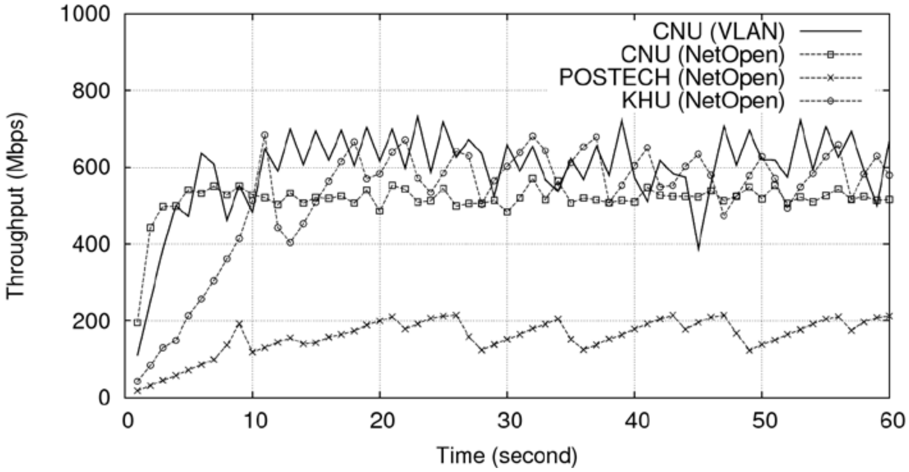

Figure 3. TCP throughput of tunneled connections.

To verify the performance of tunneled connections among NetOpen nodes, we perform TCP throughput measurement of tunneled connections by using a well-known network performance measurement tool, iperf. We measure TCP throughput of the connections by placing an iperf server at GIST and running an iperf client at each institute. Figure 3 shows the throughput measurement result. The throughput result of VLAN and NetOpen tunneling of CNU shows that the throughput of VLAN is higher than NetOpen tunneling. This is due to the encapsulation overhead of NetOpen tunneling (42bytes) is bigger than that of VLAN (4bytes) and NetOpen tunneling does not guarantee lost packets as it uses UDP (User Datagram Protocol) transport. Note that NetOpen tunneling of CNU is using KOREN showing 3ms RTT (round trip time) and VLAN of CNU is connected through KREONET with 3.33ms RTT which affects the time to reach their maximum throughput.

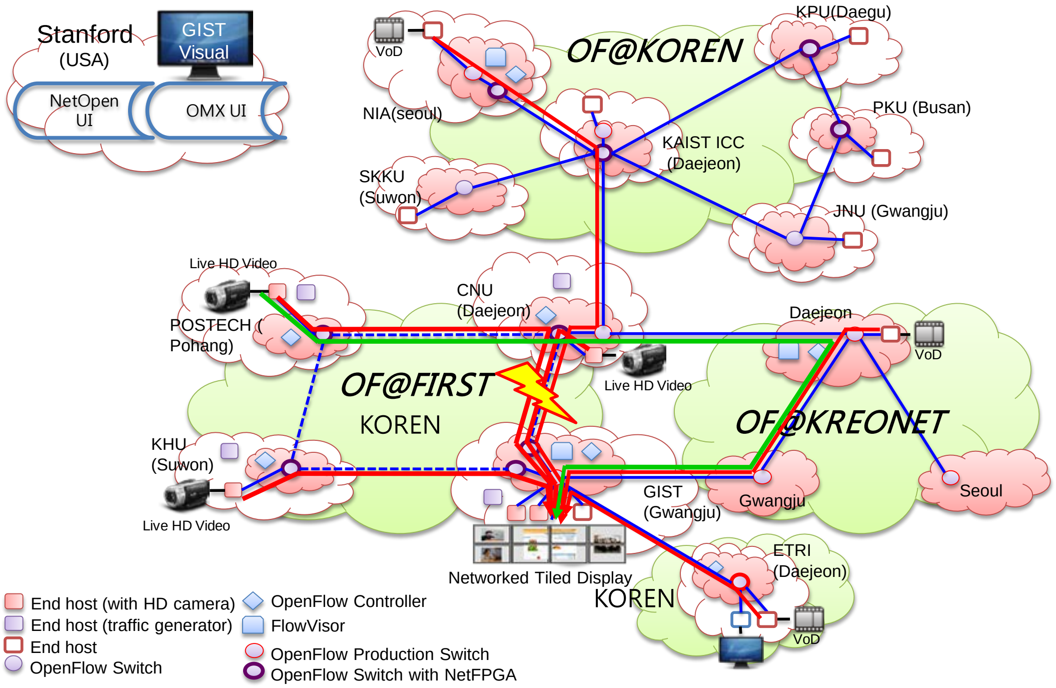

Figure 4. QoS control experiment over OF@Korea.

With the NetOpen RA, we build OF@Korea, an integrated OpenFlow testbed by linking OpenFlow-enabled RAs managed by several research institutes and universities in Korea as depicted in Fig. 4. The goal of OF@Korea is giving an experimental environment that experimenters can run their services through a portal-like interface on the integrated OF@Korea. Three major entities collaborate for this work with their own OpenFlow-enabled network substrates. First, OF@FIRST (Future Internet Research for Sustainable Testbed) RA came out of the FIRST project, jointly carried out by ETRI (Electronics and Telecommunications Research Institute) and a university team (FIRST@PC: GIST/CNU/Postech/KHU). Inside OF@FIRST RA, the university team is working on prototyping a PC-based virtualized computing/networking substrates using NetFPGA/OpenFlow while ETRI is contributing an OpenFlow-enabled NP-accelerated ACTA-based virtualization platform, denoted as the FIRST platform. The NetOpen RA participates as part of OF@FIRST. Next, KREONET (Korea Research Environment Open NETwork) and KOREN (Korea Advanced Research Network) RAs are respectively providing OF@KREONET and OF@KOREN resources together with domestic R&D network connections for this integration.

In this demonstration, we show a QoS (Quality of Service) control experiment that assists the transport of flows on the integrated OpenFlow-enabled resources of OF@Korea, starting from OF@FIRST (denoted as NetOpen RA). In this demo, as shown in Fig. 4, we display multiple videos from all distributed video senders on the Networked Tiled Display (NeTD) at GIST while meeting the QoS requirements of each flow via the proposed combination of NetOpen QoSrelated software. Initially, we start by providing shortest-path connections for all video flows that connect them using minimum-hop paths. Then, by adding disturbance traffic flows, we emulate possible changes in the underlying OpenFlow-enabled network and at the same time we turn on the NetOpen QoS control. If the invoked change breaks the QoS requirements of some flows, the NetOpen QoS control will automatically change the paths of QoS-affected flows to exploit alternative flow connection possibilities.

4. Conclusions

In this paper, we present how we build NetOpen RA by connecting NetOpen nodes. We show three tunneling techniques for connecting NetOpen nodes over L3-based Internet. We verify the NetOpen RA by measuring the throughput of tunneled connections and show our demonstration with the NetOpen RA integrated with other OpenFlow-enabled RAs.

Acknowledgements

This research was one of KOREN projects supported in part by National Information Society Agency (11-951-00-001) and in part supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. 2011-0027558).

References

1. N. McKeown. Keynote talk: Software-defined networking. In *Proc. of IEEE Conference on Computer Communications (INFOCOM'09)*, Apr. 2009.
2. OpenFlow Switch Specification, Version 1.0.0 (Wire Protocol 0x01), http://www.openflow.org/documents/openflow-spec-v1.0.0.pdf, Dec. 2009
3. E. Kohler; R. Morris; B. Chen; J. Jannotti; M. F. Kaashoek. The click modular router. *ACM Transactions on Computer Systems*, 2000; Volume 18, Number 3, pp. 263–297.
4. R. Sherwood; M. Chan; G. Gibb; N. Handigol; T.-Y. Huang; P. Kazemian; M. Kobayashi; D. Underhill; K.-K. Yap; G. Appenzeller; N. McKeown. Carving research slices out of your production networks with OpenFlow. In *Proc. of the ACM SIGCOMM 2009 (Demo)*, Aug. 2009.