Network Working Group
Internet Draft

Intended status: Informational
Expires: April 2014

Young Lee
Huawei Technologies

Greg Bernstein
Grotto Networking

Ning So
Tata Communications

Luyuan Fang
Cisco

Daniele Ceccarelli
Ericsson

Diego Lopez
Telefonica

Oscar Gonzalez de Dios
Telefonica

October 21, 2013

Network Control Function Virtualization for Transport SDN

draft-lee-network-control-function-virtualization-01.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

Abstract

This presentation explores the concept of network control function virtualization for transport SDN to help evolve transport networks to provide programmable virtual network services with infrastructure changes to the traditional control plane architecture.

Table of Contents

1. Terminology..3
2. Requirements Language..3
3. Introduction...3
4. Transport SDN Control Architecture.............................4
5. Transport SDN Virtual Network Service..........................6
6. Summary and Conclusion..11
7. References..11
   7.1. Informative References...................................11
8. Contributors..12
Authors' Addresses...12
Intellectual Property Statement..................................12
Disclaimer of Validity...13

1. Terminology

This document uses the terminology defined in [RFC4655], and [RFC5440].

CVI   Client-VNC Interface

PNC   Physical Network Control

VL   virtual Link

VNC   Virtual Network Control

VNE   Virtual Network Element

VNS   Virtual Network Service

VPI   VNC-PNC Interface

2. 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 [RFC2119].

3. Introduction

One of the main drivers for SDN is a physical separation of the network control plane from the forwarding plane. This separation of the network control plane from the forwarding plane has been already achieved with the development of GMPLS/ASON and PCE for transport networks. In fact, in transport networks such separation of data and control plane was dictated at the onset due to the very different natures of the data plane (circuit switched TDM or wavelength) and a packet switched control plane. The physical separation of the control plane and the forwarding plane is a major step toward allowing operators to gain the full control for optimized network design and operation. Another attraction of SDN technology is its logically centralized control regime which allows a global view of the underlying networks under its control. The centralized control of SDN helps improve network resource utilization from a distributed network control. Transport networks have long used this centralized model via network management systems and currently supplement it with topology and resource status information gathered dynamically via GMPLS routing. For transport network control, PCE technology is

essentially equivalent to a logically centralized control for path computation function. By combining the strength of GMPLS/ASON [GMPLS] and PCE [PCE], and open standard interfaces, transport network control plane technology is readily in a position to fully embrace the SDN concepts.

However, the current transport network control plane technology is not suitable for virtualization and client programmability, which are the two of the main drivers for SDN in recent years. Virtualization refers to allowing the clients to utilize a certain amount of network resources as if they own them and thus control their allocated resources in a way most optimal with higher layer or application processes. This empowerment of client control facilitates introduction of new services and applications as the clients are given to create, modify, and delete their virtual network services. The level of virtual control given to the clients can vary from a tunnel connecting two end-points to virtual network elements that consist of a set of virtual nodes and virtual links in a mesh network topology. As part of the VNS, a client control concept is added to the traditional VPN along with a client specific virtual network view. Client control is operated on the view of virtual network resources allocated to the client. This view is called abstracted network topology. Such a view may be specific to the set of client services as well as the particular client. As the client control is envisioned to support a plethora of applications, there is another level of virtualization from the client to individual applications.

4. Transport SDN Control Architecture

To allow virtualization, the network has to provide open, programmable interfaces in which clients/applications can create, replace, modify virtual network services in an interactive manner while having no impact on other network clients. Traditional transport network infrastructure is not suitable for providing programmable interfaces to clients. Direct client control of transport network elements over existing programmable interfaces (control or management plane) is not perceived as a viable proposition for transport network providers due to security and policy concerns among other reasons.

Hence, the need for network control function virtualization where there is a "virtualizer" that interfaces directly with client control and translates/allocates resources (from physical to virtual and vice versa) and creates abstracted network topology for each client and interacts with physical network connection control functions (e.g., GMPLS/ASON for provisioning, PCE for path

computation). The Network control function virtualizer maintains two interfaces: one interface with physical network control functions assumed by GMPLS/ASON and PCE, which is termed as VNC-PNC Interface (VPI); another interface with client control of virtual network, which is termed as Client-VNC Interface (CVI). Figure 1 depicts the overall architecture for transport SDN control in which the virtual network control entity provides network control function virtualization.

```
              ------------------------------------------
             |             Application Layer            |
              ------------------------------------------
                    /|\       /|\          /|\
                     |         |           \|/   North Bound API
                     |         |     ---------------
                     |        \|/   |       Client  |
                     |       -------------- Control |
                    \|/     |      Client  |--------
                    -------------- Control |   /|\
                   |    Client    |-------      |
                   |    Control   |   /|\       |
                    --------------     |        |
                           /|\         |        |  Client-VNC Interface
                            |          |        |   (CVI)
                           \|/        \|/      \|/
                    ----------------------------------
                   |  Virtual Network Control (VNC)   |
                    ----------------------------------
                                     /|\
                                      |     VNC-PNC Interface (VPI)
                                     \|/
                    ----------------------------------
                   | Physical Network Control (PNC)   |
                    ----------------------------------
                                     /|\
                                      |     Control Interface to NEs
                                     \|/

                        Physical Network Infrastructure
```

Figure 1: Transport SDN control architecture

Figure 1 shows that there are multiple client controls which are independent to each other and that each client supports various business applications over its NB API. There are layered client-server relationships in this architecture. As various applications are clients to client control, client control itself is also a client to virtual network control; likewise, virtual network control is also a client to physical network control. This layered relationship is important in protocol definition work on NB API, CVI and VPI interfaces as this allows third-party software developers to program client control and virtual network control functions in such a way to create, modify and delete virtual network services.

This architecture in Figure 1 is conceptually in alignment with the Network Functions Virtualization (NFV) architecture [NFV-AF].

5. Transport SDN Virtual Network Service

Virtual Network Service is instantiated by the client control via the CVI. As client control directly interfaces the application stratum, it understands multiple application requirements and their service needs. It is assumed that client control and VNC have the common knowledge on the end-point interfaces based on their business negotiation prior to service instantiation. End-point interfaces refer to client-network physical interfaces that connect client premise equipment to network provider equipment. Figure 2 shows an example physical network topology that supports multiple clients. In this example, client A has three end-points A.1, A.2 and A.3. The interfaces between clients and transport networks are assumed to be 40G OTU links. For simplicity's sake, all network interfaces are assumed to be 40G OTU links and all network ports support ODU switching and grooming on the level of ODU1 and ODU2. Client control for A provides its traffic demand matrix that describes bandwidth requirements and other optional QoS parameters (e.g., latency, diversity requirement, etc.) for each pair of end-point connections.

Figure 2 shows that three independent clients A, B and C provide its respective traffic demand matrices to the VNC. The physical network topology shown in Figure 2 is the provider's network topology created by the PNC's topology creation engine such as the link state database (LSDB) and Traffic Engineering DB (TEDB) based on control plane discovery function. This topology is internal to PNC and not available to the client. What is available to the client is abstracted network topology (aka, virtual network topology) based on the negotiated level of abstraction. This is a part of VNS instantiation between a client control and VNC.

```
             +------+           +------+          +------+
   A.1 ------o      o-----------o      o----------o      o------- A.2
   B.1 ------o   1  |           |   2  |          |   3  |
   C.1 ------o      o-----------o      o----------o      o------- B.2
             +-o--o-+           +-o--o-+          +-o--o-+
               |  |               |  |              |  |
               |  |               |  |              |  |
               |  |               |  |              |  |
               |  |             +-o--o-+          +-o--o-+
               |  `-------------o      o----------o      o------- B.3
               |                |   4  |          |   5  |
               `----------------o      o----------o      o------- C.3
                                +-o--o-+          +------+
                                  |  |
                                  |  |
                                  |  |
                                C.2  A.3

       Traffic Matrix           Traffic Matrix           Traffic Matrix
       for Client A             for Client B             for Client C

         A.1  A.2  A.3            B.1  B.2  B.3           C.1  C.2  C.3
    -------------------      ------------------       -----------------
    A.1  -    20G  20G       B.1  -    40G  40G       C.1 -    20G  20G
    A.2  20G   -   10G       B.2  40G   -   20G       C.2 20G   -   10G
    A.3  20G  10G   -        B.3  40G  20G   -        C.3 20G  10G   -
```

Figure 2: Physical network topology shared with multiple clients

Figure 3 depicts illustrative examples of different level of topology abstractions that can be provided by the VNC topology abstraction engine based on physical topology base maintained by the PNC. The level of topology abstraction is expressed in terms of the number of virtual network elements (VNEs) and virtual links (VLs). For example, the abstracted topology for client A shows there are 5 VNEs and 10 VLs. This is by far the most detail topology abstraction with a minimal link hiding compared to other abstracted topologies in Figure 3.

```
           Abstracted Topology for Client A (5 VNEs and 10 VLs)

             +------+           +------+          +------+
   A.1 ------o      o-----------o      o----------o      o------- A.2
             |   1  |           |   2  |          |   3  |
             |      |           |      |          |      |
             +-o----+           +-o----+          +-o----+
               |                  |                 |
               |                  |                 |
               |                  |                 |
               |                +-o----+          +-o--o-+
               |                |      |          |      |
               |                |   4  |          |   5  |
               `----------------o      o----------o      |
                                +----o-+          +------+
                                     |
                                     |
                                     |
                                    A.3


           Abstracted Topology for Client B (3 VNEs and 6 VLs)

             +------+                             +------+
   B.1 ------o      o-----------------------------o      o------ B.2
             |   1  |                             |   3  |
             |      |                             |      |
             +-o----+                             +-o----+
                \                                    |
                 \                                   |
                  \                                  |
                   `-------------------              |
                                       `          +-o----+
                                        \         |      o------ B.3
                                         \        |   5  |
                                          `-------o      |
                                                  +------+
```

```
            Abstracted Topology for Client C (1 VNE and 3 VLs)


             +--------------------------------------------+
             |                                            |
             |                                            |
   C.1 ------o                                            |
             |                                            |
             |                                            |
             |                                            |
             |                                            o--------C.3
             |                                            |
             +--------------------o-----------------------+
                                  |
                                  |
                                  |
                                  |
                                 C.2
```

Figure 3: Topology Abstraction Examples for Clients

As different client has different control/application needs, abstracted topologies for client B and C, respectively show much less degree of abstraction. The level of topology abstraction is determined by the policy (e.g., the granularity level) placed for the client and/or the path computation results by the PNC's PCE. The more granular the abstraction topology is, the more control is given to the client control. If the client controller has applications that require more granular control of virtual network resources, then abstracted topology for client A may be the right abstraction level for such client controller. For instance, if the client is a third-party virtual service broker/provider, then it would desire much more sophisticated control of virtual network resources to support differing application needs. On the other hand, if the client were only to support simple tunnel services to its application, then abstracted topology for client C that consists of one VNE and three VLs would suffice.

Figure 4 shows workflows across client control, VNC and PNC for the VNS instantiation, topology exchange, and VNS setup. Client control "owns" a VNS and initiates by providing the instantiation identifier with the traffic demand matrix with path selection constraints for that instance. This VNS instantiation request from Client Control triggers a path computation request by the virtual PCE (vPCE) agent in the VNC after VNC's proxy's interlay of this request to the vPCE. vPCE requests a concurrent path computation request that is converted based on the traffic demand matrix as part of the VNS instantiation request from Client Control. Upon receipt of this path computation request, the PCE in the PNC block computes paths and updates network topology DB and informs the vPCE agent of the VNC of the paths and topology updates.

```
 ------------------------------------------------------------------
| Client    ----------------------------------------------        |
| Control  |               VNS Control                    |       |
|           ----------------------------------------------        |
 ------------------------------------------------------------------
1.VNS            |     /|\  4. Abstracted           |     /|\
Instantiation    |      |   Topology                |      |
(instance id,    |      |                           |      |
 Traffic Matr.)  |      |                           |      | 8. VNS
                 |      |                  5. VNS   |      | Set-up
                \|/     |                  Set-up  \|/     | Confirm
 ------------------------------------------------------------------
| Virtual   ----------------------------------------------        |
| Network  |               VNS Proxy                      |       |
| Control   ----------------------------------------------        |
|           -----------------------     -----------------------   |
|          |      vPCE Agent       |   |  vConnection Agent    |  |
|           -----------------------     -----------------------   |
 ------------------------------------------------------------------
2. Path          |     /|\  3. PC Reply             |     /|\
Computation      |      |   with updated            |      |
Request          |      |   topology                |      |
                 |      |             6. Network    |      |8.Network
                 |      |             Provisioning  |      |Provisioning
                \|/     |             Request      \|/     |Confirm
 ------------------------------------------------------------------
| Physical       -------------           -------------------------- |
| Network       |     PCE     |         |  Network Provisioning    ||
| Control        -------------           -------------------------- |
 ------------------------------------------------------------------
```

Figure 4. Workflows across Client control, VNC and PNC

It is assumed that the PCE in PNC is a stateful PCE [PCE-S]. vPCE agent abstracts the network topology into an abstracted topology for the client based on the agree-upon granularity level. The abstracted

topology is then passed to the VNS control of the Client Control block. The Client Control's VNS control computes and assigns virtual network resources for its applications based on the abstracted topology and creates VNS setup command to the VNC. The VNC's vConnection module turns this VN setup command into network provisioning requests over the network elements using control plane messages such as GMPLS, etc.

6. Summary and Conclusion

This presentation explores the concept of network control function virtualization for transport SDN to help evolve transport networks to provide programmable virtual network services with infrastructure changes to the traditional control plane architecture. The VNC and its interfaces with the Client Control and the PNC provide control plane function virtualization over programmable interfaces such as virtual network path computation and optimization, topology abstraction hiding details of physical topology while supporting service-specific objectives the clients demand, maintaining virtual network service instances and the states, policy enforcement for virtual network services. With this evolutionary architecture, virtual network services can be readily introduced while re-using physical network control plane functions.

7. References

7.1. Informative References

[PCE]   Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", IETF RFC 4655, August 2006.

[PCE-S]   Crabbe, E, et. al., "PCEP extension for stateful PCE",draft-ietf-pce-stateful-pce, work in progress.

[GMPLS]   Manning, E., et al., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, October 2004.

[NFV-AF] "Network Functions Virtualization (NFV); Architectural Framework", ETSI GS NFV 002 v1.1.1, October 2013.

8. Contributors

Authors' Addresses

Young Lee
Huawei Technologies
5340 Legacy Drive
Plano, TX 75023, USA
Phone: (469)277-5838
Email: leeyoung@huawei.com

Greg Bernstein
Grotto Networking
Fremont, CA, USA
Phone: (510) 573-2237
Email: gregb@grotto-networking.com

Ning So
Email: Ning.So@tatacommunications.com

Luyuan Fang
Email: luyuanf@gmail.com

Daniel Ceccarelli
Email: daniele.ceccarelli@ericsson.com

Diego Lopez
Email: diego@tid.es

Oscar Gonzalez de Dios
Email: ogondio@tid.es

Intellectual Property Statement

Disclaimer of Validity

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.