Network Working Group   X. Geng
Internet-Draft   Z. Hu
Intended status: Experimental   Huawei
Expires: December 5, 2021   June 03, 2021

YANG Model for Scalable VTN
draft-geng-teas-enhanced-vpn-scalable-vtn-yang-01

Abstract

This document defines the Yang data model for scalable Virtual Transport Network(VTN).

Requirements Language

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 RFC 2119 [RFC2119].

Table of Contents

1. Introduction

[I-D.ietf-teas-ietf-network-slice-definition] defines IETF network slices that provide connectivity coupled with network resources commitment between a number of endpoints over a shared network infrastructure.

Enhanced VPN (VPN+) aims to provide enhancements to existing VPN services to support network slicing. VPN+ is composed of a VPN overlay and an underlying Virtual Transport Network (VTN) which has a customized network topology and a set of dedicated or shared resources in the underlay network. VPN+ and VTN are defined in [I-D.ietf-teas-enhanced-vpn].

[I-D.dong-teas-enhanced-vpn-vtn-scalability] describes the scalability considerations in the control plane and data plane to enable VPN+ services. In control plane, decoupling the topology and resource attributes of VTN allows that multiple VTNs share the same topology. In data plane, a global VTN-ID in the data packet is used to determine the set of resources reserved for the corresponding VTN.

This document defines the configuration yang model for scalable VTN solution.

2. VTN Yang Module Requirement

The general process of VTN configuration includes:

1. Creat VTN instance based on the network slice requirement

2. Configure the overlay network to initiate VTN in the network

3. Steer the traffic to the corresponding VTN to provide network slice service

The corresponding requirement of VTN configuration data model during the process is defined in this section.

2.1. VTN Creation

After collecting information about the underlying network topology and available resources. Each VTN can have a customized topology and a set of network resources allocated. Flexible combination is allowed when multiple VTNs may shared the same topology, or multiple VTNs may share the same set of network resources.

VTN is created with the following attributes:

- o VTN Topology: Based on the existing work in IETF, topology specification for VTN could be implemented by Multi-Topology Routing (MTR) which defined in [RFC4915], [RFC5120], or Flex-algo which is defined in [I-D.ietf-lsr-flex-algo]. Correspondingly, the topology attribute of a VTN could be determined by MT-ID or algorithm ID; Signaling extensions for VTN topology is defined in [I-D.zhu-lsr-isis-sr-vtn-flexalgo] and [I-D.ietf-lsr-isis-sr-vtn-mt] respectively.

- o Network Resource: network resource is allocated for VTN based on the requirement. For example, VTN could be bound with a layer 2 sub-interface with a subset of the link bandwidth.

- o VTN Data Plane Identifier: VTN data plane identifier is uesed to identify network resource that has been allocated for the VTN. VTN data plane identifier depends on the encapsulation type of the traffic, for example IPv6 defined in [I-D.dong-6man-enhanced-vpn-vtn-id]. VTN data plane identifier is not mandatory when there are other methods to distinguish VTN instances.

2.2. VTN Initiation

VTN initiation in the network also includes two aspects: resource allocation and traffic steering through VTN specified topology. Resource allocation is defined in this section and traffic steering is defined in the next section.

Several technologies could be used for resource allocation in the network device, for example: TSN defined in IEEE 802.1 introduces the concept of time aware shaping; FlexE provides the ability to multiplex multiple channels over one or more Ethernet links; Existing Diffserv scheduling/shaping allow the construction of virtual sub-interfaces. All these technologies could be used to dedicated resource in a shared physical interface.

The configuration of these technologies play the role of VTN initiation when the allocated resource is bound with a specified VTN instance.

2.3. VTN Traffic Steering

Just as color in SR policy defined in [I-D.ietf-spring-segment-routing-policy], color is defined as an attribute of VTN to steer the traffic.

With SR policy, traffic could be steered into a SR policy by :

o SR policy with color is provisioned to the headend;

o The route with some particular color matchs the SR policy with the corresponding color, which could satisfy the requirement of the route

o Traffic with the route is steered into the SR policy;

Similarly, traffic could be steered into VTN by:

o VTN is configured with the attribute of color;

o The route with some particular color matchs VTN with the correponding color, which could satisfy the requirement of the route

o Traffic with the route is steered to the VTN

SR policy could also be bound with VTN to provide resource reservation in the network. BGP SR Policy extensions for VTN is defined in [I-D.dong-idr-sr-policy-vtn] and similarly, YANG model

which is used to bound SR policy to a specified VTN is defined in this document by:

o SR policy with color is provisioned to the headend; The preferred candidate path is bound to VTN;

o The route with some particular color matchs the SR policy with the corresponding color, which could satisfy the requirement of the route

o Traffic with the route is steered into the SR policy; Packet is encapsulated with the VTN data plane identifier.

3. VTN Yang Module Tree

```
module: ietf-vtn
  +--rw vtn-instance
     +--rw vtn-instance* [vtn-id]
        +--rw vtn-id             uint32
        +--rw vtn-topology
        |  +--rw (vtn-topolgy-type)?
        |     +--:(flex-algo)
        |     |  +--rw flex-algo
        |     |     +--rw flex-algo-id?   uint32
        |     +--:(multi-topology)
        |        +--rw multi-topology-id?   uint32
        +--rw data-plane
        |  +--rw vtn-data-plane-identifier?   uint32
        +--rw steering-policy
           +--rw vtn-color-id?   uint32

  augment /rt:routing/sr-policy:segment-routing/sr-policy:traffic-engineering/
sr-policy:policies/sr-policy:policy/sr-policy:candidate-paths/sr-
policy:candidate-path:
    +--rw vtn-id-sr-policy
       +--rw vtn-id?   uint32

  augment /if:interfaces/if:interface:
    +--rw interface-configuration-for-vtn
       +--rw (vtn-interface-binding-type)?
          +--:(layer-2-sub-interface)
          |  +--rw layer-2-sub-interface
          |     +--rw sub-interface-id?   uint32
          |     +--rw vtn-id?             uint32
          |     +--rw bandwidth?          uint32
          +--:(queue)
             +--rw queue
                +--rw queue-id?    uint32
                +--rw vtn-id?      uint32
                +--rw bandwidth?   uint32

  augment /ni:network-instances/ni:network-instance:
    +--rw vtn-traffic-steering
       +--rw color-index?   uint32
       +--rw vtn-id?        uint32
```

4. VTN Yang Module

```
<CODE BEGINS>
module ietf-vtn {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-vtn";
```

```
  prefix "ietf-vtn";
```

```
  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-routing {
    prefix "rt";
  }

  import ietf-routing-types {
    prefix "rt-types";
  }

  import ietf-yang-types {
    prefix "yang";
  }

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-network-instance {
    prefix "ni";
  }

  import ietf-sr-policy {
    prefix "sr-policy";
  }

  organization "IETF TEAS Working Group";

  contact
    "
         WG Web: <http://tools.ietf.org/wg/teas/>
         WG List:<mailto:teas@ietf.org>

         Editor: Xuesong Geng
                 <mailto:gengxuesong@huawei.com>
     Editor: Zhibo Hu
                 <mailto:huzhibo@huawei.com>
        ";

  description
    "This YANG module defines a data model for VTN(Virtual Transport Network)";

  revision "2021-04-14" {
    description
      "This is the initial version of VTN yang module";
        reference
```

```
          "RFC XXX: YANG Data Model for VTN";
  }

  grouping vtn-instances{
    description
          "VTN instances";
    list vtn-instance {
          key "vtn-id";
      description
        "vtn instance list";
          leaf vtn-id {
            type uint32;
            description
              "vtn-id";
          }
          container vtn-topology {
            description
                  "vtn topology is nt";
                choice vtn-topolgy-type{
                  description
                    "customized topology of VTN";
                  case flex-algo {
                    container flex-algo {
                          description
                            "flex-algo could be used as topology specification
for VTN";
                          leaf flex-algo-id {
                            type uint32;
                            description
                              "flex-algo-id for VTN";
                          }
                        }
                  }
                  case multi-topology {
                    description
                          "MT could be used as topology specification for VTN";
                    leaf multi-topology-id{
                          type uint32;
                          description
                            "MT-id for VTN";
                        }
                  }
                }
          }
      container data-plane {
            description
                  "vtn data plane mechanism";
            leaf vtn-data-plane-identifier {
```

```
        type uint32;
```

```
                  description
                    "VTN identifier of data plane for vtn distinguishment";
            }
          }
          container steering-policy {
            description
                  "Policy set that matches to a VTN";
                leaf vtn-color-id {
                  type uint32;
                  description
                    "VTN color ID for VTN traffic steering";
                }
          }
    }
  }

  grouping interface-configuration-for-vtn{
    description
          "interface configuration for vtn";
    container interface-configuration-for-vtn {
          description
            "interface configuration for vtn";
          choice vtn-interface-binding-type{
        description
                  "vtn interface binding type";
                case layer-2-sub-interface {
                  description
                    "vtn is bound to a layer-2 sub-interface";
                  container layer-2-sub-interface {
                    description
                          "sub-interface configuration";
                        leaf sub-interface-id {
                          type uint32;
                          description
                            "sub-interface id";
                        }
                        leaf vtn-id {
                          type uint32;
                          description
                            "vtn-id";
                        }
                        leaf bandwidth {
                          type uint32;
                          description
                            "bandwidth allocation for the slice";
                        }
                  }
                }
```

```
                case queue {
                  description
                    "vtn is bound to a queue in the interface";
              container queue {
                    description
                          "queue configuration";
                        leaf queue-id {
                          type uint32;
                          description
                            "queue id";
                        }
                        leaf vtn-id {
                          type uint32;
                          description
                            "queue id";
                        }
                        leaf bandwidth {
                          type uint32;
                          description
                            "bandwidth allocation for the slice";
                        }
                  }
                }
          }
        }
  }

  grouping sr-policy-traffic-steering{
    container vtn{
          description
            "candidata path is bound to VTN";
          leaf vtn-id{
            type uint32;
                description
                  "vtn";
          }
        }
  }

  grouping vtn-traffic-steering{
    container vtn-traffic-steering {
          leaf color-index {
            type uint32;
                description
                  "color index";
          }
          leaf vtn-id {
            type uint32;
```

```
                description
                  "vtn id";
          }
        }
  }

  container vtn-instance {
    description
          "vtn instance configuraiton";
    uses vtn-instances;
  }

  grouping vtn-id-sr-policy {
    description
          "VTN ID for SR policy";
        container vtn-id-sr-policy {
          description
            "VTN ID for SR policy";
          leaf vtn-id {
            type uint32;
                description
                  "vtn id";
          }
        }
  }

  augment "/rt:routing/sr-policy:segment-routing/sr-policy:traffic-engineering/
sr-policy:policies/sr-policy:policy/sr-policy:candidate-paths/sr-
policy:candidate-path" {
    description
          "VTN ID for candidate path in SR path";
        uses vtn-id-sr-policy;
  }

  augment "/if:interfaces/if:interface" {
    description
          "interface model extension for vtn";
        uses interface-configuration-for-vtn;
  }

  augment /ni:network-instances/ni:network-instance{
    description
          "network instance model extension for vtn";
    uses vtn-traffic-steering;
  }
}
<CODE ENDS>
```

5. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

6. Contributor

Zhenbin Li
Huawei

Email: lizhenbin@huawei.com

Jie Dong
Huawei

Email: jie.dong@huawei.com

7. References

7.1. Normative References

[I-D.dong-6man-enhanced-vpn-vtn-id]
Dong, J., Li, Z., Xie, C., and C. Ma, "Carrying Virtual Transport Network Identifier in IPv6 Extension Header", draft-dong-6man-enhanced-vpn-vtn-id-03 (work in progress), February 2021.

[I-D.dong-idr-sr-policy-vtn]
Dong, J., Hu, Z., and R. Pang, "BGP SR Policy Extensions for Virtual Transport Network", draft-dong-idr-sr-policy-vtn-00 (work in progress), October 2020.

[I-D.dong-teas-enhanced-vpn-vtn-scalability]
Dong, J., Li, Z., Qin, F., Yang, G., and J. N. Guichard, "Scalability Considerations for Enhanced VPN (VPN+)", draft-dong-teas-enhanced-vpn-vtn-scalability-02 (work in progress), February 2021.

[I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-15 (work in progress), April 2021.

[I-D.ietf-lsr-isis-sr-vtn-mt]
Xie, C., Ma, C., Dong, J., and Z. Li, "Using IS-IS Multi-Topology (MT) for Segment Routing based Virtual Transport Network", draft-ietf-lsr-isis-sr-vtn-mt-00 (work in progress), March 2021.

[I-D.ietf-spring-segment-routing-policy]
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-11 (work in progress), April 2021.

[I-D.ietf-teas-enhanced-vpn]
Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Network (VPN+) Services", draft-ietf-teas-enhanced-vpn-07 (work in progress), February 2021.

[I-D.ietf-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Definition of IETF Network Slices", draft-ietf-teas-ietf-network-slice-definition-01 (work in progress), February 2021.

[I-D.zhu-lsr-isis-sr-vtn-flexalgo]
Zhu, Y., Dong, J., and Z. Hu, "Using Flex-Algo for Segment Routing based VTN", draft-zhu-lsr-isis-sr-vtn-flexalgo-02 (work in progress), February 2021.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <https://www.rfc-editor.org/info/rfc4915>.

[RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <https://www.rfc-editor.org/info/rfc5120>.

7.2. Informative References

[InfRef]   , 2004.

Appendix A. An Appendix

Authors' Addresses

Xuesong Geng
Huawei

Email: gengxuesong@huawei.com

Zhibo Hu
Huawei

Email: huzhibo@huawei.com