Network Abstraction and Control Models for Hierarchical SDN Controllers

Y. Iizawa[a*], M. Morimoto[a], T. Koide[b], Y. Ashida[a], and H. Shimonishi[a]
[a]*Knowledge Discovery Research Laboratories, NEC Corporation*
[b]*Optical IP Development Division, NEC Corporation of America*
[*] y-iizawa@cd.jp.nec.com

1. Introduction

The advent of SDN has removed the constraints resulting from rigid control protocols, thereby enabling a flexible description of control software. OpenFlow [1] is commonly used as a unified south-bound protocol but concurrent support of multiple south-bound protocols is also required for integrated control of various networks, including packet/optical transport, mobile/wireless, and overlay networks. Thus, lack of unified model for abstracting these networks results in less-reusable SDN applications. For example, developing different path computation codes for different networks and manually combining the codes for multi-domain routing would make end-to-end path optimization difficult. OpenDaylight [2] controller has a SAL (Service Abstraction Layer) but it does not propose any modeling for unified network abstraction nor control.

In this paper, we first propose a network abstraction model to simplify the description of various types of networks. We also propose a network control model to structure complex networks in a hierarchical way. Then we introduce our implementations of flexible SDN controller platform called ODENOS (Object-Defined Network OS) based on the proposed abstraction and control models. As a use case of ODENOS, we show integrated control of heterogeneous networks including wide-area transport networks consisting of optical and packet transport, as well as multi-layer data center networks consisting of OpenFlow and VXLAN.

2. Network abstraction and control models

Physical/logical networks are represented through a technology-independent network graph. There are standard models for network management provided by ITU-T [3], TMF [4], and so on. These models are suitable for strict management but highly complex to implement various control algorithms including dynamic routing. We rearrange these models for supporting unified control of various physical/logical networks. The model maintains a base information model easily applicable to various types of networks, as well as extensions for expressing various characteristics and constraints of physical devices. As shown in Fig. 1, the network abstraction model is defined by three types of data:

Topology: graph-based representation of network structure and statistics. Changes of states are notified as events through this model.

Flow: point/multipoint-to-point/multipoint communications, such as a set of OpenFlow flow entries, MPLS / optical paths, and overlay tunnels. A flow is characterized by a set of matches (input point information), path (route), and edge actions (output point information).

Packet: in-band protocol-dependent communication like OpenFlow packet_in/out.

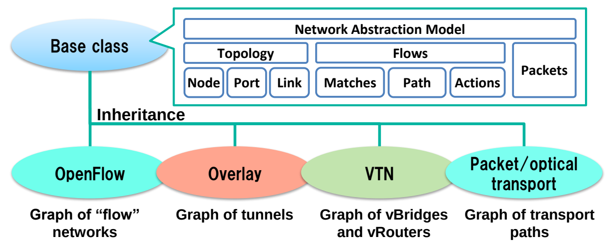

Fig. 1: Network abstraction model

We also define a technology-independent control model to construct complex networks as a combination of unit networks. As shown in Fig. 2, based on the abstraction model, we define atomic graph operators including;

Aggregator: aggregating a whole network into a single logical node for hierarchical control,

Slicer: slicing a network into multiple virtual networks having same topology but isolated name spaces,

Federator: combination of multiple network domains into a single network instance,

Layerizer: mapping multiple network layers into a single network instance.

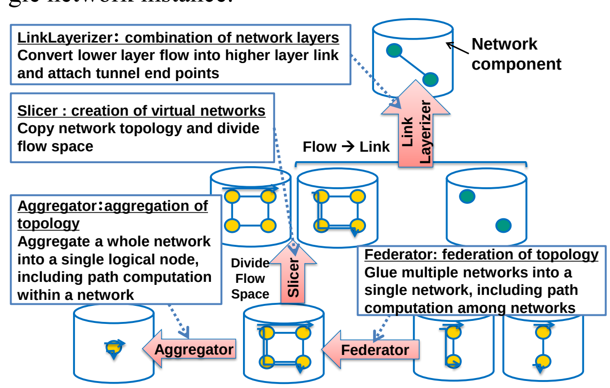

Fig. 2: Network control model

Combination of these operators realizes integration and virtualization of networks, and facilitates multi-layer / multi-domain resource optimization, end-to-end service provisioning, and holistic visualization of the networks. The constructed network instance by the operators can also be exposed to various SDN applications through simple service level interface.

3. Controller implementation

Based on the proposed abstraction and control models, we designed and implemented a prototype of an SDN controller platform called ODENOS (Object-Defined Network OS). With this platform, a SDN controller is built as a combination of various network components and logic components. A network component is an instance of network abstraction model and a logic component is an instance of operators. Basic components with technology-independent control model are easily reused for various controllers, and derivatives of these components are also easily designed by exploiting extensions for physical characteristics and constraints.

Fig. 3 shows an example of a SDN controller by combining the components in a hierarchical way. The logic components include the drivers that abstract the network information of the respective underlying systems through south-bound APIs (CDPIs), the federator that integrates them as one network, and the slicer that slices them further into multiple users. Constructed networks components can also be exposed to various external SDN applications like BGP peering or trouble-shooting through north-bound APIs. External applications can easily control virtual networks without caring about underlying network characteristics by accessing high-order network components, i.e. #4-#6, or can obtain deeper information for holistic failure analysis by accessing all network components.

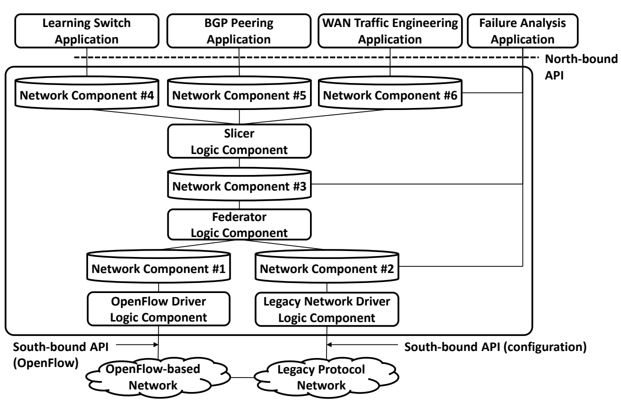

Fig. 3: Example of SDN controller with ODENOS

4. Application of ODENOS to multi-layer control

Fig. 4 shows an application of ODENOS to multi-layer network control. Applying ODENOS to wide-area transport networks consisting of an optical layer (lower layer) and a packet transport layer (upper layer), appropriate flows (paths) in the lower layer are automatically set up before setup of a requested flow in the upper layer. In this case, LinkLayerizer makes virtual links in the upper layer associated with potential flows in the lower layer, which can be set up using remaining resources in the lower layer. LinkLayerizer also makes links in the upper layer corresponding to the flows in the lower layer that are already set up. When a flow through virtual links in the upper layer is requested, LinkLayerizer sets up flows in the optical layer associated with the virtual links before setup the requested flows in the upper layer.

Applying ODENOS to data center networks consisting of OpenFlow (underlay) and VXLAN (overlay), requested bandwidth in VXLAN can be guaranteed over OpenFlow network. In this case, NodeLayerizer combines OpenFlow networks and VXLAN networks as multi-layered networks, maintains remaining bandwidth of OpenFlow networks and deploys VXLAN virtual networks to the OpenFlow networks which can guarantee bandwidth requested by the VXLAN virtual networks.

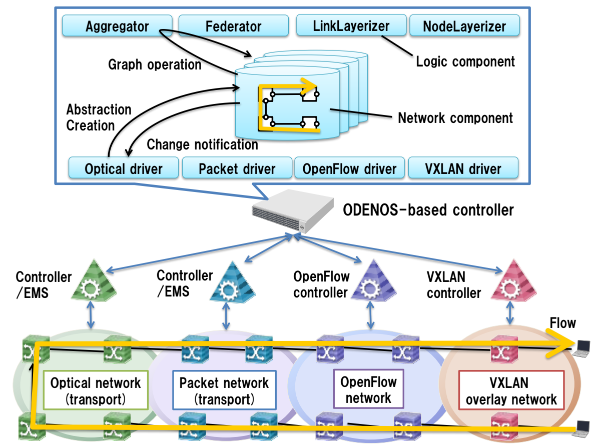

Fig. 4: Application to multi-layer control

5. Conclusion

This paper proposed network abstraction and control models for hierarchical SDN controllers. We also introduced implementation of SDN controller platform called ODENOS and applications to multi-layer control. Our SDN controller platform enables network operators to integrate the control of heterogeneous networks.

Acknowledgement

This work is partly supported by the MIC (Ministry of Internal Affairs and Communications).

References

[1] 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. 2008.

[2] OpenDaylight, http://www.opendaylight.org

[3] "Unified functional architecture of transport networks," ITU-T Recommendation G.800, 2012.

[4] Information Framework (SID), TM Forum, http://www.tmforum.org/InformationFramework/1684/home.html