

Model of Transport SDN for Path Computation

Chang-Gyu LIM*, Soo-Myung PAHK*

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

{human, smpahk}@etri.re.kr

Abstract— SDN (Software Defined Networking) allows network administrators to manage network services through abstraction of higher-level functionality. It has applied to even transport networks. We already develop T-SDN(Transport SDN) controller which manages network resources between NBI(North Bound Interface) and SBI(South Bound Interface), based on OpenDaylight for the management of transport network services. But, we need a method of creating network services easily. Therefore, the model of T-SDN needs to add a path computation functionality. This paper proposes the model of transport SDN for path computation. With using proposed model, T-SDN controller can make network services easily.

Keywords— Transport SDN, path, computation, model

I. INTRODUCTION

Nowadays, network providers want to handle a network management. It is done by decoupling the legacy network system composed of control and data planes. SDN(Software Defined Networking) divides a network system into a control plane and a data plane. It allows network administrators to manage network services through abstraction of higher-level functionality. It has applied to even transport networks.

We already develop T-SDN(Transport SDN) model[1] and T-SDN controller which is based on OpenDaylight[2] controller(Lithium version), for managing network services and resources with NBI(North Bound Interface) and SBI(South Bound Interface) for the management of transport network services. We use YANG data models for transport network abstraction. YANG is a data modeling language based on standard IETF document[3]. Figure 1 shows an architecture of T-SDN controller.

But, we need a method of creating network services easily. Therefore, the model of T-SDN needs to add a path computation functionality.

In Section 2, we describe proposed YANG data models for path computation. Former T-SDN model[1] is expanded to the proposed model. We explain a path computation procedure, briefly.

Section 3 shows our implementation and the result of the path computation using the proposed model. We implement the path computation plugin using the proposed model in T-SDN controller.

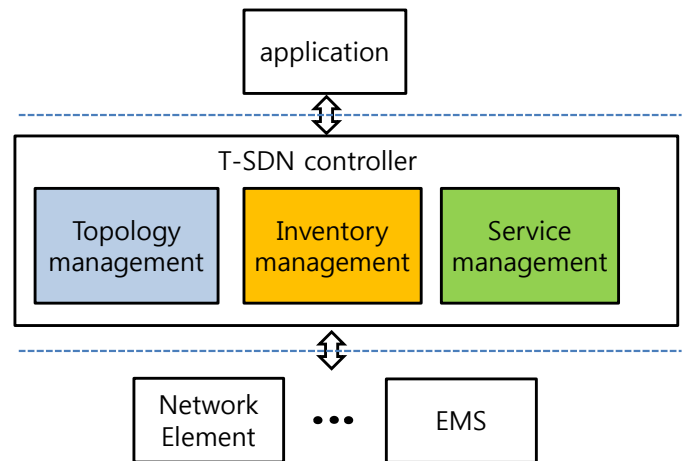


Figure 1. An architecture of T-SDN controller

II. PROPOSED YANG DATA MODEL FOR PATH COMPUTATION

In this section, we describe a proposed YANG data model for path computation. Former T-SDN model[1] is expanded to the proposed model. We also explain a path computation procedure, briefly.

```

container network-topology {
  list topology {
    key "topology-id";
    .....
    list node {
      key "node-id";
      .....
      list node-connector {
        key "node-connector-id";
        .....
      }
    }
    list link {
      key "link-id";
      .....
    }
    list tunnel {
      key "tunnel-id";
      .....
    }
  }
}

```

Figure 2. T-SDN network topology YANG data model

A. Path Computation Procedure

Firstly, a path computation needs network topology resources, like nodes, links, topologies and bandwidths. Former T-SDN model has not only the topology model but also the inventory model. But the proposed path computation model is forced to define to get information from the topology model only. Therefore, new topology model should be expanded and we don't need the T-SDN inventory model data.

Secondly, we have to calculate the metric values for a path computation. Links and tunnels have a metric value. And a path decision maker chooses the path which has a smallest total metric values. So, we have to add a metric value attribute to new topology data model.

Finally, we make input parameters for a path request RPC. In this step, we select source and destination nodes, a network type, some path constraints, a bandwidth for input parameters. Path request RPC output parameters also should be defined. They are composed of candidate paths which are sequences of path connections.

B. T-SDN Network Topology YANG Data Model

Figure 2 shows the former model of T-SDN network topology YANG data model. Network topology container contains a topology list and a tunnel list. Tunnels means paths. A tunnel is a path from source node to destination node.

```

list tunnel {
  key "tunnel-id";
  .....
  leaf te-metric-value {
    type decimal64 {
      fraction-digits 3;
      range "0.000..100.000";
    }
  }
}
list link {
  key "link-id";
  .....
  leaf te-metric-value {
    type decimal64 {
      fraction-digits 3;
      range "0.000..100.000";
    }
  }
}

```

Figure 3. T-SDN network topology YANG data model modification

Now, we expand this former T-SDN model. For path computation, we add a te-metric-value attribute to tunnels and links. In Figure 3, te-metric-value leaf is shown. If we use MPLS-TP(Multiprotocol Label Switching - Transport Profile) networks, it should be considered that MPLS label allocation is necessary. A specific MPLS in-label value is permitted only once per MPLS-TP node. Therefore, we expand a generic topology node to a MPLS in-label allocation enabled node with adding supported/available MPLS in-label range containers. Figure 4 is shown. Mpls-tp-topology-node is

created by the augment method. Expanded node has supported/available MPLS in-label range containers. We also expand a generic topology link to a bandwidth aware link with adding maximum/available bandwidth attributes. This is shown at Figure 5.

```

augment "/topo:network-topology/topo:topology/topo:node" {
  ext:augment-identifier "mpls-tp-topology-node";
  .....
  container supported-mpls-in-label-range {
    leaf start-point {
      type uint32;
    }
    leaf end-point {
      type uint32;
    }
  }
  container available-mpls-in-label-range {
    list "label" {
      key "index";
      leaf index {
        type string;
      }
      leaf start-point {
        type uint32;
      }
      leaf end-point {
        type uint32;
      }
    }
  }
}

```

Figure 4. T-SDN network topology YANG data model node expansion

```

augment "/topo:network-topology/topo:topology/topo:link" {
  ext:augment-identifier "mpls-tp-topology-link";
  .....
  leaf maximum-bandwidth {
    type uint32;
  }
  leaf available-bandwidth {
    type uint32;
  }
}

```

Figure 5. T-SDN network topology YANG data model link expansion

C. Path Computation Request Remote Procedure Call

YANG provides the definition of RPCs which are used to call implemented by a provider. We can make a call with input parameters of the defined path request RPC and get a wanted path result from output parameters of the defined path request RPC. Figure 6 shows the request-path RPC definition in tsdn-pce YANG module.

We describe input attributes first. Network-types attribute can choose a network type of a request path. For example, there are some types of transport networks, like MPLS-TP, OTN(Optical Transport Network), SDH(Synchronous Digital Hierarchy), WDM(Wavelength Division Multiplexing) and multi-layered networks. Source and destination containers mean each edge node of a request path. CIR means a request bandwidth of a service in MPLS-TP network. We also have path-computation-constraint list for some defined constraints

for a request path, like number-of-candidate, include-node, exclude-node.

```

rpc request-path {
  input {
    leaf network-type {
      type ttypes:service-network-type;
    }
    container source {
      leaf node-ref {
        type tnode:node-ref;
      }
    }
    container destination {
      leaf node-ref {
        type tnode:node-ref;
      }
    }
    leaf cir {
      type uint32;
    }
    .....
    list path-computation-constraint {
      key "index":
      .....
    }
  }
  output {
    list candidate {
      key "candidate-index":
      list path {
        key "path-index":
        list connection {
          key "connection-index":
          leaf node-ref {
            type tnode:node-ref;
          }
          container mpls-tp-ingress-interface {
            .....
          }
          container mpls-tp-egress-interface {
            .....
          }
        }
      }
    }
  }
}

```

Figure 6. Request-path RPC definition

Output parameter is composed of a candidate list. A candidate has a path list. A path list has a connection list hierarchically. A connection has ingress and egress interfaces of an specific node which is on the request path. Simply, a path list stands for a sequence of ingress and egress interfaces node by node.

III.IMPLEMENTAION AND RESULT

We implement the path computaion plugin using the proposed model in T-SDN controller. Figure 7 shows the block diagram of the T-SDN controller with the path computation plugin and the topology simulator plugin which is a topology data loader. The simulator plugin transmits abstraction topology data to the topology data store in MD-SAL(Model Driven Service Abstraction Layer) which is supported by OpenDaylight controller. Operators call the path request RPC. This call is received by MD-SAL.

MD-SAL send this call to the path computation plugin by RPC broker. Finally, the path computation plugin received this call with input parameters. The path computation plugin gathers topology data from the topology data store and calculate a request path.

Figure 8 shows the result of the path request. The result is presented by web browser. There are 13 candidate paths. One of the candidate paths is drawn in Figure 8.

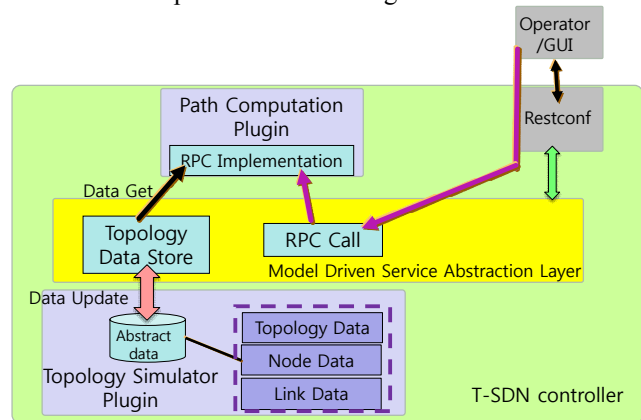


Figure 7. A block diagram of T-SDN controller

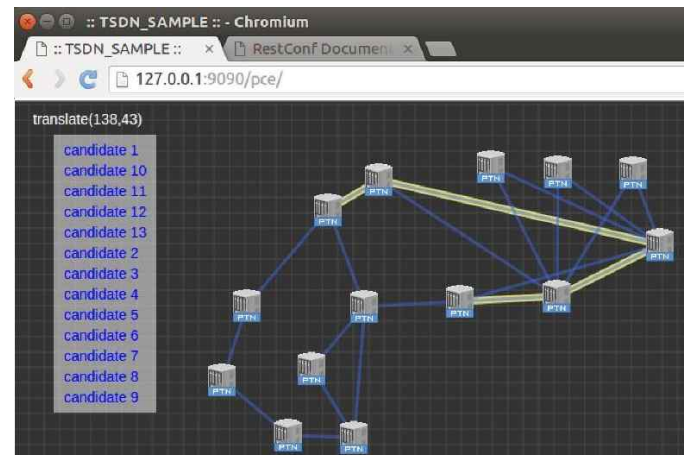


Figure 8. A result of the path request

IV.CONCLUSIONS

This paper proposes the model of transport SDN for path computation. We create and update some models of transport SDN. We also implement the path computaion plugin using the proposed model in T-SDN controller. With using proposed model, T-SDN controller can make network services easily.

ACKNOWLEDGMENT

This work was supported by ICT R&D program of MSIP/IITP. [B0101-16-0233, Smart Networking Core Technology Development]

REFERENCES

- [1] C. Lim, S. Pahk and Y. Kim, "Model of Transport SDN and MPLS-TP for T-SDN Controller", in ICAC2016, 2016, pp. 526-530.
- [2] <http://www.opendaylight.org>
- [3] 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.