Experimenting Content-Centric Networks in the Future Internet Testbed Environment

Pedro Henrique V. Guimarães, Lyno Henrique G. Ferraz, João Vitor Torres,
Diogo M. F. Mattos, Andrés F. Murillo P., Martin E. Andreoni L.,
Igor D. Alvarenga, Claudia S. C. Rodrigues, Otto Carlos M. B. Duarte
Grupo de Teleinformática e Automação
Universidade Federal do Rio de Janeiro (COPPE/UFRJ)
Rio de Janeiro – RJ – Brasil
Email:{guimaraes,lyno,jvitor,menezes,afmurillo,martin,alvarenga,susie,otto}@gta.ufrj.br

***Abstract*—Future Internet Testbed with Security (FITS) is a testbed for experimenting Next-Generation Internet proposals. FITS provides two virtualization schemes, one based on Xen and the other on OpenFlow. Experimenting new protocol proposals for the Future Internet requires an experimental environment able to provide realistic conditions for packet forwarding. FITS nodes are spread in Brazilian and European universities. In this paper, we present the FITS and we use it to test a Content Centric Network (CCN), which is one of the main proposals for the Future Internet. The experiment creates a virtual network on the testbed with CCNx stack and measure the file transfer performance under real Internet traffic conditions. The results show that CCN presents an overhead of 19%. Nevertheless CCN outperforms TCP as the number of consumers increases and CCN download time is approximately 25% smaller than TCP on the Internet.**

I. INTRODUCTION

The rapid Internet growth and access popularization has completely transformed the Internet. The original client-server communication model becomes a multimedia content distribution network and, as consequence, a new Internet is required. Future Internet architecture proposals and protocol stack experimentation demands real-scale and real-traffic testing. Network virtualization paradigm is a solution to this issue, because different virtual routers share a physical router in order to simultaneously provide different network services [1]. Therefore, current TCP/IP Internet production traffic can be shared with other experimental networks. Key aspects of this network virtualization are isolation and performance on packet forwarding. Isolation ensures independent virtual network operation, preventing malicious or fault virtual routers interference in the operation of other virtual networks. Privacy is also an important issue and a virtual network can not eavesdrop another virtual network traffic.

We developed the Future Internet Testbed with Security (FITS) [2], an experimentation environment based on virtual networks that offers network isolation, secure access, and quality of service differentiation. FITS nodes are spread over Brazilian and European universities. This virtual network environment allows performance tests and comparisons between Future Internet proposals by virtualizing routers with Xen [3] and managing data flows with OpenFlow [4].

Concerning Future Internet proposals, the content distribution model considers content itself as fundamental resource to share, therefore the network main service is the distribution of content instead of host-to-host communication. The Content-Centric Networking (CCN) [5], also known as Named Data Networking (NDN), is a network architecture that intrinsically supports content distribution with efficiency. CCN forwards packets based on packet content names. This architecture splits content identification from its location, providing support to equal content name requests aggregation, copying and caching responses, balancing request-response pair flow on a hop-by-hop approach, flow multipath, signing, and ciphering content independently of its source and destination host. This paper presents Future Internet Testbed with Security (FITS) and an evaluation of Content-Centric Networks on it. Virtual routers run CCN protocol stack, based on CCNx [6] and OSPFN [7] software packages. FITS cooperating universities connects to each other through the Internet.

The rest of this paper is organized as follows: the interuniversity testbed FITS is discussed in Section II. The Content-Centric Network paradigm and its main components are presented in Section III. The experiments with CCNx's implementation and its results are described at Section IV. Section V contains a final discussion and concludes this paper.

II. FITS EXPERIMENTATION PLATFORM

Future Internet Testbed with Security (FITS)[1] is the platform used in this paper to experiment the Content-Centric Network. FITS platform presents a virtual environment for experimenting new Future Internet network proposals. It is a flexible, open and shared platform for innovative proposals.

FITS allows the creation of multiple virtual networks in parallel, based on virtualization tools Xen and OpenFlow. The testing environment is geographically distributed, with the collaboration of Brazilian and European institutions. Those institutions participate with physical machines that act as nodes of this environment. The experimenting platform follows the pluralist approach that divides physical network in virtual networks, each containing its own protocol stack, routing rules and management. Therefore, FITS allows the creation of many isolated virtual networks, working over the same infrastructure for experimentation. The access control for virtual network management and creation uses a secure platform, based on OpenID [8] and secure microcontrollers. FITS also offers quality of service differentiation and virtual network migration features [9]

FITS implement virtual networks using virtual machines acting as routers and/or flow virtualization. Each virtual net-

[1]http://www.gta.ufrj.br/fits

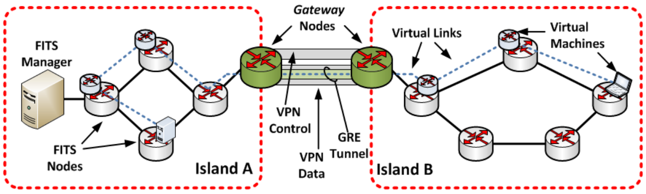


Figure 1. Connection between FITS islands. VPNs interconnect islands, routing control messages. GRE tunnels, inside a VPN dedicated for data, create a unique Ethernet diffusion domain and a unique Link Layer between physical nodes.

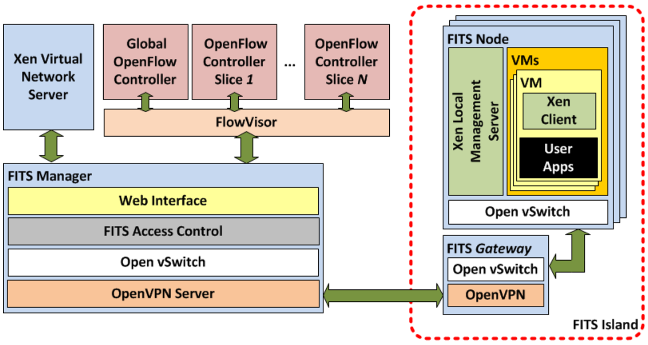


Figure 2. FITS experimentation platform. The FITS Manager controls FITS Nodes and virtual networks through secure connections between the Gateway Nodes.

work runs a different protocol stack and physical machines host instances of these virtual networks. The FITS platform offers a web interface for management, flexible network virtualization through flow virtualization with OpenFlow [10] and machine virtualization with Xen [11]. Besides, the platform offers innovative functionalities, for example virtual network migration.

A. *FITS Platform's Architecture*

The experimentation platform allows the participating institutions to install islands. Each island containing its own policies for experimentation. FITS nodes act as physical substratum for virtual network formation, in which islands are connected to each other through Generic Routing Encapsulation (GRE) tunnels and Virtual Private Networks (VPNs) to emulate virtual layer 2 links on the Internet.

There are three different types of physical nodes in FITS: FITS Manager, Connection Gateways and FITS Nodes (physical machines in the FITS network). Figure 1 shows the network infrastructure and the role of each FITS node. The connection Gateway is a special node, which provides islands' interconnection by creating communication tunnels with other islands' Gateways. The FITS Manager coordinates the platform's operations with user and node authentication, creation of resources and virtual networks and measurement from the islands, all these controls are presented through a web interface.

Figure 2 shows FITS services. FITS Gateway Nodes provide communication between the FITS Nodes and Xen Virtual Network Server and Global OpenFlow Controller, that are FITS Manager services. Xen Virtual Network Server creates and manages virtual networks and connects to each Xen Local Management Server, installed in FITS Nodes to send commands and usage information of physical resources. Xen Local Management Server connects with each Xen Client, inside the virtual machines, to retrieve information about virtual resources usage. The Global OpenFlow Controller has a global view of the OpenFlow network and connects to the FlowVisor [12]. The FlowVisor acts as an interface between OpenFlow switches and OpenFlow network controllers and is the responsible for dividing OpenFlow network into shares.

The FITS Node network operation is based on an Open vSwitch [13], a switch implemented on software that supports OpenFlow control API and has good overall routing performance. The Open vSwitch on FITS Nodes are connected to the OpenFlow Global Controllers, that manages the platform network and virtual links in each virtual network. The OpenFlow Global Controller is the responsible for routing and virtual network isolation.

III. Content-Centric Network

The Content-Centric Network paradigm is an alternative for the current Internet Protocol (IP), which unbinds location and content addressing. In this new paradigm, communication is centered in content instead of host's location. The communication is done with two types of packets: Interest and Content. Interest packets notify requests for a particular content in the network. Content packet is sent in response to an Interest packet, transmitting the desired content or part of it (chunks, as it is called in CCN). The content may be sent by its original producer or the nearest repository that contains this content, for example, a CCN router. This approach enhances mobility and multiple sources and multiple destinations communication. CCN connections work in a link-by-link basis instead of end-to-end, following Delay-Tolerant Network (DTN) concept.

A. *Content-Centric Network Forwarding*

The CCN model is based on Interest and Content packet parity, where a consumer sends Interest packet whenever he desires information. Network forwards Interest to producers that reply with Content packets, traveling the reverse path of its Interest. The CCN router has three main data structures for packet forwarding: Content Store (CS), Pending Interest Table (PIT) and Forwarding Information Base (FIB). The CS repository stores temporarily the searched content, allowing local response to repeated requests. The router replaces stored contents in CS for more relevant items using strategies like Least-Recently-Used (LRU) or Least-Frequently-Used (LFU). The FIB table is similar to IP routers' FIB, storing prefix-based rules, a map of the output interfaces and more specific name prefix patterns. However, CCN allows FIB to map a prefix pattern to a list of output interfaces ordered in priority. The PIT table stores Interest packets, input and output interface by which the packet was forwarded but not replied [14].

The CCN forwarding also comprises an adaptive procedure to choose the best content forwarding path. This procedure has several mechanisms, for example: i) Interest packet time calculation between dispatch and arrival, ordering interfaces with the best metrics, ii) congestion control limiting the number of simultaneous pending Interests, iii) traffic limitation [15].

IV. Experimental Results

We implemented a CCN network as a virtual network in FITS. Using the virtualized CCN network, we measured file download time and analyzed the causes of CCN download delay.

The CCN prototype was implemented on a virtual network in FITS (Future Internet Testbed with Security). The CCN stack was implemented using CCNx [6], version 0.7.1rc1, with OSPFN (Open Shortest Path First for Named Data Networking) [7], version 2.0. The OSPFN extends routing calculation for data names, since OSPFN disseminates data name adjacencies and IPs, as done by OSPF. The OSPFN executes over IP and calculates CCNx network prefix routes in a distributed way based on name adjacencies.

The experiment network proposed in [5] is extended in this paper to the Internet using a virtual network containing CCNx stack. The tests allow to compare the performance of TCP and CCN when multiple consumers are downloading the same content across the Internet, at the same time from the same content producer. For the TCP/IP stack, the file download uses *wget*[2] from a HTTP server.

The content producer (or HTTP server in the TCP/IP network) and consumers (or clients) are connected through a single virtual router. There are 12 virtual machines acting as consumers and downloading the same 6 MB file from one content producer. The consumers and the content producer are connected by one virtual router. The virtual network runs CCN and TCP/IP stacks. The experiment is divided into two different scenarios: first, all 14 virtual machines run inside the same physical node. This scenario was constructed to simulate conditions like those seen in [5]. The second scenario, composed by two physical nodes, with islands across the Internet, will run the virtual machines. In this scenario, the content producer and the router are connected by a virtual link through the Internet. Therefore, bandwidth limitations between content producer and router exist because of real conditions delay existing between the two physical nodes. One physical node hosts the virtual machine acting as the content producer, while the second one contains the router and the consumers. Network's scheme is presented in Figure 3.

The test is run by the control node that executes a measurement script, responsible for starting the tests. The script was written in Python and prepares the file to be downloaded using CCN, through CCNx implementation by using *ccnputfile* command. As soon as the CCN experimentation is done, the test is repeated using *wget*.

In the first scenario, all the 14 virtual machines run in the same physical node and bandwidth between the virtual router and the content producer is limited by the *tc* command. In the second scenario, two physical machines are chosen in FITS and they are placed on different islands and the connection limitation is due to traffic conditions on the Internet that degrades the bandwidth.

The result analysis, as seen in Figure 4, considers the download time of a 6 MB file according to the number of consumers downloading the file at the same time. Results seen in Figure 4(a) shows that when all virtual machines are in the same physical machine and the content producer's bandwidth is limited, the results are as expected and CCN is faster than TCP. CCN outperforms TCP after two consumers downloading the file at the same time. The test scenario is executed over a network with consumers connected to a router, each one using a rapid connection, while the router is connected to a content producer on another island, using a link that has its bandwidth limited by Internet's traffic. When the number of consumers increases, Internet traffic conditions limits total bandwidth and acts as the *tc* command did on the first scenario. However, bandwidth limitations vary drastically during the day. The TCP connection is limited by the bandwidth of the virtual link between the content producer and the router. Whereas on CCN, downloading the file means downloading the file to router cache and distributing this file between the consumers. This allows bandwidth usage optimization in the available connections and accelerating the process.

[2] http://www.gnu.org/software/wget/.

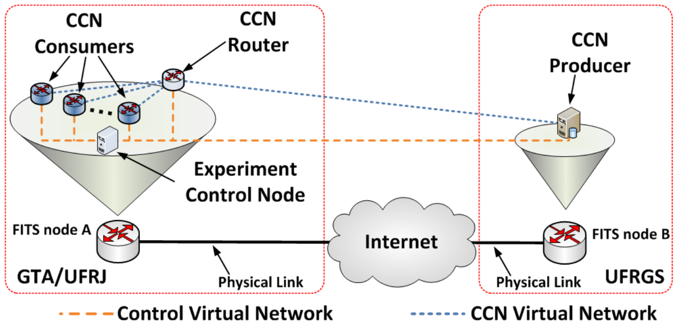
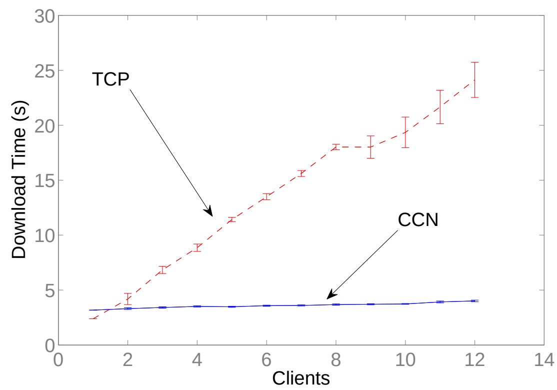
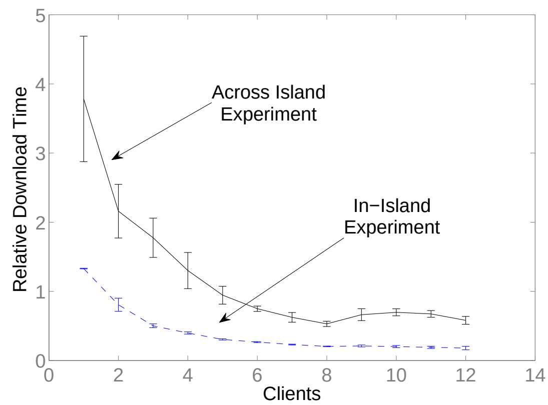


Figure 3. Virtual networks used on the experiments. Two virtual networks were created, the experimentation network and the control network. The control network is an isolated network that sends execution commands to the experimentation network's elements. The experiments use the communication provided by experimentation network.



(a) In-island Scenario.



(b) Comparison between In-island and Across-islands Scenarios.

Figure 4. Comparison of Scenarios.

Figure 4(a) presents results of download time when all virtual machines of the experiment are inside the same physical node. The virtual link between the content producer and the virtual router is limited to 10 Mbps. As expected, results are close to the one seen in [5]. This first scenario shows the proposed platform, FITS, as a valid environment for testing the CCN proposal. As the number of consumers increases, the bandwidth limitation degrades TCP performance. This happens because, in TCP, each download in TCP will share the available bandwidth with the others. CCN, however, is more robust to bandwidth limitations. CCN uses the available bandwidth more efficiently because the router forwards only one request per content demand, but replies the content received to all interfaces that it received requests. TCP routers forwards all requests for certain IP and the HTTP server replies to each request while CCN routers forwards only one Interest for certain content, no matter how many Interests arrive after and the data producer replies with only one Content packet.

Figure 4(a) compares results seen in the first testing scenario (in-island test), Figure 4(b) and the CCN/TCP download time ratio for in-island and across-island scenarios. The main conclusion is that with the Internet traffic conditions, CCN follow the same pattern seen in the first scenario, Figure 4(a), and CCN stack outperforms TCP/IP as the number of consumers increases.

The second experiment consists of capturing transmitted packets between consumers and content producer in the TCP/IP and CCN. This experiment is important for the comprehension of the CCNx's protocol stack implementation and the results of the first experiment. The experiment was done by capturing packets from a consumer while it consumes (or downloads) data of a content (file) in CCN or TCP/IP. For packet capture the tcpdump[3] was used. During the experiment, a 20 MB file was downloaded. In this way, the mean transfer rate was evaluated for each scenario. The mean transfer rate in the CCN scenario was 1,7 MB/s while for the TCP/IP was about 11,8 MB/s. This difference in transfer rate is the reflex of the implementation of the two stacks. TCP is implemented in an optimized way, directly in the kernel, while CCN stack, in special the CCNx implementation is a user space application written in Java [5]. The implementation of the CCNx router, upon arrival of an Interest, passes through to the next link in the network. However, at a data arrival, the first is stored in the local cache of the router and, after, is resent to the consumer that requested it. Another important fact to understand the CCNx's smaller transfer rate is the CCNx 0.7.0rc1 implementation uses a default chunk size of 4 KB. This default size is defined in the *ccnputfile*, the application responsible for content distribution. The size of the chunk is important, because it is the minimum data transfer unit of the CCN. In the experiment we can see that CCNx is based on UDP transport protocol, therefore, there is no connection between consumers and the content producer, so the chunks are sent as UDP datagrams. Another point is that each chunk is mapped in a UDP datagram, so UDP datagrams have 4 KB of content. As the *Maximum Transmission Unit* (MTU) of the virtual network is 1500B, all transmitted UDP datagrams suffer fragmentation, generating a higher delay for datagram reconstruction before handling it to the application.

Figure 5 highlights the file transfer behavior in CCN and in TCP/IP according to time. As mentioned before, TCP/IP

[3]http://www.tcpdump.org/.

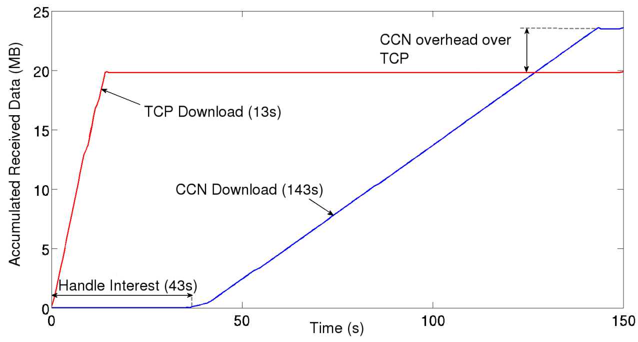


Figure 5. File download duration using a TCP application, compared with duration for downloading the same file using the CCNx stack. Original file's size is 20 MB.

presents a higher file transfer rate, ending the transfer in 13 s. However CCN download took 143 s. The greater amount of time spent by CCN is due to smaller transfer rate of the CCNx implementation. Another point in Figure 5 is the CCN presents an initial delay of 43 s. This delay is due to the fact that the downloaded file is not initially inside the nearest router's cache and, so, it is necessary that the router treats the incoming Interest of the consumers for each chunk and resends it to the content producer that has the content. After the time needed to download the file to the router, CCN's file transfer rate is smaller than TCP/IP's, since CCN demands the fragmentation of 4 KB datagrams, does the data storage in the intermediary router before resending to the consumers and counts upon a user space implementation, while TCP/IP forwarding is optimized directly by the kernel. At the end, Figure 5 still highlights that the CCNx implementation introduces a header overload of 19% in comparison to the same content in the HTTP application over TCP/IP stack.

V. Conclusion

This paper presented FITS, a testbed environment for Future Internet protocols, and experimental results of a Content-Centric Network. FITS is a collaboration of universities to test Future Internet proposals. This environment allows the creation of isolated virtual networks with secure access, quality of service differentiation and virtual network migration features.

Content-Centric Network (CCN) is pointed out as one of the most viable Future Internet proposals. Nowadays, the protocol stack implementation for Content-Centric Network is the CCNx. The presented experiments compare the CCNx with TCP/IP protocol stacks on a virtual network.

The results show that CCNx, when compared with TCP/IP stack, presents an overhead of 19%. Nevertheless CCN outperforms TCP as the number of consumers increases and CCN download time is approximately 25% smaller than TCP when working with 12 consumers downloading content from another FITS island across the Internet. As future works we will study and develop new routing mechanisms for CCN and experiment new Future Internet proposals on the FITS.

Acknowledgments

The authors would like to thank FINEP, FUNTEL, CNPq, CAPES, FAPERJ, and UOL for their financial support. The authors would also like to thank the institutions that collaborate with FITS test platform. A special acknowledgment to Professors Luciano Gaspary and Marinho Barcellos and researchers Éderson Vieira and Lucas Muller of UFRGS; Prof. Igor Moraes of UFF; Prof. Marcelo Rubistein and researcher Leopoldo Mauricio of UERJ; Profs. Edmundo Madeira and Nelson Fonseca and researcher Estéban Rodriguez of UNICAMP; Prof. Artur Ziviani of LNCC; Prof. André dos Santos and researcher Edgar Tarton of UECE; Prof. Djamel Sadok and researcher Marcelo Santos of UFPE; Prof. Cesar Marcondes of UFSCar; Prof. Paulo Veríssimo and researchers Oleksandr Malichevskyy and Diego Kreutz of Universidade de Lisboa, which were essential for the installation and subsequent experiments was possible. Authors would like to thank GTA/UFRJ's team that, in the last three years, worked on virtualization and made the experimentations possible.

VI. Bibliography

[1] N. Fernandes, M. Moreira, I. Moraes, L. Ferraz, R. Couto, H. Carvalho, M. Campista, L. Costa, and O. Duarte, "Virtual networks: isolation, performance, and trends," *Annals of Telecommunications*, vol. 66, pp. 339–355, 2011.

[2] FITS, "Future Internet Testbed with Security, (2012)." 2012. [Online]. Available: http://www.gta.ufrj.br/fits/

[3] P. S. Pisa, R. S. Couto, H. E. T. Carvalho, D. J. S. Neto, N. C. Fernandes, M. E. M. Campista, L. H. M. K. Costa, O. C. M. B. Duarte, and G. Pujolle, "VNEXT: Virtual NEtwork management for Xen-based Testbeds," in *2011 International Conference on the Network of the Future (NoF'11)*, Paris, France, Nov. 2011, pp. 41–45.

[4] D. Mattos, N. Fernandes, V. da Costa, L. Cardoso, M. Campista, L. Costa, and O. Duarte, "OMNI: OpenFlow MaNagement Infrastructure," in *Network of the Future (NOF), 2011 International Conference on the*, nov. 2011, pp. 52 –56.

[5] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 2009, pp. 1–12.

[6] CCNx, "CCNx Project, (2011)." 2011. [Online]. Available: http://www.ccnx.org/

[7] L. Wang, A. Hoque, C. Yi, A. Alyyan, and B. Zhang, "OSPFN: An OSPF Based Routing Protocol for Named Data Networking," University of Memphis and University of Arizona, Tech. Rep., Jul. 2012.

[8] D. Recordon and D. Reed, "OpenID 2.0: A platform for user-centric identity management," in *ACM Workshop on Digital Identity Management (DIM)*, 2006, pp. 11–16.

[9] P. S. Pisa, N. C. Fernandes, H. E. T. Carvalho, M. D. D. Moreira, M. E. M. Campista, L. H. M. K. Costa, and O. C. M. B. Duarte, "Openflow and xen-based virtual network migration," in *Communications: Wireless in Developing Countries and Networks of the Future*. Springer Boston, 2010, vol. 327, pp. 170–181.

[10] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S., and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, Apr. 2008.

[11] N. Egi, A. Greenhalgh, M. Handley, M. Hoerdt, F. Huici, and L. Mathy, "Towards high performance virtual routers on commodity hardware," in *Proceedings of the 2008 ACM CoNEXT Conference*. ACM, 2008, p. 20.

[12] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "Can the production network be the testbed?" in *USENIX Conference on Operating Systems Design and Implementation (OSDI)*, 2010, pp. 1–6.

[13] B. Pfaff, J. Pettit, T. Koponen, K. Amidon, M. Casado, and S. Shenker, "Extending networking into the virtualization layer," in *ACM Workshop on Hot Topics in Networks*, Oct. 2009.

[14] L. Z. *et al.*, "Named Data Networking (NDN) Project," Tech. Rep., Oct. 2010.

[15] C. Yia, A. Afanasyevb, I. Moiseenkob, L. Wangc, B. Zhanga, and L. Zhangb, "A Case for Stateful Forwarding Plane," University of Arizona, University of California, Los Angeles and University of Memphis, Tech. Rep., 2012.