

Model of Transport SDN and MPLS-TP for T-SDN Controller

Chang-Gyu LIM*, Soo-Myung PAHK*, Young-Hwa KIM*

*ETRI (Electronics and Telecommunications Research Institute), Daejeon, Korea

{human, smpahk, yhwkim}@etri.re.kr

Abstract— SDN (Software Defined Networking) is an approach to computer networking that allows network administrators to manage network services through abstraction of higher-level functionality. It has attracted attentions for even transport networks. For the management of transport network service: especially MPLS-TP(Multiprotocol Label Switching – Transport Profile), we use T-SDN(Transport SDN) controller which manages network services and resources between NBI(North Bound Interface) and SBI(South Bound Interface), based on OpenDaylight. This paper proposes the model of transport SDN and MPLS-TP. With using proposed model, T-SDN controller can make an intra/inter-domain network service easily.

Keywords— Transport, SDN, model

I. INTRODUCTION

In these days, network providers want to simplify a network management. This is done by decoupling the legacy network system that is composed of a control plane and a data plane. SDN(Software Defined Networking) divides a network system into a decision plane(control plane) and a forwarding plane(data plane) and it is an approach to computer networking that allows network administrators to manage network services through abstraction of higher-level functionality. It has attracted attentions for even transport networks.

For the management of transport network service: especially MPLS-TP(Multiprotocol Label Switching – Transport Profile), we use T-SDN(Transport SDN) controller, which is based on OpenDaylight[1] controller, for managing network services and resources with NBI(North Bound Interface) and SBI(South Bound Interface). T-SDN controller uses YANG data models for transport network abstraction.

YANG is a data modeling language used to model configuration and state manipulated by the NETCONF(Network Configuration Protocol). YANG is based on standard IETF document [2].

In Section 2, we describe YANG data modeling and features of the OpenDaylight controller.

In Section 3, we propose a generic transport SDN model and an architecture of T-SDN controller.

Section 4 shows a MPLS-TP extended model for T-SDN.

II. YANG DATA MODEL AND OPENDAYLIGHT CONTROLLER

In this section, we describe some syntax of YANG language and features of the OpenDaylight controller.

A. YANG Data Module

A YANG data module defines a hierarchy of data including configuration data, state data, RPCs(Remote Procedure Calls) and notifications. Figure 1 shows an example of a yang data module and using its operations.

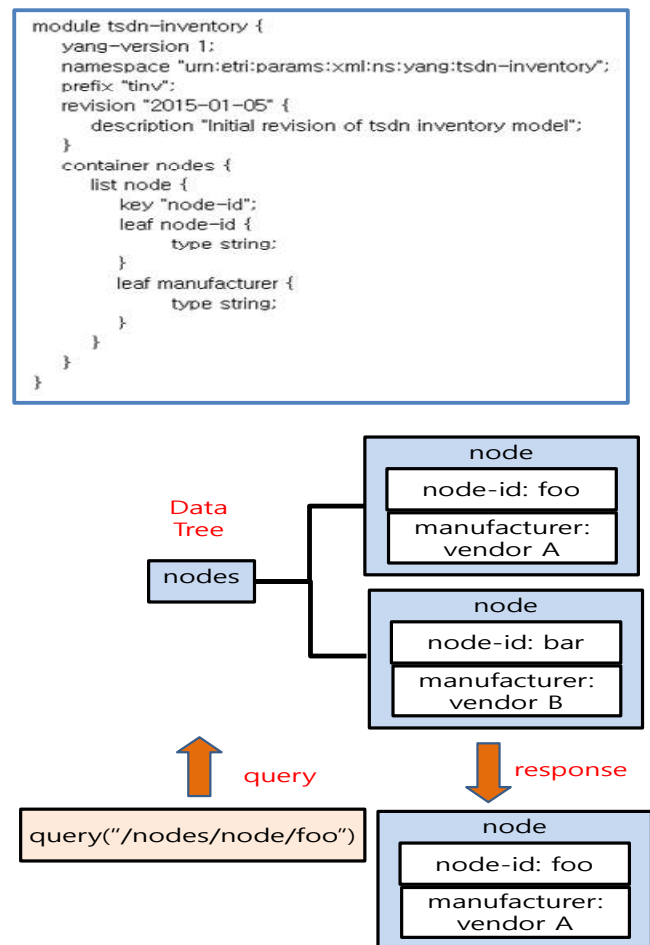


Figure 1. An example of a yang data module and using its operations

A container node is used to group related nodes in a subtree. A list defines a sequence of list entries. Each entry is like a structure or a record instance, and is uniquely identified by the values of its key leafs. A leaf node contains simple data link an integer or a sting. We can create, read, update and delete some nodes in the data tree.

B. Remote Procedure Call

YANG provides the definition of RPCs. The operations' names, input parameters and output parameters are modeled using YANG data definition statements. Figure 2 shows an example of a RPC definition in the yang module.

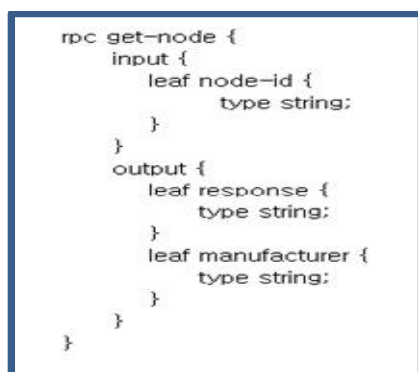


Figure 2. An example of a RPC definition in the yang module

RPCs are used to model any procedure call implemented by a provider(server), which exposes functionality to consumers(clients). RPCs are triggered by consumers and usually have return value.

C. Notification

YANG provides the definition of notifications. The notifications' names and parameters are modeled using YANG data definition statements. Figure 3 shows an example of a notification definition in the yang module.

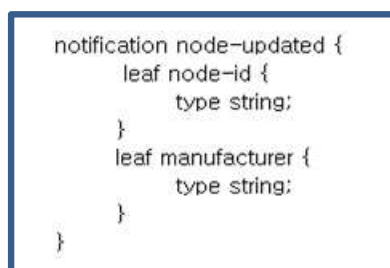


Figure 3. An example of a notification definition in the yang module

Notifications represent asynchronous events, published by providers for listeners. A provider publishes notifications and Consumers listen notifications which are concerned to know.

D. Extending Data Models

YANG provides a module to insert additional nodes into data models, including both the current module or an external module. Figure 4 shows an example of an augmentation

definition in another yang module and using its data model extension.

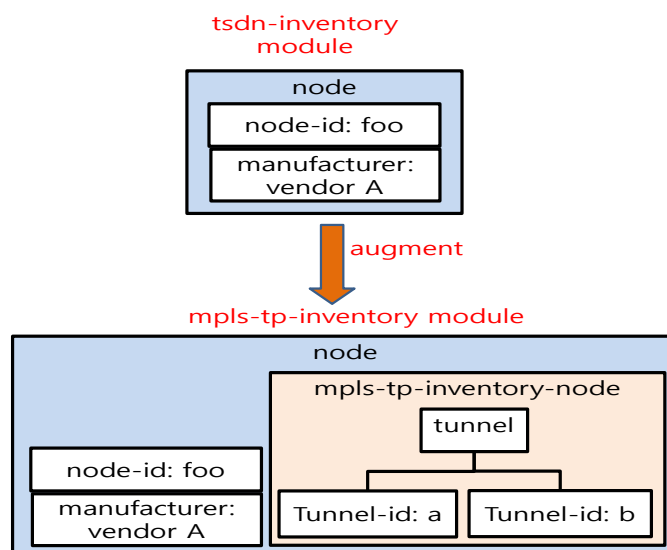
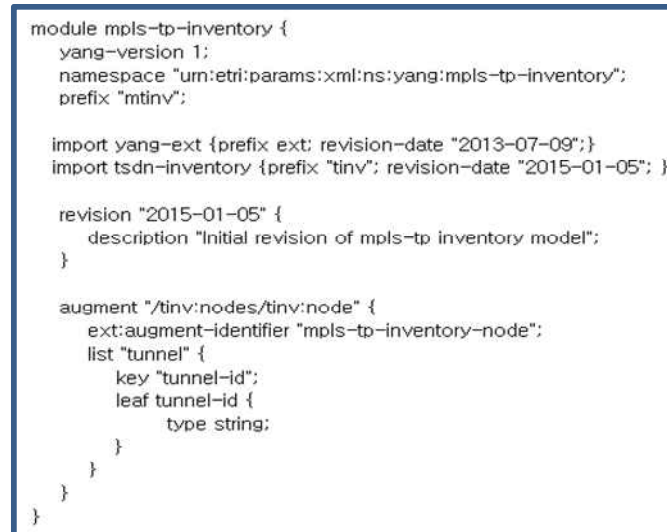


Figure 4. An example of an augmentation definition in another yang module and using its data model extension

Additionally, augmentation is also used to extend RPC's input parameters, output parameters and notification's parameters.

E. OpenDaylight Controller

MD-SAL(Model-Driven Service Abstraction Layer) is a set of infrastructure services aimed at providing common and generic support to application and plugin developers. The modeling language of MD-SAL is YANG.

MD-SAL currently provides infrastructure services for data services(datastore), RPC routing, notification brokering.

Data services provide configuration and operational data trees which are like data stores. Figure 5 shows an example of data trees for a data model defined in a specific yang module. Each tree structure represents the configuration(red) or operational(blue) state of components and systems.

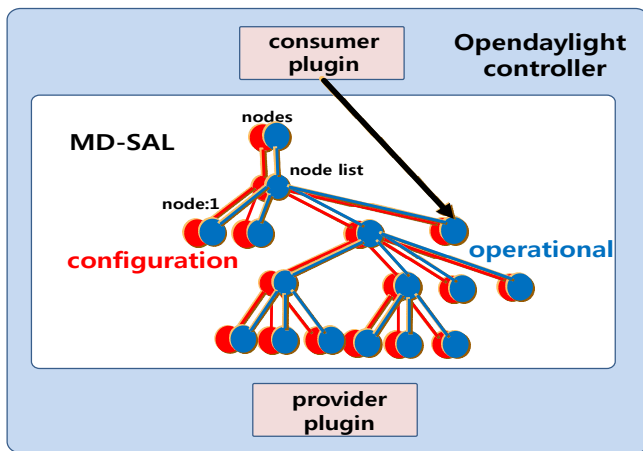


Figure 5. An example of data trees for a data model

An OpenDaylight controller has some types of plugins. A consumer type plugin consumes some data from MD-SAL and a provider type plugin provides some data to MD-SAL. If a consumer calls a RPC, MD-SAL connects this call to a provider which implements the RPC. If a provider publishes a notification, MD-SAL notifies this notification to consumers which implement the notification. Figure 6 shows an example of RPC routing and notification brokering for a data model defined in a specific yang module.

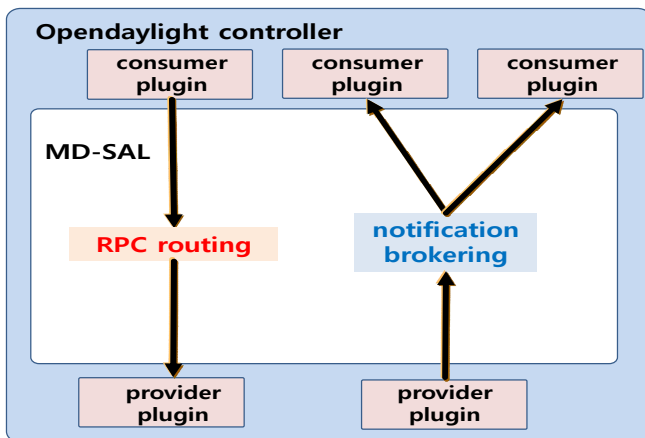


Figure 6. An example of RPC routing and notification brokering for a data model

III. TRANSPORT SDN DATA MODEL AND CONTROLLER

The purpose of this model provides a generic data model defined by YANG data modeling language that consists of data structures, NBI and SBI for T-SDN controller to control transport networks. This proposed model addresses to meet the requirements of interface specification used by a series of processes such as receiving service requests, setting up service profiles, gathering and updating information, storing topology and inventory information according to the network service scenario. Figure 7 shows an architecture of T-SDN controller.

T-SDN controller needs some features, inventory, topology and service management for supporting a network service. In order to provide these features, we design a generic T-SDN YANG modules.

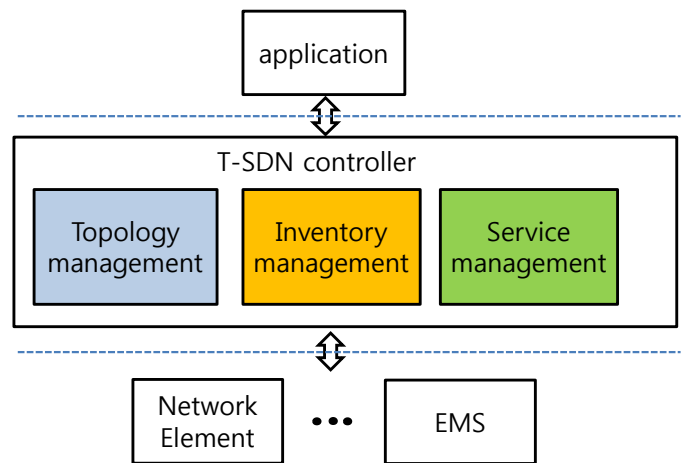


Figure 7. An architecture of T-SDN controller

The generic T-SDN inventory YANG module provides the lists below.

- Nodes container: it contains a node list.
- Node list: it is composed of a node-id(key value), node's attributes and a node-connector list.
- Node-connector list: it contains a node-connector-id(key value) and node-connector's attributes.
- RPCs: to control each provider (connected to NE(Network Element) or EMS(Element Management System)) for providing network services by calling RPCs. There are tunnels', tunnel cross-connects' and access interfaces' RPCs for CRUD(Create, Read, Update, Delete) respectively.
- Notifications : to notify a state of a component from a provider (connected to NE or EMS) to consumers(a inventory manager plugin) by notification of updated/ removed nodes, node-connectors, tunnels, tunnel cross-connects and access interfaces respectively.

The generic T-SDN topology YANG module provides the lists below.

- Network-topology container: it contains a topology list and a tunnel list.
- Topology list: it is composed of a topology-id(key value), topology's attributes, a node list and a link list.
- Tunnel list: it is composed of a tunnel-id(key value) and tunnel's attributes.
- Node list: it is composed of a node-id(key value), node's attributes and a node-connector list.
- Node-connector list: it contains a node-connector-id(key value) and node-connector's attributes.
- Link list: it contains link-id(key value) and link's attributes.

The generic T-SDN topology discovery YANG module provides the lists below.

- Grouping link: it contains link's attributes.
- Grouping topology: it contains topology's attributes.
- Notifications : to notify a state of a component from a provider (NE or EMS) to consumers(a topology

manager plugin) by notification of updated/removed topologies and links respectively.

The generic T-SDN service YANG module provides the lists below.

- Services container: it contains a service list.
- Service list: it is composed of a service-id(key value), and service's attributes.
- RPCs: applications call to request a network service. There are service RPCs for CRUD respectively.
- Notifications : to notify a state of a component from a provider (connected to NE or EMS) to consumers(a service manager plugin) by notification of updated services and removed services.

IV. MPLS-TP DATA MODEL

Figure 8 shows the block diagram of the proposed model which is related between a generic model and an extended model. If you want to make a new transport network model, you can extend the generic T-SDN module for designing a new transport network model easily.

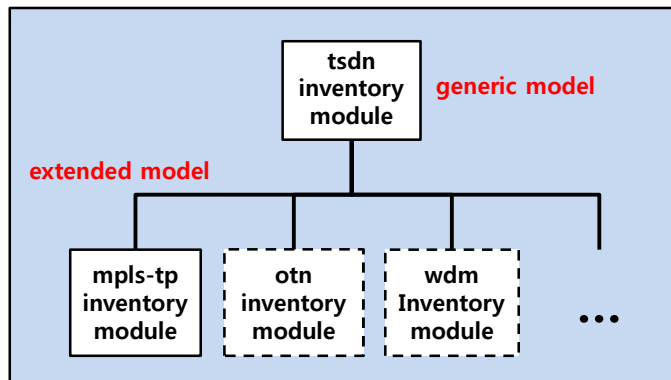


Figure 8. An extension of the generic T-SDN inventory YANG module

The extended MPLS-TP inventory YANG module provides the lists below.

- Augmentation of the generic T-SDN inventory node: a MPLS-TP supported node needs new components like a tunnel list, a tunnel cross-connect list, a mpls interface list, an access interface list, a pseudo wire list and a pseudo wire cross-connection list.
- Each new component list: it is composed of its id(key value) and component's attributes.
- Augmentation of the generic T-SDN inventory RPCs: RPC's input or output parameters are extended. There are mpls-tp extended tunnels', tunnel cross-connects' and access interfaces' RPCs for CRU(Create, Read, Update) respectively.
- RPCs: adds new RPCs for MPLS-TP. There are pseudo wires', pseudo wire cross-connects' and mpls interfaces' RPCs for CRUD respectively.
- Augmentation of the generic T-SDN inventory notifications: There are notifications of updated mpls-tp nodes, node-connectors, tunnels, tunnel cross-connects and access interfaces respectively.

- Notifications : adds new notification for MPLS-TP. There are notifications of updated/removed mpls-tp pseudo wires, pseudo wire cross-connect and mpls interfaces respectively.

In the same way, rest of the generic T-SDN YANG modules are extended.

The extended MPLS-TP topology inventory YANG module provides the lists below.

- Augmentation of the generic T-SDN topology node: a MPLS-TP supported topology node needs new components like an inventory node reference and a switching capacity, etc.
- Augmentation of the generic T-SDN topology link: a MPLS-TP supported link needs new components like a link type, a link status and an available bandwidth, etc.
- Augmentation of the generic T-SDN topology tunnel: a MPLS-TP supported tunnel needs new components like an available bandwidth, etc.

The extended MPLS-TP topology discovery YANG module provides the lists below.

- Augmentation of the generic T-SDN topology discovery notification: There is a notification of updated mpls-tp links.

The extended MPLS-TP service YANG module provides the lists below.

- Augmentation of the generic T-SDN service: a MPLS-TP supported service needs new components like a service constraint list, a node list, a ero-pw-connection list and mpls-tp service attributes.
- Each new component list: it is composed of its id(key value) and component's attributes.
- Augmentation of the generic T-SDN service RPCs: RPCs' input or output parameters are extended. There are mpls-tp extended service RPCs for CRU respectively.
- Augmentation of the generic T-SDN service notifications: There is a notification of updated mpls-tp services.

V. CONCLUSIONS

This paper proposes the model of transport SDN and MPLS-TP. We create the generic model of transport SDN. Therefore, if we want to make a new transport network model, we can extend the generic T-SDN module for designing a new transport network model like MPLS-TP, OTN, WDM, etc. With using proposed model, T-SDN controller can make an intra/inter-domain network service easily.

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (B0101-15-233, Smart Networking Core Technology Development).

REFERENCES

- [1] <http://www.opendaylight.org>
- [2] M. Bjorklund, *RFC6020 YANG-A Data Modeling Language for the Network Configuration Protocol*, IETF, Oct. 2010.



Chang-Gyu LIM is a senior engineer of SDN Research Section, ETRI, Korea. He received his Master degree at KAIST in 2002. His key research interests are: Future Internet, Software Defined Networking and Transport Network.



Soo-Myung PAHK is a principal engineer of SDN Research Section, ETRI, Korea. He received his Ph.D. degree at Konkuk Univ. in 1999. His key research interests are: Future Internet, Software Defined Networking and Transport Network.



Young-Hwa KIM is a principal engineer of SDN Research Section, ETRI, Korea. He received his Ph.D. degree at Chungnam Univ. in 2005. His key research interests are: Future Internet and Software Defined Networking.