PCE Working Group
Internet Draft
Intended status: Informational

Young Lee
Xian Zhang
Haomian Zheng
Huawei
Guoying Zhang
CATR

Expires: August 14, 2014
February 14, 2014

Application-oriented Stateful PCE Architecture and Use-cases for Transport Networks

draft-lee-pce-app-oriented-arch-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.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on August 14, 2014.

Abstract

This draft presents an application-oriented stateful PCE architecture for transport networks. Under this architecture, several use cases are described.

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].

Table of Contents

1. Introduction

With the emerging applications requiring large bandwidth and dynamic provisioning, such as Data Center Interconnection(DCI), cloud bursting and so on, the traditional transport network architecture is limited as it only provides "dumb pipe" services. These services lack the flexibility for operation and management. In order to support the demands, including large bandwidth, low service latency as well as dynamic and flexible resource allocation, transport networks may need to be enhanced architecturally such that it could be aware of application requirements in a dynamic fashion. The Path Computation Elements (PCE) architecture and the corresponding protocol extensions provide a mechanism that enables path computation for transport network. As specified in [RFC4655], a PCE supports the request for path computation issued by a Path Computation Client (PCC). When the PCC is external to the PCE, a communication protocol, i.e., PCE Protocol (PCEP), is required to support the path computation request/reply process. Furthermore, extensions to PCEP are proposed in [PCE-S] , [PCE-I], and [PCE-S-GMPLS] to enable stateful control over networks including transport networks.

This draft provides an application-oriented stateful PCE architecture for transport networks. In particular, this architecture introduces transport network controller (TNC) component in which transport PCE plays a central role. Given the high demands from applications, an interface between the transport network controller and the application client controller is also introduced to enable the communication function between these entities. The application client controller is a special type of PCC with respect to PCE capability within the transport network controller. This interface and its communication mechanism between the application client controller and the transport network controller enables operation of the transport network with more flexibility. Specifically, in a larger-scale transport network with multiple layers or multiple domains, the communication mechanism between different PCEs and the application client controllers is very important to satisfy the request from the application stratum. Current PCEP can provide communication between PCE and PCCs, and further extensions to PCEP may be desirable to cooperate with new types of PCCs such as application client controllers.

2. Terminology

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. Architecture and Key Features

In this draft, a PCE-centric architecture which supports application-oriented transport network is defined. The architecture is illustrated in Figure 1. The functions of each architectural component are described. And then interfaces between the stateful PCE and the other functional blocks in the transport network are defined.

```
                                     ---------------
                                    |       Client  |
                             -------------- Control |
                            |      Client  |--------
                    -------------- Control |   /|\
                   |    Client    |-------      |
                   |    Control   |   /|\       |
                    --------------     |        |
                           /|\         |        |  Client-VNC Interface
                            |          |        |   (CVI)
                           \|/        \|/      \|/
                    ----------------------------------
               /   |  Virtual Network Control (VNC)   |
    Transport  |    ----------------------------------
    Network    |                      /|\
    Controller |                       |     VNC-PCE Interface (VPI)
      (TNC)    |                      \|/
               |    ----------------------------------
               \   |          Transport PCE           |
                    ----------------------------------
                                     /|\
                                      |     PCEP
                                     \|/

                        Physical Network Infrastructure
```

Figure 1: Application-oriented PCE Architecture for Transport Network

Transport Network Controller (TNC) in Figure 1 is the core of the application-oriented PCE architecture for transport network. It is built around the Transport PCE and provides additional functions that facilitate multi-layer control, virtual network service control and other functionalities such as topology abstraction via the Virtual Network Control (VNC) block. The VNC interfaces can be different types of client controllers, such as packet network controllers, data center provider controllers, enterprise network controllers, virtual service provider controllers, etc. The VNC provides network control function virtualization to the PCE and to the clients via the VNC-PCE Interface (VPI) and the Client-VNC

Interface (CVI), respectively. The VNC allows the clients (via their client controllers) to program their client-defined virtual network services (VNS) over the CVI. The VNC also provides abstract network topology for each client based on the network resources allocated to the client. In order to facilitate this capability, the VNC needs to communicate with the PCE via the VPI interface. It is worth noting that the CVI can be an internal interface with respect to the TNC, or be an external interface from the perspective of transport PCE, according to the application. In this draft, it is assumed that VPI is an external interface from the PCE. The VNC is considered as a PCC to the PCE.

The VNC provides 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. See [NCFV] for details of control function virtualization concept. With this evolutionary architecture built on top of transport PCE, a number of challenging use-cases can be supported. Transport PCE is a stateful PCE and supports all the generic stateful PCE functions as described in [PCE-S] and [PCE-S-GMPLS].

The CVI is an external interface with respect to the transport network controller (TNC). Client controller is an external client. Figure 1 shows that there are multiple client controls which are independent to each other and that each client supports various business applications. There could be multiple recursive layered client-server relationships in this architecture. For example, 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. In each relationship, client can be considered as PCC that requesting service from server which can be considered as PCE. It is worth noting that one client or PCC can be connected with multiple servers and vice versa. Such layered relationship is important in protocol definition work on 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.

4. Use-cases

This section provides a number of use-cases to which the architecture discussed in Section 3 is applied.

4.1. Dynamic Data Center Network Interconnection

In the context of multiple data center networks where there is a need to move large data dynamically from one location to other location(s), data center network controller is a type of client controller that coordinates with the virtual network controller (VNC). This coordination across data center client controller and the VNC allows multiple instances of inter data center connections need for different applications. In this case there are multiple client-server pairs. Each of the data center can be a client that request resources from the data center network controller, which is considered as a server for service provisioning. The data center network controller is then a client that requesting services from transport server. The two client pairs are slightly different in this use case. The first pair is communicating on virtual resources, while the second pair focuses on allocating physical resources. From an extended PCE architecture point of view, client-server can be regarded as PCC-PCE.

For each virtual resource request from client or application, the VNC keeps the instance and creates an abstracted network topology based on the network resources allocated to a particular request. The data center client controller has the view of this abstracted network topology and is given a full control of how to use the allocated virtual resources.

The topology abstraction created by the VNC for the client is based on the transport PCE's physical network resource information and is needed to be filtered via the VNC's filtering mechanism based on contract, policy and security. In this way, physical resources are abstracted into virtual resources.

The VNC interlays client control's request for inter data center connection and converts into a PC request to the PCE. Then a PCE instantiates a network path via its provisioning mechanism described in [PCE-I].

4.2. Packet-Optical Integration (POI)

Client controller can also be a router network controller that needs transport network interconnections. The router network controller can request different connection services from the transport network based on different QoS needs.

Note that this POI use-case is different from multi-layer PCE work [RFC5623] in that it allows more flexible interactions and more

granular level of abstracted network topologies than tunnel-based virtual network topology.

4.3. Virtual Network Service (VNS)

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. The different level of topology abstractions 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). As different client has different control/application needs, abstracted topologies for a certain client can 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 finer granularity the abstraction topology is, the more control is given to the client control. 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 such client is a simple abstracted topology with a set of end-point tunnels.

Synchronization is an important issue in layered path computation procedure. As we assumed, the transport PCE is a stateful PCE with TED and LSPD. Meanwhile the VNC also needs to maintain databases that contain virtual topology information, including a virtual TED and virtual LSPD. The information in these databases is obtained from the transport PCE databases, with proper mapping scheme. The mapping mechanism is out of scope for this draft.

4.4. Time-based Scheduling

Transport services with time constraints are another highly-demanded task in the network. In this scenario, a client controller can

   request to reserve some bandwidth for future use. This 'time-based' service needs to be considered together with the traffic Engineering Database (TED) and Label Switched Path Database (LSPD). PCE will compute the scheduled network resource for this 'time-based' service, and reserve such resources for future use.

In this scenario, the LSPD contains two categories of LSP information, current LSP in use and scheduled LSP. These two groups of LSP can be included in a single LSPD or two separate ones, with internal interface to PCE. PCEP should also be extended to include the scheduled information for service requests, such as proposed in [Time-based]. With these extensions, the PCC (for example, application stratum) can generate the path computation request.

4.5. Multi-vendor Interoperation

PCE orchestration is essential in multi-vendor scenario. VNC can be connected with PCEs from more than one vendor to orchestrate the path computation. For simplicity, in this case we assume a 'two domains with two vendors', i.e., each vendor has a PCE within their respective domain, as shown in Fig. 2.

```
   +--------------+              +--------------+
   |    Client    |              |    Client    |
   | Controller A |              | Controller B |
   +--------------+              +--------------+
          /|\                           /|\
           |                             |
          \|/                           \|/
   +--------------------------------------------+
   |        Virtual Network Controller          |
   |         +----------------------+           |
   |         |     Orchestrator     |           |
   |         +----------------------+           |
   +--------------------------------------------+
          /|\                           /|\
           |                             |
          \|/                           \|/
     +-----------+                 +-----------+
     | Transport |                 | Transport |
     |   PCE A   |                 |   PCE B   |
     +-----------+                 +-----------+
          /|\                           /|\
           |                             |
```

```
          \|/                            \|/
      ///------\\\                   //-------\\
   |||   Vendor   |||            |||   Vendor  |||
    ||     A      ||              ||     B     ||
      \\\------///                  \\-------//
                 Fig. 2 Architecture for Interoperation
```

The original path computation request comes from one of the client controllers, with multiple vendors involved. This request is sent to VNC, which will then categorize the request into a 'multi-vendor' class. Before allocating virtual resources, the VNC will determine the domain path and decompose the path computation request from client controller, and then send PCReq to two transport PCEs respectively. The path will be replied to VNC after computation, and corresponding databases are updated in VNC after establishing the path.

It is worth noting that the architecture in multi-vendor use case is quite similar to the Hierarchical PCE [H-PCE] but slightly different. The PCEs belong to different vendors and the VNC may play as a parent PCE by a service provider, for example, operators.

5. References

5.1. Informative References

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[RFC5440] Vasseur, J. P. and J. L. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

[RFC5623] Oki, E., Takeda, T., et al, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, September 2009.

[PCE-S] Crabbe, E., Medved, J., Minei, I., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce, work in progress. [PCE-I] Crabbe, E., Minei, I., Sivabalan, S., and Varga, R., "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-crabbe-pce-pce-initiated-lsp, work in progress.

[PCE-S-GMPLS] Zhang, X., Lee, Y., Casellas, R., Gonzalez de Dios, O., "Path Computation Element (PCE) Protocol Extensions for Stateful PCE Usage in GMPLS-controlled Networks", draft-zhang-pce-pcep-stateful-pce-gmpls-03.txt, work in progress.

[NCFV] Lee, Y. Bernstein, G., So, N., Fang L., and Ceccarelli, D. "Network Control Function Virtualization for Transport SDN", draft-lee-network-control-function-virtualization, work in progress.

[H-PCE] Zhang, F., Zhao, Q., Gonzalez de Dios, O., Casellas R., King D., Extensions to Path Computation Element Communication Protocol (PCEP) for Hierarchical Path Computation Elements (PCE), draft-zhang-pce-hierarchy-extensions-04, work in progress.

[Time-based] Zhang, X., Lee, Y., Casellas, R., Ganzalez, O., "Stateful Path Computation Element (PCE) for Time-based Scheduling", draft-zhang-pce-stateful-time-based-scheduling-00, work in progress.

6. Authors' Addresses

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

Xian Zhang
Huawei Technologies
Email: zhang.xian@huawei.com

Haomian Zheng
Huawei Technologies
Email: Zhenghaomian@huawei.com

Guoying Zhang
China Academy of Telecommunication Research of MII
11 Yue Tan Nan Jie Beijing, P.R.China
Phone: +86-10-68094272
Email: zhangguoying@mail.ritt.com.cn

7. Acknowledgment

Intellectual Property

The IETF Trust takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in any IETF Document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights.

Copies of Intellectual Property disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement any standard or specification contained in an IETF Document. Please address the information to the IETF at ietf-ipr@ietf.org.

The definitive version of an IETF Document is that published by, or under the auspices of, the IETF. Versions of IETF Documents that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of IETF Documents. The definitive version of these Legal Provisions is that published by, or under the auspices of, the IETF. Versions of these Legal Provisions that are published by third parties, including those that are translated into other languages, should not be considered to be definitive versions of these Legal Provisions.

For the avoidance of doubt, each Contributor to the IETF Standards Process licenses each Contribution that he or she makes as part of the IETF Standards Process to the IETF Trust pursuant to the provisions of RFC 5378. No language to the contrary, or terms, conditions or rights that differ from or are inconsistent with the rights and licenses granted under RFC 5378, shall have any effect and shall be null and void, whether published or posted by such Contributor, or included with or in such Contribution.

Disclaimer of Validity

All IETF Documents and the information contained therein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.