Network Working Group
Internet Draft
Intended status: Standard Track
Expires: May 2016

J. Bi
Tsinghua Univ.
Q. Sun
C. Xie
China Telecom
November 4, 2015

Declarative Policy Model
draft-bi-declarative-policy-00

Abstract

This document describes a declarative model for traffic steering policies in Distributed Data Center (DDC) scenarios. The policy model is a specific data model for traffic steering using VPN technology. It helps the service management in Simplified Use of Policy Abstractions (SUPA) to model the policy (a set of constraints and rules) that defines how a VPN service is monitored by bandwidth and managed during its lifecycle.

Table of Contents

1. Introduction

In order to support the DDC service with VPN connection as well as new services, it brings new requirements for both network providers and service providers. Rapid uptake of new services requires dynamic service provisioning capabilities in the service management. This is achieved using policies that can be created by the operators once and the service management refers to these policies to infer how a given service needs to be provisioned considering the current state of the network.

In SUPA framework, network policy is a predefined rule or a set of rules that the service management use to map the service to the lower level network infrastructures.

Meanwhile, DDC service which is mainly relied on VPN [RFC4110] needs policy based management and controlling capability from the service management systems to facilitate the service deployment both inter data centers and within data center.

This document introduces YANG [RFC6020] [RFC6021] data models for SUPA configuration. Such models can facilitate the standardization for the interface of SUPA, as they are compatible to a variety of protocols such as NETCONF [RFC6241] and [RESTCONF]. Please note that in the context of SUPA, the term "application" refers to an operational and management applications employed, and possibly implemented, by an operator. The policy model is based on the first example - DDC services.

With respect to the scope, defining an information model for the policy exchange between the policy manager and policy agent and a corresponding data model based on yang to support specific DDC service use case is initial goal of this document. The protocol specific aspects are deferred to respective implementations. Also certain foundational concepts of the model are intentionally left open to enable future extension. Here the traffic steering policy in DDC use case provides a concrete example for a specific network technology and service, as what constitutes a policy could itself vary depending on the context where it is used, e.g. there could be tenant specific policies, site specific, network domain specific etc.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying [RFC2119] significance.

3. Policy Based Service Management Framework

Figure 1 illustrates the network configuration model which contains several modules for specific services such as VPN management. Basically, service model is to define the creation and configuration of the VPN service, while the policy model is more

focused on the adjustment or optimization of the flow path during the lifecycle of the VPN service based on the predefined policy.

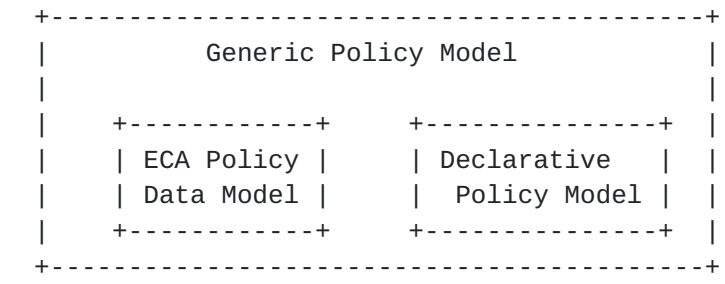


Figure 1: Overview of configuration model structure

4. Declarative Based Policy Configuration Modules

In this section, a policy model is defined with an application for traffic steering between data centers are provided to illustrate how to use the policy model. The policy model and policy configuration are based on a set of specific network services and the framework of SUPA [SUPA-framework]. On the other hand, the policy model should be working on the orchestration level which is above network element and below OSS level based on the YANG model classification in [draft-bogdanovic-netmod-yang-model classification-02]

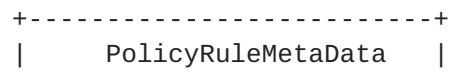
4.1. Declarative Based Policy Framework

Desired state: The description of the final state of the system, in another word, the goal or the declarative of the policy management. In SUPA scope, it consists of constraints.

Behavior constraint: a set of constraints to limit the possible operations or states in processing the policy goal to achieve the final state.

Each constraint is expressed in the form of logical , numeric and set relations which use service objects defined in service model.

Note that event, condition, and action can each be made up of Boolean clauses



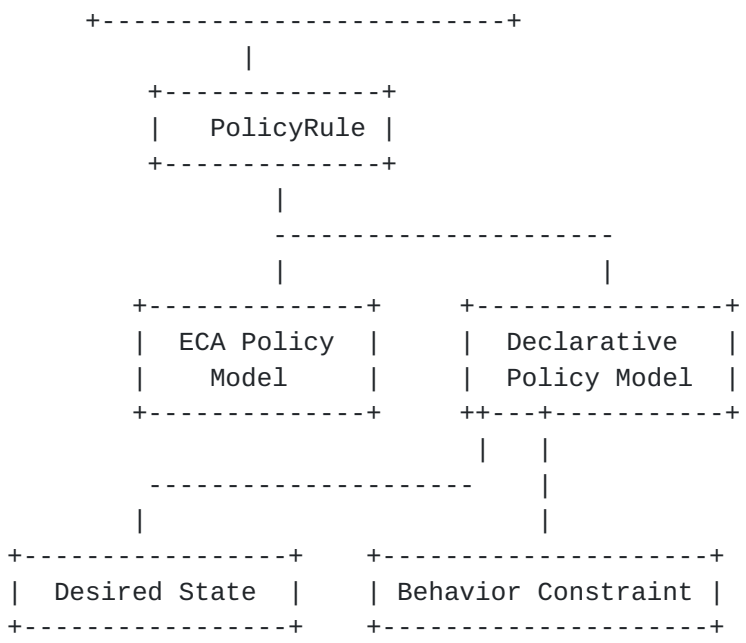


Figure 2: Overview of information declarative based policy model

```
module: SUPA-declarative-policy
   +--rw policy-set
      +--rw set-name               string
      +--rw set-type                enumeration
     +--rw applicable-service-type enumeration
     +--rw policy-rule-metadata   uint32
       +--rw policy-rule
          +--rw rule-name        string
          +--rw rule-type           enumeration
          +--rw policy-rule-priority?       uint8
          +--rw policy-validity-period
          |  +--rw start?          yang:date-and-time
          |  +--rw end?            yang:date-and-time
          |  +--rw duration?       uint32
          | +--rw periodicity      enumeration
          +--declarative-policy-rule
            +--rw desired-state
            |  +--rw constraint           string
            |  +--rw constraint-priority     uint8
            +--rw behavior-constraint
               +--rw constraint            string
               +--rw constraint-priority     uint8
```

4.2. declarative Based Policy Model

```
<CODE BEGINS> file "ietf-declarative-policy@2015-10-10.yang"
module ietf-declarative-policy {
  namespace "urn:ietf:params:xml:ns:yang:ietf-declarative-policy";
  // replace with IANA namespace when assigned
  prefix policy;

  import ietf-inet-types {
    prefix inet;
  }
  import ietf-yang-types {
    prefix yang;
  }
  organization "IETF";
  contact
    "Editor: ";

  description
    "This YANG module defines a component that describing
     the ddc policy model for monitoring and optimizing
     tenant's DC (data center) services that are deployed
     in multiple data centers.
```

```
        Terms and Acronyms
          DDC: Distributed Data Center
          L2VPN: Layer 2 Virtual Private Network
          L3VPN: Layer 3 Virtual Private Network";

     revision 2015-10-10 {
       description
         "Initial revision.";
         reference
           " Network Policy YANG Data Model ";
     }

   container policy-set{
     description
       "Policy set.";
     leaf set-name {
           type string;
           description
             "The name of the policy.";
      }
     leaf set-type {
          type enumeration {
             enum local {
               description "local";
             }
             enum globe {
               description "globe";
             }
          }
      }
     leaf policy-rule-metadata {
             type uint32;
     }

      container policy-rule {
       description
         "Declarative policy rule.";
       leaf rule-name {
           type string;
           description
             "Policy rule name.";
         }
       leaf rule-type {
           type enumeration {
             enum local {
               description "local";
             }
```

```
            enum globe {
              description "globe";
            }
          }
        }
     leaf policy-rule-priority {
            type uint8;
     }

     container policy-validity-period {
        description
         "The working period of the policy.";
       leaf start {
         type yang:date-and-time;
       }
       leaf end {
         type yang:date-and-time;
       }
       leaf duration {
            type uint32;
       }
       leaf periodicity {
           type enumeration {
            enum daily {
              description "daily";
            }
            enum monthly {
              description "monthly";
            }
          }
        }
     }
     container Declarative-policy-rule {
        description
           "Define declarative policy rule";
       container desired-state {
          description
           "Describe desired state.";
         leaf constraint {
              type string;
              description
              "Define the constraint.";
           }
         leaf constraint-priority {
           type uint8;
         }
       }
```

```
         container behavior-constraint {
            description
             "Describe the constraint on action behavior.";
           leaf constraint {
                type string;
                description
                "Define the constraint.";
             }
           leaf constraint-priority {
             type uint8;
           }
         }
       }
     }
   }
}
<CODE ENDS>
```

5. Declarative Policy Applications in DDC services

5.1. Policy Based Traffic Steering Case study

Traffic Steering use case description:

In one set of links, keep all link utilization below 70%.

If some flows need to move to other link, keep Gold user flows untouched.

After analyze above case, we structure the description as following:

Related objects: links   flow(usertype)

Goal   all link utilization < 70%

constraint: keep Gold user flows untouched

The service model of this use case:

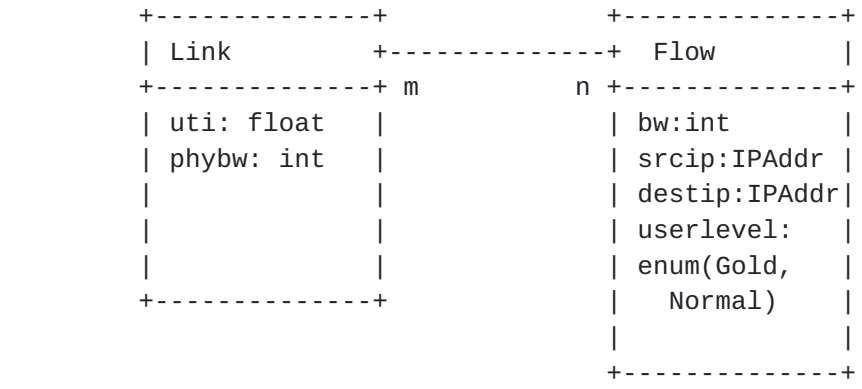


Figure 3. service model of traffic steering policy use case

Link attribute

Uti: link bandwidth utility

Phybw: physical bandwidth of the link

Flow attribute

bw: the bandwidth of the flow

srcip,destip: the source and dest ip address of the flow

userlevel: the user's service level of the flow, it can be gold or normal.

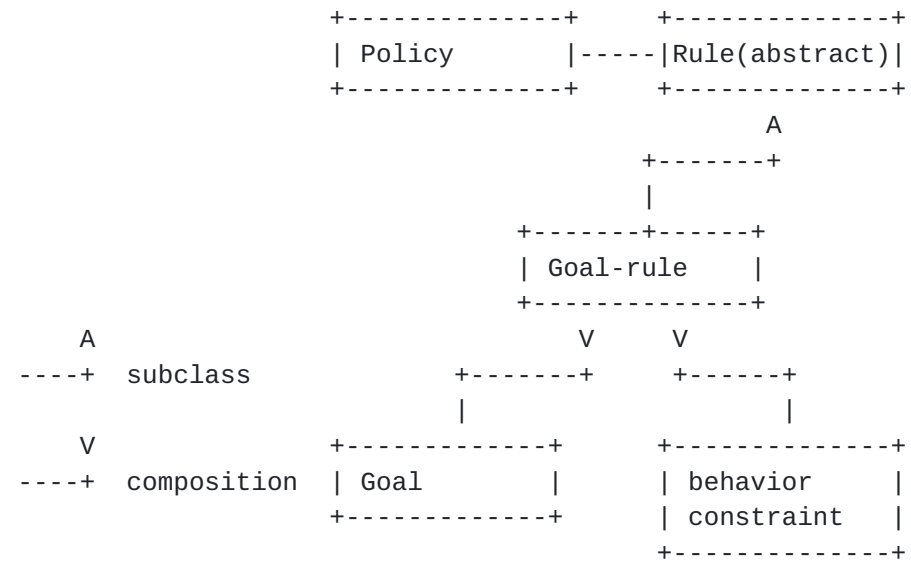
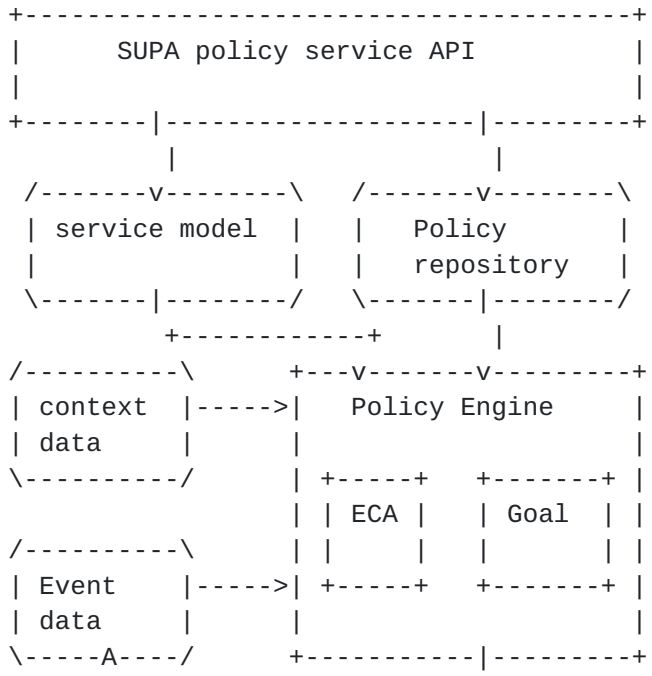


Figure 2. policy model of traffic steering policy use case

5.2. Declarative Based Policy Enforcement

Based on the service model and policy model traffic steering use case introduced in above section. This section introduce an example of policy framework and briefly illustrate how to enforce the declarative based policy.



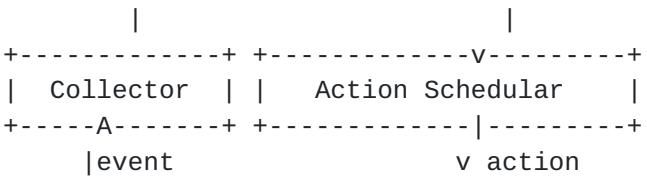


Figure 3. declarative policy framework as an example

Figure 3 shows the example framework. In the framework, the policy engine take the key role who translate the declarative policy to lower layer actions. The policy engine is problem domain agnostic. It depends service model and policy model to operate on problem domains.

Following give some brief illustration around the traffic steering use case that how the policy engine do declarative based policy enforcement.

First, to inject the capability of operating on traffic steering policy to the policy system, the guide model(see section 5.1) and the service mode(see section 5.1) is input to the system.

Then, under some concrete traffic steering scenario, one user can express the declarative by transfer the desired state and constraint to the system. After verifying the policy language script against the service model and policy syntax, the policy framework will save the user's declarative policy in policy repository.

When the policy is activated, the policy engine may get data from context data store, in this case, the data include the link, flow and there relationship information. The policy engine is guided by the guide model and user's declarative policy model, evaluate whether some selected data is compatible with the constraints and desired states.

After finding out data, the policy engine will fill the <flow,link> tuples which is the result of the 'select' to the action container, in this case is move(flow:Flow,tolink:Link)

Finally the policy engine output a list of actions such like
move(flow1,link3)
move(flow2,link5)
move(flow4,link2)

The flow1,flow2,flow4,link3,link5,link2 all comes from context data store as seen in figure 4

The policy engine even does not know what 'move' is, but the system can map the abstract move action to a concrete function at lower layer to perform the movement.

The collecting context data, making decision and output action circle may perform one or multiple times to change the traffic steering system to a new steady state and meet the user's goal.

6. Security Considerations

TBD

7. IANA Considerations

This document has no actions for IANA.

8. Acknowledgments

This document has benefited from reviews, suggestions, comments and proposed text provided by the following members, listed in alphabetical order: Junru Lin, Felix Lu, Yiyong Zha, and Min Zha.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6021] Schoenwaelder, J., "Common YANG Data Types", RFC 6021, October 2010.

[RFC3272] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and X. Xiao, "Overview and Principles of Internet Traffic Engineering", RFC 3272, May 2002.

9.2. Informative References

[SUPA-framework] C. Zhou, L. M. Contreras, Q. Sun, and P. Yegani, " The Framework of Shared Unified Policy Automation (SUPA) ", IETF Internet draft, draft-zhou-supa-framework, January 2015.

[SUPA-problem-statement] G. Karagiannis, Q. Sun, Luis M. Contreras, P. Yegani, and JF Tremblay, "Problem Statement for Shared Unified Policy Automation (SUPA)", IETF Internet draft, draft-karagiannis-supa-problem-statement, January 2015.

[SUPA-DDC] Y. Cheng,and JF. Tremblay, "Use Cases for Distributed Data Center Applications in SUPA", IETF Internet draft, draft-cheng-supa-ddc-use-cases, January 2015.

[RESTCONF] Bierman, A., Bjorklund, M., Watsen, K., and R. Fernando, "RESTCONF Protocol", draft-ietf-netconf-restconf (work in progress), July 2014.

[POLICY MODEL] Z. Wang, L. Dunbar, Q. Wu, "Network Policy YANG Data Model" draft-wang-netmod-yang-policy-dm, January 2015.

Authors' Addresses

Jun Bi
Tsinghua University
Network Research Center, Tsinghua University
Beijing 100084
China

Email: junbi@tsinghua.edu.cn

Qiong Sun
China Telecom
No.118 Xizhimennei street, Xicheng District
Beijing 100035
P.R. China

Email: sunqiong@ctbri.com.cn

Chongfeng Xie
China Telecom
No.118 Xizhimennei street, Xicheng District
Beijing 100035
P.R. China

Email: xiechf@ctbri.com.cn