# Software-Defined Infrastructure and the Future Central Office

Joon-Myung Kang, Hadi Bannazadeh, and Alberto Leon-Garcia
Department of Electrical and Computer Engineering
University of Toronto, Toronto, ON, Canada
Email: {joonmyung.kang, hadi.bannazadeh, alberto.leongarcia}@utoronto.ca

***Abstract*—This paper discusses the role of virtualization and software-defined infrastructure (SDI) in the design of future application platforms, and in particular the Future Central Office (CO). A multi-tier computing cloud is presented in which resources in the Smart Edge of the network play a crucial role in the delivery of low-latency and data-intensive applications. Resources in the Smart Edge are virtualized and managed using cloud computing principles, but these resources are more diverse than in conventional data centers, including programmable hardware, GPUs, etc. We propose an architecture for future application platforms, and we describe the SAVI Testbed (TB) design for the Smart Edge. The design features a novel Software-Defined Infrastructure manager that operates on top of OpenStack and OpenFlow. We conclude with a discussion of the implications of the Smart Edge design on the Future CO.**

## I. Introduction

New content and applications (YouTube, Facebook, Skype, ...) provide value and utility in numerous and varied socioeconomic contexts, and this drives investment for major changes in computing and network infrastructure. We use the term *application platform* to refer to the many elements of infrastructure and software involved in the delivery of applications and content, from users and devices to access and core networks, the Internet, and data centers. In this paper we report on the SAVI[1] project that explores the role of virtualization and software-defined infrastructure in application platforms, and we present the design and implementation of a prototype to support experimentation in application platforms.

We envision a future open marketplace where a vast number of providers and vendors offer applications, content, and services to consumers as well as to other vendors. This marketplace will be characterized by extremely large scale and very high churn, with new applications and content being introduced, modified and retired at very fast rates. This dynamicity will place extreme demands on the supporting application platform infrastructure for agility in resource allocation, as well as scalability, reliability, and security. Cost-effectiveness will require that the infrastructure be flexible, so that it can be readily re-purposed, essentially reprogrammed, to provide new capabilities. Therefore the management and control systems must be designed to provide efficient resource usage and high availability of *converged network and computing resources* at low ongoing operations expense. Multiple infrastructure owners will be involved and so the architecture for the infrastructure must be open and allow for interconnection and federation. And crucially, the architecture of the infrastructure should support the rapid introduction of applications, the delivery of applications with targeted levels of Quality of Experience (QoE), and the rapid retirement of applications and redeployment of their supporting resources.

[1]Canada NSERC Strategic Network for Smart Applications on Virtual Infrastructure (SAVI)

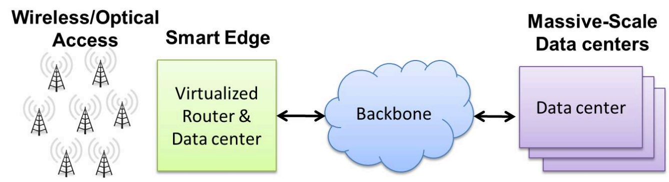


Fig. 1. Conceptual view of multi-tier cloud

Cloud computing and virtualization provide the foundation for the SAVI resource management system for future application platforms. Virtualization simplifies resource management and allows operation over heterogeneous infrastructures. SAVI takes a converged view of infrastructure in that all resources whether computing, processing, or networking are viewed as being part of sharable resource pools that can be controlled and managed using cloud computing principles.

In section II we discuss application platforms as encompassing a multi-tier cloud that includes a "Smart Edge" that could be located where service providers are currently placed. Section III describes the SAVI TB architecture, design, and implementation. We introduce a Software-Defined Infrastructure manager that leverages OpenStack [1] and OpenFlow [2] open source software. Section IV discusses the Smart Edge in the context of the Future CO. Section V presents conclusions.

## II. The Smart Edge in a Multi-Tier Cloud

In a typical future setting, a user will access the applications platform through a mobile device that connects to a very-high-bandwidth, integrated wireless/optical access network as shown in Figure 1. The application platform provides network connectivity to services that support the application of interest. Most services reside in distant data centers at sites of inexpensive or renewable energy. Services that require low latency (e.g. alarms in smart grids, safety applications in transportation, monitoring in remote health, seamless handover for mobile users) or processing large volumes of local information (e.g. instant identification of individuals at security sites) will be provided by converged network and computing resources at the *Smart Edge* of the network, e.g. in the premises of telecom service providers.

As shown in Figure 1, SAVI views application platforms as including a multi-tier cloud with a Smart Edge tier. Existing data centers provide conventional computing resources such as VMs and storage. SAVI investigates the hypothesis that *all* computing and networking resources can be virtualized and managed using Infrastructure-as-a-Service (IaaS). The Smart Edge will need to meet highly demanding applications such as video distribution, and fast and secure communication. As discussed in Section IV, the Smart Edge may also support software radio and other intensive signal processing. Therefore the Smart Edge will be a heterogeneous data center that includes virtualized programmable hardware, GPUs, and storage and specialized hardware accelerators. Furthermore, while current edge nodes provide mostly one-way (core-to-edge) content caching and distribution, the Smart Edge is two-way (core-to-edge and edge-to-core) providing flexibility for load balancing across multiple Smart Edge nodes as well as core nodes.

To design a Smart Edge system, we address two challenges: 1) Managing heterogeneous resources along with conventional resources, ideally using existing open source cloud management software; and 2) Integrating Software Defined Networking (SDN) in the Smart Edge. In the next section we present a design that meets these challenges. The associated control and management system allocates virtual slices of the SAVI TB to create virtual infrastructures dedicated to specific experiments or applications. This enables experimentation in: Future Internet protocols; novel networking, e.g. secure, green and energy-efficient networking; novel wireless signal processing and access protocols, including software-defined radio; real-time media processing; content caching, content routing, publish/subscribe, and other higher-layer protocols and services.

## III. SAVI Testbed Architecture and Design

In this section we describe the design of the SAVI TB based on the architecture presented in [4]. First, we summarize the high-level architecture of the control and management (C&M) plane. Next, we describe a design of the Smart Edge that features a novel manager for Software-Defined Infrastructure.

### A. High-level architecture

The SAVI TB includes Core (Data center) Nodes, Smart Edge Nodes, Access Nodes, SAVI network, and a Control Center for applications and experiments as shown in Figure 2. The testbed has a two-plane architecture: Control and Management (C&M) plane and Application & Experiment (A&E) plane. The C&M plane allocates and secures a slice of virtualized resources to a researcher's application or experiment. These resources may span any combination of Edge nodes, Core Nodes, Access Nodes, and network resources. Researchers run their applications and experiments in the A&E plane using the virtualized resources allocated to them.

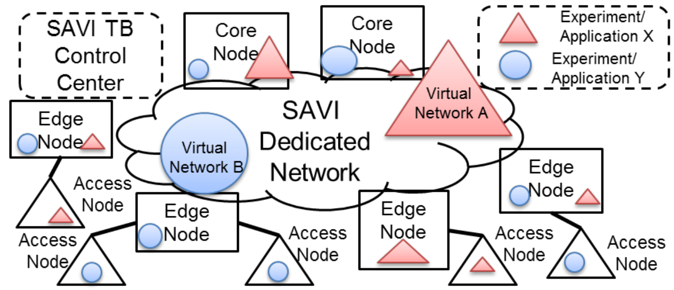


Fig. 2. SAVI testbed main entites

Figure 3 shows the logical components in the C&M plane and their scope of authority in the SAVI TB. The figure shows that the physical resources in each Edge node and each Core node are virtualized by the node's resource virtualization layer. The resources in the Core nodes are conventional cloud computing resources such as computing and storage, but Edge nodes also include reconfigurable hardware and advanced programmable resources. The virtualization layer in Edge nodes also include functionalities for virtualizing the Access Node resources (discussed in Section III-B)

The middle tier in Figure 3 shows that each Core and Edge node has its own C&M component as well as configuration management, and Monitoring and Measurement (M&M) modules that are responsible for conducting node level operations on resources at the node. The C&M at the Edge node also handles C&M tasks required for the Access nodes. The top tier in Figure 3 shows that there are testbed-wide components for resource allocation and management to oversee C&M functionalities for experiments that span multiple nodes. There is also a testbed-wide component responsible for SAVI network virtualization and monitoring and measurement. In terms of physical deployment of these logical components, the node-level components usually reside in their respective nodes. The testbed-wide components can reside on the SAVI TB Control Center, or be distributed on SAVI Core and Edge Nodes. In practice, some testbed-wide components such as Clearinghouse reside in the SAVI TB Control Center while others are distributed to the Edge and Core nodes.

A Clearinghouse is a system (consisting of software, operations, and policies) that brokers trust between the C&M plane and resources. It is the only component that every entity in the SAVI TB trusts fully. The Clearinghouse component makes the various elements in the SAVI TB interoperable. It is also responsible to interface out-of-zone trust and security management systems. The Clearinghouse functionalities are AAA (Authentication, Authorization, and Accounting), TB configuration management, TB resource registry, and Incident Handling (Managing the TB's trusted state).

The SAVI TB provides a secure framework in which all SAVI TB components can operate and interact without any violations. It primarily includes functionalities that are required by all (or many) components such as middleware as well as API and Web portal processing. All SAVI TB components

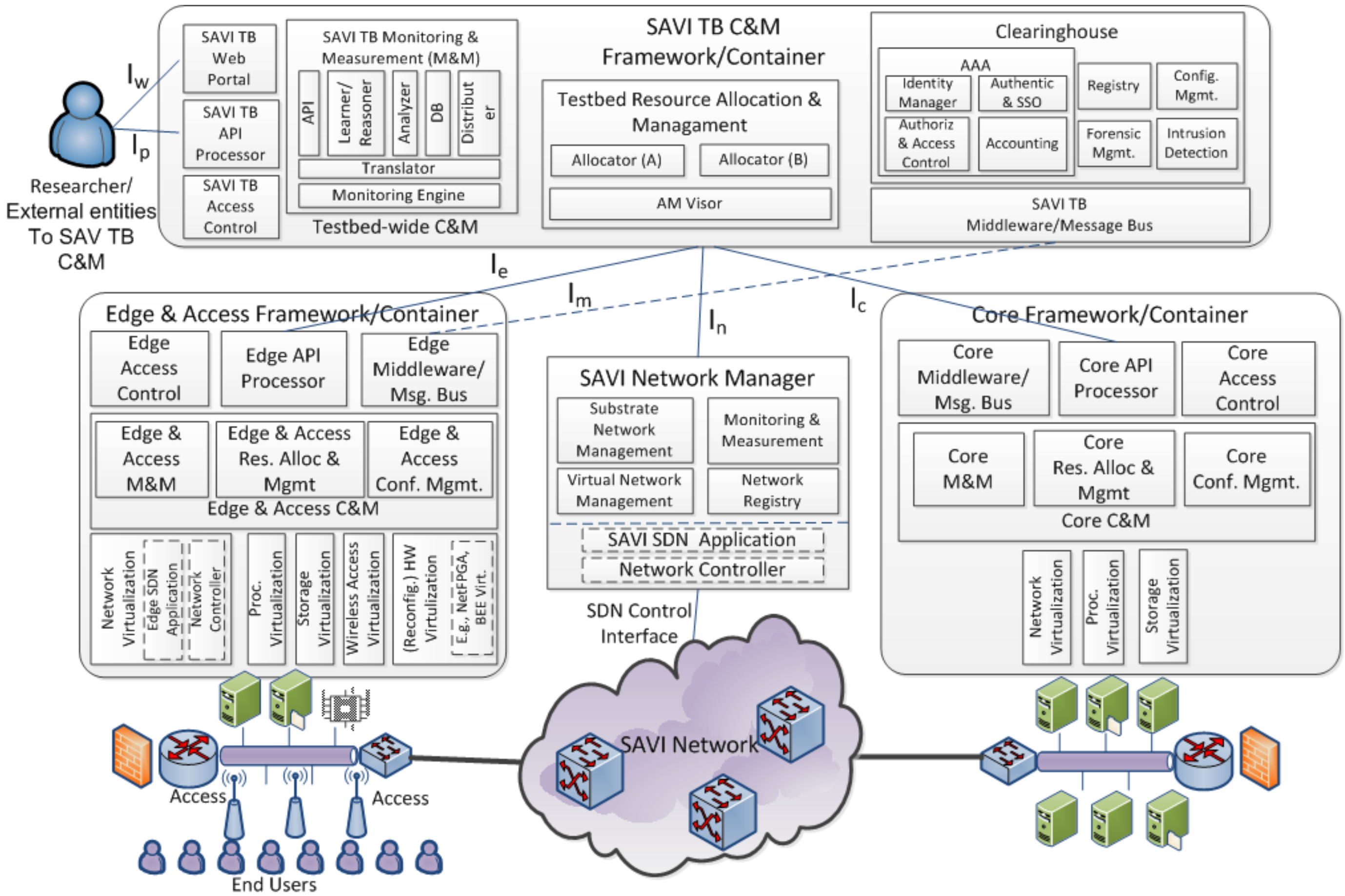


Fig. 3. SAVI testbed reference architecture with C&M plane logical component

based on the framework run independently and can communicate asynchronously. Researchers or other external entities can see a SAVI TB service list, an access control list of each service and utilize services via a web portal or an application. The framework also acts as a distributed container that hosts different SAVI TB components and facilitates their communication through the middleware. This framework should be secured by a strict access control/authentication mechanism, to deny access to SAVI TB components by malicious users, and by an optional encryption mechanism (i.e. Secure Socket Layer (SSL)) to ensure confidentiality of communication with SAVI TB components. The framework should send a warning message to the researcher or other components and stop the experiment or application if there is any problem in the SAVI TB in terms of operations and management. The SAVI TB Web Portal provides a SAVI TB service list, C&M status of SAVI TB, and an access control list. The SAVI C&M API provides general interfaces for accessing SAVI TB C&M. An Access Control Manager manages subscribers and their access control list. The framework provides a message-oriented middleware for asynchronous communication between different components by handling both request-response and notification type messages.

### B. Design of Smart Edge

Figure 4 shows that the design of the Smart Edge includes four majors parts: 1) Edge node network, 2) OpenStack, 3) Openflow controller, and 4) SDI Manager. In the Edge node network, many heterogeneous computing and networking resources are available. The Smart Edge controls and manages virtual resources by virtualizing physical resources using OpenStack [1] which is an open source cloud operating system that controls large pools of compute, storage, and networking resources throughout a data center. In the SAVI TB, we have used and extended the following projects from OpenStack: 1) Keystone for Identity management, 2) Nova for Compute and a cloud computing fabric controller, 3) Swift for Storage, a highly available, distributed, eventually consistent object/blob store, 4) Glance for Image management, 5) Quantum for network management, and 6) Cinder for volume management.

In the SAVI architecture, a Clearinghouse is responsible for AAA service throughout the entire SAVI TB. We have used Keystone for identity management in the SAVI TB. All users and resources can be authenticated and authorized via a Representation State Transfer (REST) [5] API that Keystone has. All images for VMs are managed by a centralized Glance registry. Glance has also a REST API that allows querying of

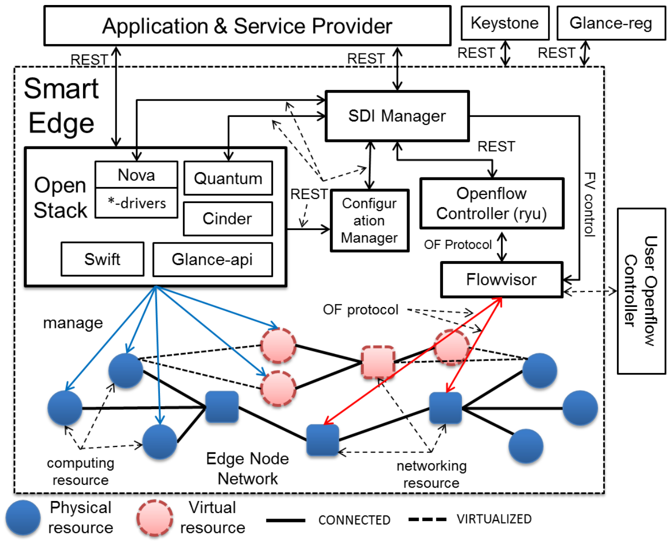


Fig. 4. Design of Smart Edge

VM image metadata as well as retrieval of the actual image.

The Nova in the original OpenStack have not supported virtualization of unconventional resources such as FPGA, NetFPGA or GPU. To meet our needs, we have extended Nova to support virtualization of such resources by adding their device drivers. These are depicted with `*-drivers` in Figure 4. Normally, Nova can provide a virtual resource based on a given flavor (an available hardware configuration for a server). In the original OpenStack, each flavor has a unique combination of disk space, memory capacity, and the number of virtual CPUs. We have added new flavors that specify unconventional resources. As a result, we can obtain both conventional and unconventional resources using the same REST APIs of Nova. For example, suppose that we wish to develop highly parallelizable programs using GPUs in the SAVI TB. As GPUs are considered *many-core* processors, our parallelizable program can obtain substantially greater speed-ups compared to a regular multi-threaded version running on a CPU. Getting a GPU resource simply involves obtaining and booting up a virtual machine as a computing resource server attached to a GPU. As mentioned before, the only difference in obtaining a GPU resource server versus a VM is the flavor type and the image name. There exists a flavor type for GPU's called '*gpu*', as well as a default GPU image named '*UbuntuGPU*' that contains all the proper libraries necessary to interface with the GPU. After getting the GPU, we can develop and test the application on the VM where the GPU connects. Whenever a new resource needs to be introduced, we only need to add a driver and a new flavor for the resource to Nova.

We have used Flowvisor [3] as a controller that acts as a transparent proxy between OpenFlow switches and multiple OpenFlow controllers. Flowvisor creates rich slices of network resources and delegates control of each slice to a different controller. Flowvisor enforces isolation between slices. Internally, we have used the Ryu Openflow controller [6] and integrated it with Quantum in OpenStack through a Quantum plugin. Due to Flowvisor, any user can then use its own OpenFlow controller, even though it is outside the SAVI TB as shown in Figure 4. We have also used an application in the OpenFlow controller to discover topology information using LLDP (Link Layer Discovery Protocol).

As in SDN, we have separated data and control plane in the Smart Edge. The OpenStack and OpenFlow controller are modules for communicating directly with computing and networking resources, whereas the SDI Manager in Figure 4 of the Smart Edge is responsible for C&M tasks. The SDI manager has four major tasks: 1) Configuration management, 2) Monitoring and Measurement, 3) Resource allocation, and 4) Network Management. The topology of a converged cloud network will be dynamic due to various physical and logical activities, such as the addition of new racks, or the addition, deletion, or migration of VMs. The Configuration Manager in Figure 4 periodically gathers all physical and virtual resource information from OpenStack and maintains Smart Edge configuration for support management under dynamic changes. It also gathers M&M data from the resources. The SDI Manager periodically gets information about links between the resources from the OpenFlow controller. The SDI Manager also interacts with Nova for mapping the Nova-API calls to the appropriate physical resource. The SDI Manager selects a computer server from a pool of available resources depending on the allocation algorithm that is in use.

We have built a prototype of a Smart Edge node in our lab. The prototype currently provides FPGAs, NetFPGAs, GPUs, low-power servers and high-end servers all interconnected through a high-capacity OpenFlow switch. Any client can request resources from the SAVI TB using a cloud-API or a CLI interface over SSH. In addition, we are preparing to make access resources available from the Smart Edge in the near future. We are also in the process of connecting multiple Smart Edge nodes in several Ontario universities (University of Waterloo, York University, and University of Toronto) using the Ontario Research and Innovation Optical Network (ORION) [7].

We have completed testing of C&M functionalties in terms of managing resources in the Smart Edge and setting up several mockup systems for experiments. We also organized a one-day hands-on SAVI TB workshop for over fifty SAVI researchers. In the workshop, we demonstrated: How to control and manage resources in the SAVI TB; How to develop cloud-based applications using its computing and storage resources, and then accelerating the applications using FPGAs; How to control virtual networks using OpenFlow, and how to set up and operate the SAVI TB.

## IV. Future CO

In this section we focus on the Future CO as a use case for the SAVI TB that provides the resources to enable packet transport as well as multimedia services and associated management capabilities.

The SAVI Smart Edge consists of resources attached to a high-capacity Openflow switched fabric. The SAVI Smart Edge design is a hybrid system that combines the structure of a computing cluster with that of a high-end router where packet and other higher layer processing is performed on shared resources located on the "other side" of the high-speed fabric. This approach transfers the more demanding header processing to virtual resources that can be allocated on demand, making the resources consumed proportional to the traffic demand. The approach places greater demand on the fabric that is in the heart of the Smart Edge, but current trends in OpenFlow-enabled data center networking demonstrate that low-cost large-scale flexible fabrics are likely attainable.[2] The Smart Edge can provide packet transport services that include IPv6 as well as Future Internet protocols. These future protocols may differ from IP in terms of their naming and addressing conventions, e.g. Content Centric Networking [8]. We currently have projects implementing CCN directly over the SAVI TB, and we are planning a project to implement IPv6 as well. In contrast to current systems where network protocols are rigidly built into the infrastructure, a programmable infrastructure allows multiple protocol stacks to be supported simultaneously in different resource slices.

The Service Layer in Next Generation Networks provides enhanced multimedia services. IMS is an architecture that enables the combination of voice, data, video and mobility into rich communication services. While IMS-based services are currently deployed on dedicated conventional computing infrastructure, with appropriate measures to provide reliability and availability, these services can be transferred to a cloud environment such as that anticipated by the SAVI TB.

A crucial challenge in NGN is to deliver a multiplicity of services that meet target QoE levels. Addressing this challenge will require the real-time monitoring of service-specific QoE's and the correlation with the underlying network QoS conditions. Our experimentation with OpenFlow suggests that passive network monitoring at a fairly large-scale may be feasible for such a QoE/QoS monitoring. This real-time state information can be used to generate alarms to initiate corrective actions when QoE levels are falling; at some point in the future, these adjustments may be implemented autonomically. The SAVI Network has a research team investigating adaptive resource management in this context. The team's scope includes the management of resources to support applications that span multiple tiers in the cloud.

Another major challenge in future networks will be to continue provide increasingly faster wireless access for mobile users. We believe that in many urban settings, the future access infrastructure will be based on ubiquitous, dense-small-cell, very high bit-rate, reliable heterogeneous wireless access integrated with a pervasive optical backhaul network. These access services will support not only mobile users, but also machine-to-machine, sensor and environmental networks. With this scenario in mind, we are exploring how the SAVI Smart Edge can be leveraged to provide virtualized resources for the required novel high-speed, energy-efficient wireless signal processing, and reconfigurable networking techniques. The highly demanding processing required for these techniques provides a strong motivation for including unconventional resources, such as programmable hardware and GPUs, in the SAVI Smart Edge. A SAVI research team is investigating an integrated radio over fiber approach that leverages the programmable resources in the Smart Edge to make the access network virtualizable and programmable.

[2] Achieving strong isolation in the fabric among different slices still requires attention.

## V. Conclusion

This paper has presented a novel approach to controlling and managing a converged computing and networking infrastructure in a multi-tier cloud to support future applications. The infrastructure uses cloud computing principles in its control and management but offers heterogeneous resources such as FPGA, NetFPGA, GPU, and so on. The infrastructure encompasses a multi-tier cloud that includes a Smart Edge. First, we have introduced virtualization and the role of SDI in the design of future application platforms. Next, We have described a design and presented implementation of a Smart Edge prototype that features a novel SDI manager that operates on top of OpenStack and OpenFlow for controlling and managing both computing and networking resources. In the context of future telecom networks, we have also presented the Future CO as a use case of the Smart Edge which provides flexible and programmable packet transport, multimedia service support, and faster wireless access support, while affording efficiency and cost-effectiveness of cloud computing.

## Acknowledgment

This work was funded by NSERC under the Strategic Networks program.

## References

[1] "Openstack community," http://www.openstack.org, [Online; accessed 23-January-2013].

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[3] R. Sherwood, G. Gibb, K. Yap, G. Appenzeller, N. McKeown, , and G. Parulkar, "Flowvisor: A network virtualization layer," Deutsche Telekom Inc. R&D Lab and Stanford University and Nicira Networks, Tech. Rep., 2009.

[4] J.-M. Kang, H. Bannazadeh, and A. Leon-Garcia, "Savi testbed: Control and management of converged virtual ict resources," in *Proc. of the 13th IFIP/IEEE Symposium on Integrated Network and Service Management (IM 2013)*, 2013, (Accepted to appear).

[5] R. T. Fielding, "REST: architectural styles and the design of network-based software architectures," Doctoral dissertation, University of California, Irvine, 2000. [Online]. Available: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

[6] K. Morita and I. Yamahata, "Ryu: Network Operating System," in *OpenStack Design Summit & Conference*, April 16-20 2012.

[7] "Ontario Research and Innovation Optical Network (ORION)," http://www.orion.on.ca/, [Online; accessed 23-January-2013].

[8] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. Briggs, and R. Braynard, "Networking named content," *Commun. ACM*, vol. 55, no. 1, pp. 117–124, 2012.