# ICN-OMF: A Control, Management Framework for Information-Centric Network Testbed

Lee Hyunwoo, Donghyun Kim, Junho Suh, Ted "Taekyoung" Kwon
Seoul National University
1 Gwanak-ro, Gwanak-gu
Seoul, Korea
{hwlee2014, dhkim, jhsuh, ted}@mmlab.snu.ac.kr

***Abstract*—A publisher/subscriber model dominates today's Internet usage behavior instead of a location-based host access. Along with this stream, Information-Centric Network (ICN) is proposed for Future Internet Architecture to remedy the problems the current Internet is encountered. Although there are lots of research efforts on ICN, but still evaluation and validation of their proposals stay at simulation or emulation. Thus, this requires a scalable, real physical testbed for fast prototyping.**

**In this paper, we develop and deploy ICN-OMF: a Control, Management Framework for a scalable CCN testbed, including controlling and management of multiple CCN nodes dispersed across geographically. To this end, we leverage, extend a Testbed Control, Measurement and Management Framework (OMF) and an overlay. We believe experimenters can utilize this framework to validate and evaluate their ideas in more convenience.**

***Keywords*—network testbed, information-centric network, testbed control and management framework(OMF)**

## I. INTRODUCTION

*Information-Centric Network* (ICN) recently attracts many attentions from researchers as Future Internet Architecture due to its novel communication model, distributing/retrieving the contents by its name (i.e., "*what*") rather than accessing the location the contents resides ("*where*") [1]. [2] reported that the communication behavior of Internet has been shifted to publisher/subscriber model, which is more optimistic for content distribution/retrieval. However, the current IP-based Internet architecture is not designed to accommodate the communication model. With this motivation, ICN is proposed to remedy the problems the Internet encounters (i.e., an inefficient communication model for contents distribution and retrieval).

Although there are lots of proposals investigating on architecture of ICN [3], their methodologies to evaluate and validate ideas still stay at unrealistic simulation [4] or small-scale emulation [5]. However, to become a new protocol deploying at the production networks, it should be validated and evaluated on real physical testbed, providing scalability, configurability, and low-cost to researchers. Thus, if there is a formulaic testbed for ICN, the experimenters can only focus on their own experiments without concerning cumbersome works.

In testbed community, there is a framework to federate heterogeneous physical resources (e.g., switches, servers, sensors, etc.) dispersed across globally, called a Testbed *Control and Management Framework* (OMF) [6]. The objective is to supply the unified form of management, control, and measurement tools and services to experimenters. By this, the experimenter just describes the *Experiment Description* (ED) in *OMF Experiment Description Language* (OEDL) and submits it to *Experiment Controller* (EC). EC then interprets ED and directs *Resource Controller(s)* (RC) to prepare its resources adapted to the experiment. For scalable communication between EC and RC(s), the messaging protocol such as XMPP or AMQP is used. After preparation of resources, the experiment is executed. After finishing the execution, the experimenter can see the results she is specified.

In this paper, we develop and deploy ICN-OMF: a Control, Management Framework for a scalable, configurable, and low-cost ICN testbed. To this end, we leverage and extend a Testbed Control, Measurement and Management Framework (OMF) [6] to control and manage globally dispersed ICN nodes (i.e., publishers, subscribers, or routers). Further, since ICN is a clean-slate network architecture that is not compatible with the current IP-based network, we build ICN network as an overlay network. Note that we use the CCNx open-source software that is the one of candidates for ICN and widely used for ICN community. We believe experimenters can utilize this framework to validate and evaluate their ideas in more convenience.

With OMF, we construct a testbed for testing CCN network with CCNx [7]. By this, the experimenter who is interesting in CCN can build the topology and test. We offer the way of building the random topology with some parameters, such as the number of publishers, routers, and subscribers automatically. Using this testbed, we expect the development of CCN and new network technologies.

This paper is organized as follows. In section II, the entire architecture of the testbed and the scenario of the experiment will be introduced. In section III, the way of building random topology with some modules will be described. And finally, the example using the testbed will be shown.

## II. ICN-OMF

### A. Design Objectives

We build the CCN testbed with the objective as follows.

- Scalability : If the number of resources are limited, it is a constraints to the experiment. So, we use the advantage of the virtualization at most. Using the virtual machine, we offer the nodes as possible as the experimenter wants.
- Configurability : The more configurable the condition is, the more elaborate the experiment is. So, we offer the way to configure attributes as the experiment wants.

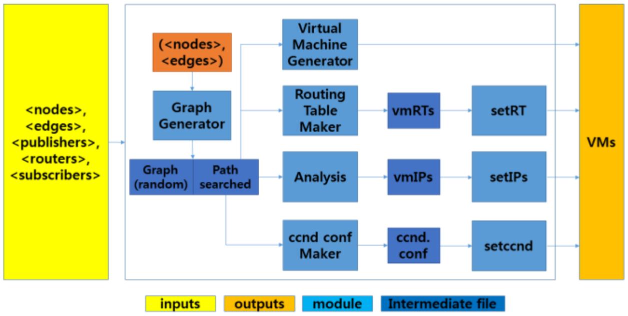


**Figure 1. A Design of ICN-RC.**

- Low-cost : The cost is the important factor in experiments. If the experiment can be made without the consideration of cost, the experiments can be invigorated. Because we use the virtual machines, the experiments can be free from the cost.

We expect scalability, configurability, and low-cost may give flexibilities and delicacy for the experiment.

## B. *Scenario*

The scenario of the test is composed of the following phases. 1) First, the experimenter describe an ED followed by including the network construction script. 2) Second, the experimenter sends the ED to the EC and executes the experiment. Then EC directs RC to prepare the random topology as directed by the experimenter. 3) Finally, the experiment is executed in the constructed network. Currently, we can test *ccnputfile* command in the publisher and *ccngetfile* command in the subscriber offered by CCNx.

## C. *Overall Design*

**Bootstrapping an overlay network**: An experimenter who wants to customize ICN experiment first describes her ED in OEDL by specifying the parameters, including a topology (i.e., #nodes, #edges, #publishers, #subscribers, #routers), a routing protocol (e.g., by default NLSR [8]), and a cache strategy (e.g., LRU or LFU). After then, ED is passed to RCs you want to use, which is also specified in ED. Given the parameters, ICN-RCs installed on physical resources (as shown in Figure 2) instantiates the number of VMs as much as the value in ED. At this point, it initializes VMs using a prebuilt CCNx image maintained at a centralized repository. Thus, there can be lots of versions that experimenters can customize their ICN node.

**Deploying a CCN network on top of the generated overlay network**: *Analysis* module analyses the graph made by *Graph generator* module to assign the IP address to each nodes' network interfaces to build an overlay. *Routing table maker* module takes the entire paths of the graph using Dijkstra algorithm applying from every nodes to every nodes, and makes the IP routing table for each nodes. *ccnd conf maker* also takes the entire paths and make configure file for each nodes. The CCN configure file describes the translation from the name prefix to the next hop to route on each nodes. *setRTs* module assigns the routing table for each nodes, *setIPs* module assigns the IP address on each hosts' adapter, *setccnd* module set the CCN configure file on each nodes.

**Running custom experiments**: After finishing the topology setup, we can now direct the publisher to put the file

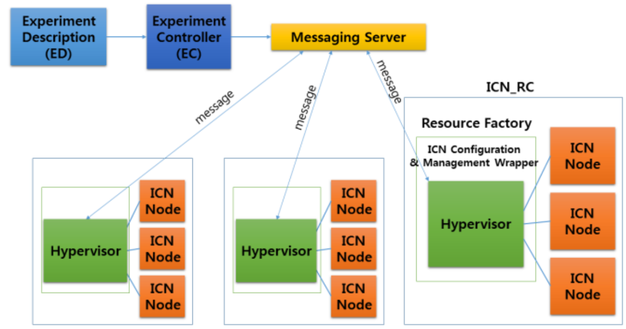


**Figure 2. An Architecture of ICN Testbed.**

in its repository, and the subscriber to get the file using the name prefix. We can find the result of the process on the command line.

# III. CONCLUSION & FUTURE WORK

In this paper, we propose the CCN testbed, as the first approach for developing the CCN network. As we mentioned, because the data traffic is increased and the content distribution is dominated in network, the importance of distributed network is also increased. Using the CCN testbed, we expect the development of the network adapting to this tendency. Furthermore, as OMF aim to federate the networks, CCN testbed can contribute to accelerate the progress of the new network technology with co-operation.

For future work, since generating VMs at large-scale using hypervisor suffers an overhead to instantiate, we are now shifting it to a container-based virtualization using LXC [9] where is a lightweight virtualization to speedup deploying VMs.


# ACKNOWLEDGMENT

This work makes use of results produced by the SmartFIRE project, which is supported by the International Research & Development Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (MSIP, Korea) (Grant number: K2013078191) and the Seventh Framework Programme (FP7) funded by the European Commission (Grant number: 611165).


# REFERENCES


[1] V. Jacobson, Diana K. Smetters, James D. Thornton, Michael F.Plass, Nicholas H.Briggs, Rebecca L. Braynard, Networking Named Content, In *CoNext*, 2009.

[2] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018, White paper, 2014

[3] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B.Ohlman, A Survey of Information-Centric Networking, IEEE Communications Magazine, 2012

[4] ndnSIM, http://www.ndnsim.net

[5] C. M. S. Cabral, C. E. Rothenberg, M. F. Magalhaes, Mini-CCNx: Fast Prototyping for Named Data Networking, ACM SIGCOMM workshop, 2013

[6] T. Rakotoarivelo, M.Ott, G.Jourjon, I.Seskar, OMF: A Control and Management Framework for Networking Testbeds, ACM SIGOPS Operating Systems Review Volume 43 Issue 4, 2010

[7] Priya Mahadevan, CCNx 1.0 Tutorial, 2014

[8] A. K. M. Mahmudul Hoque, S. O. Amin, A. Alyyan, NLSR: Named-data Link State Routing Protocol, SIGCOMM, 2013

[9] LXC – Linux Containers, https://linuxcontainers.org