SD-ICN: An Interoperable Deployment Framework for Software-Defined Information-Centric Networks

Jinfan Wang[1], Wei Gao[1], Yuqing Liang[1], Rui Qin[1], Jianping Wang[1], and Shucheng Liu[2]
[1] Department of Computer Science, City University of Hong Kong
[2] Huawei Technologies

***Abstract*—In the past couple of years, various information-centric networking (ICN) architectures have been proposed to address the existing problems of the current Internet, each of which from a different perspective. Thus, it becomes critical to deploy and interoperate different ICNs on top of the same physical network infrastructure. This demo presents SD-ICN, a software-defined interoperable deployment framework based on Open vSwitch (OVS) for ICN architectures.**

I. Introduction

Information-centric networking (ICN) is an emerging communication paradigm which aims to provide efficient content retrieval according to the content name instead of the specified location (*e.g.,* IP address) of a content object. Based on this information-centric concept, various ICN architectures have been proposed, including NDN [1] and PURSUIT [2], which bring many benefits such as shorter content delivery time and inherent content integrity. However, to enjoy these benefits from ICN, significant upgrades to the entire physical network infrastructure are needed to make the network routers support ICN operations, including name-based routing and content caching [3].

The demand of deploying ICN architectures at less pain and the desire of enjoying the benefits of various ICN architectures call for a unified deployment framework with the following features: (1) enabling rapid deployment of an ICN architecture on top of an existing physical network infrastructure; (2) supporting different ICN architectures over the same physical network; (3) enabling the interoperation among different ICN architectures for better content sharing and mobility support. However, designing such a unified interoperable deployment framework is challenging as each ICN architecture has its own features in terms of packet format, request and data routing, service mode and so on. There is no doubt that ICN deployment and interoperability will be one of the important research directions in the ICN research community.

Software-defined networking (SDN), which separates the control plane from the data plane, provides a potential method for developing a unified interoperable deployment framework for ICNs with the aforementioned features. With SDN, the control plane can programmatically make complex logic decisions, *e.g.,* name based routing and interoperability decisions, while the hardware-concerned data plane only needs to execute some simple operations, *e.g.,* forwarding packets and caching content objects, according to the decisions made by the control plane.

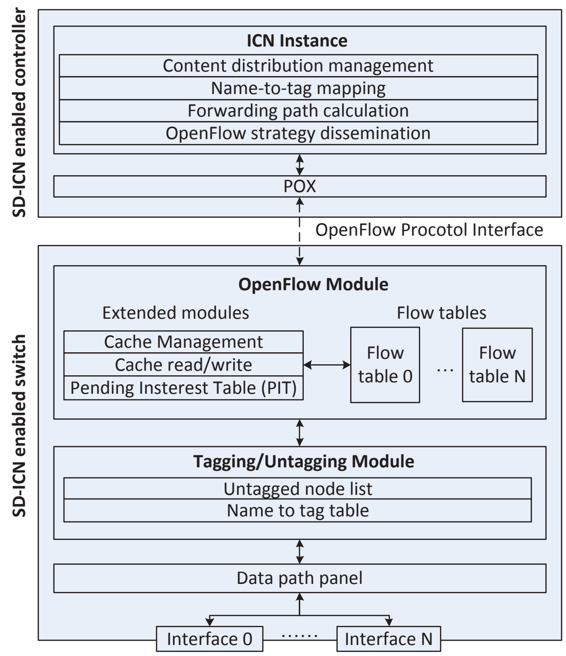

Fig. 1. An Overview of SD-ICN Design.

To enable the communication between the data plane and control plane, OpenFlow, a standardized interface, is proposed [4]. As the ICN architectures are still in the stage of evolvement, to deploy ICN architectures over virtual switches supporting OpenFlow is more desirable than that over OpenFlow-enabled hardware devices. Therefore, we choose to design a unified deployment and interoperability framework for ICNs over Open vSwitch (OVS) [5], a multilayer virtual switch fully supporting OpenFlow with production quality. The main contributions of our work are summarized as follows:

- We abstract common function modules, such as content distribution management and name-to-tag mapping, for deploying and inter-operating different ICN architectures.
- We extend the OpenFlow interface to disseminate cache and interoperability related decisions.
- We define a unified packet tagging scheme to forward packets of different ICN architectures over the same physical network.
- We extend Open vSwitch to tag the original ICN packets when they arrive at the ingress routers and untag the tagged packets when they reach the egress routers.
- We extend the flow tables to support cache-related opera-

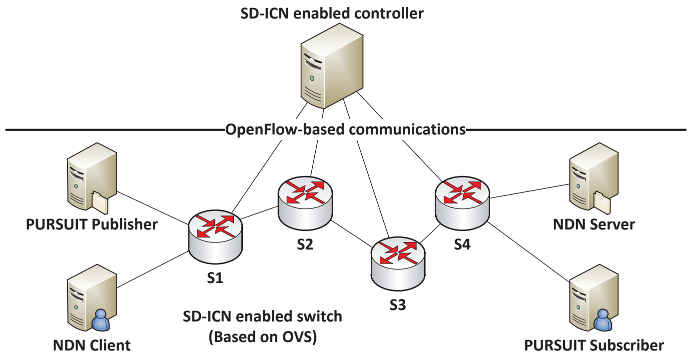

Fig. 2. The Demonstration Scenario

tions including shared cache management.

II. SD-ICN Overview

Fig. 1 illustrates the design of our SD-ICN that contains two main components: the SD-ICN enabled controller and SD-ICN enabled switches.

The SD-ICN enabled controller has four abstracted function modules: (1) **Content distribution management** that is used to manage content availability information submitted by content providers, *i.e.,* registrations in NDN or publications in PURSUIT, and to manage the cached copies in the network; (2) **Forwarding path calculation** that is responsible for calculating the best path from the requesting node to an appropriate server according to the decisions made by content distribution management; (3) **Name-to-tag mapping** that is designed to handle the interoperability issue among different ICN architectures by assigning a single content object with a unique label, which is then attached to the packet and used to identify the corresponding content in switches. (4) **OpenFlow strategy dissemination** that is implemented to disseminate control messages beyond the forwarding path, such as name mapping decisions.

A SD-ICN enabled switch is implemented based on OVS. To forward different ICN packets, we define a **unified packet tagging scheme** by exploiting the Multi-Protocol Label Switching (MPLS) tag. We use the **tagging/untagging module** to perform relevant operations. We extend the OpenFlow module of OVS (*i.e.,* **extended modules**) to perform cache-related operations and some particular operations such as updating Pending Interest Table (PIT) for NDN *INTEREST* packets.

III. Demonstration

Fig. 2 shows the demonstration scenario, in which the client and server nodes in SD-ICN are installed with the standard ICN applications. In our demonstration, we only consider two ICN architecures, *i.e.,* NDN and PURSUIT, as they are relatively more infusive than other ICNs.

The demonstration includes two parts.

Packet In →

Flow table 0: control table

Flow	Flow Key	Priority	Actions
1	dl_type=MPLS && protocol=NDN && pkt_type=Interest	default	resubmit(,1)
2	dl_type=MPLS && protocol=NDN && pkt_type=Data	default	**write_cache**
3	dl_type=MPLS && protocotol=PURSUIT && pkt_type=Data	default	resubmit(,2)
4	*	0	Controller

Flow table 1: for NDN interest packets

Flow	Flow Key	Priority	Actions
1	content_tag=X	default	**read_cache**, output: 1
2	content_tag=Y	default	**read_cache**, output: 2
...	...	...	...

Flow table 2: for PURSUIT data packets

Flow	Flow Key	Priority	Actions
1	content_tag=X && dst_mac=A	default	**write_cache**, output: 1
2	content_tag=Y && dst_mac=B	default	**write_cache**, output: 2
...	...	...	...

Fig. 3. Flow Tables in OVS

1) Implementing a particular ICN architecture: As a fundamental requirement, SD-ICN should be able to support the original functions of NDN and PURSUIT. The experiments demonstrate the basic operations for each ICN. For NDN, an *INTEREST* packet is routed according to the matching flow entry created by the SD-ICN enabled controller and a *DATA* packet is forwarded by tracking back the PIT entries created when the corresponding *INTEREST* packet was forwarded. For PURSUIT, the content distribution management and forwarding path calculation modules in the controller are implemented as the rendezvous and topology manager of PURSUIT. The flow tables in OVS created by the controller are shown in Fig. 3.

2) Interoperating two different ICN architectures: Two types of interoperations are considered in the demonstration: a PURSUIT publisher serves a request issued by an NDN client and an NDN server responds to a subscription of a PURSUIT subscriber. As switches identify the content by the unified label assigned by the controller, different ICNs can share the cached content through the extended OpenFlow actions **read_cache** and **write_cache** shown in Fig. 3. The demonstration shows the benefits of interoperability regarding to response time and network throughput in the case of requesting media files due to sharing cached content between two different ICN architectures.

IV. Conclusion and Future Work

In this demo, we demonstrate our proposed unified ICN deployment and interoperability framework, called SD-ICN, which is developed on top of OVS and supports fast ICN deployment and interoperation among different ICNs. Our next step is to develop applications such as DASH and social TV over the implemented SD-ICN framework.

References

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. of the 5th ACM CoNEXT*, New York, USA, 2009, pp. 1–12.

[2] N. Fotiou, D. Trossen, and G. C. Polyzos, "Illustrating a publish-subscribe internet architecture," *Telecommun. Syst.*, vol. 51, no. 4, pp. 233–245, 2012.

[3] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, and S. Shenker, "Less pain, most of the gain: Incrementally deployable ICN," in *Proc. of the ACM SIGCOMM'13*, New York, USA, 2013, pp. 147–158.

[4] 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, 2008.

[5] B. Pfaff, J. Pettit, T. Koponen, K. Amidon, M. Casado, and S. Shenker, "Extending networking into the virtualization layer," in *Proc. of the 8th ACM HotNets*, New York, USA, 2009.