

Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 20, 2013

M. Chen
M. McBride
Huawei Technologies Co., Ltd
L. Ou
China Telecom
January 16, 2013

Software Defined Networks Use Case for Virtual Connection and Network on Demand

[draft-mm-sdn-vc-vn-on-demand-use-case-02](#)

Abstract

Software Defined Networks (SDN) provide a way to virtualize and abstract the network and presents the virtual/abstract resources to the third-party applications/software. The applications/software can, by the programmable interface or Northbound API, request, manipulate and monitor the services that the network provided. With this SDN capability, more innovative services can be introduced and rapidly deployed. Additionally, existing services can be deployed and managed in a much more simple way.

This document outlines two use cases that leverage the SDN architecture/model to implement a kind of "self-help" and automated set of network services: Virtual Connection on Demand (VCoD) and Virtual Network on Demand (VNoD).

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](#)].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

Internet-Draft

SDN Use Case for VC/VN on Demand

January 2013

This Internet-Draft will expire on July 20, 2013.

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.

Internet-Draft

SDN Use Case for VC/VN on Demand

January 2013

Table of Contents

1.	Introduction	4
2.	Use Cases	5
2.1.	Reference Architecture	5
2.2.	VC on Demand	6
2.3.	VN on Demand	7
3.	IANA Considerations	9
4.	Security Considerations	9
5.	Acknowledgements	9
6.	Normative References	9
	Authors' Addresses	9

1. Introduction

At present, applications and networks are independent of each other, and there is almost no interaction between the two layers. The applications normally have no idea about how the application data be transmitted over the network nor the status of the network. The application requirements and constraints to the network are passed to service providers by some bill forms or contracts, and then the network operators set up the network according to the bill forms or contracts. These processes will take a considerably long time to take effect and are error-prone. If there are some new requirements and updates, the processes will be re-done. This network service architecture and model is increasingly unable to meet the needs of modern applications (e.g., Cloud service, Data Center, etc.), which may frequently request to create/delete network connections, adjust the existing paths or choose the desired paths per performance characteristic (e.g., delay, loss, jitter, etc.), price or some other conditions. So, a new network service architecture and model is required, and Software Defined Networks (SDNs) are considered to be a set of promising solutions, which opens the network and provides programmable interface (e.g., Northbound API) to the applications and third-party software.

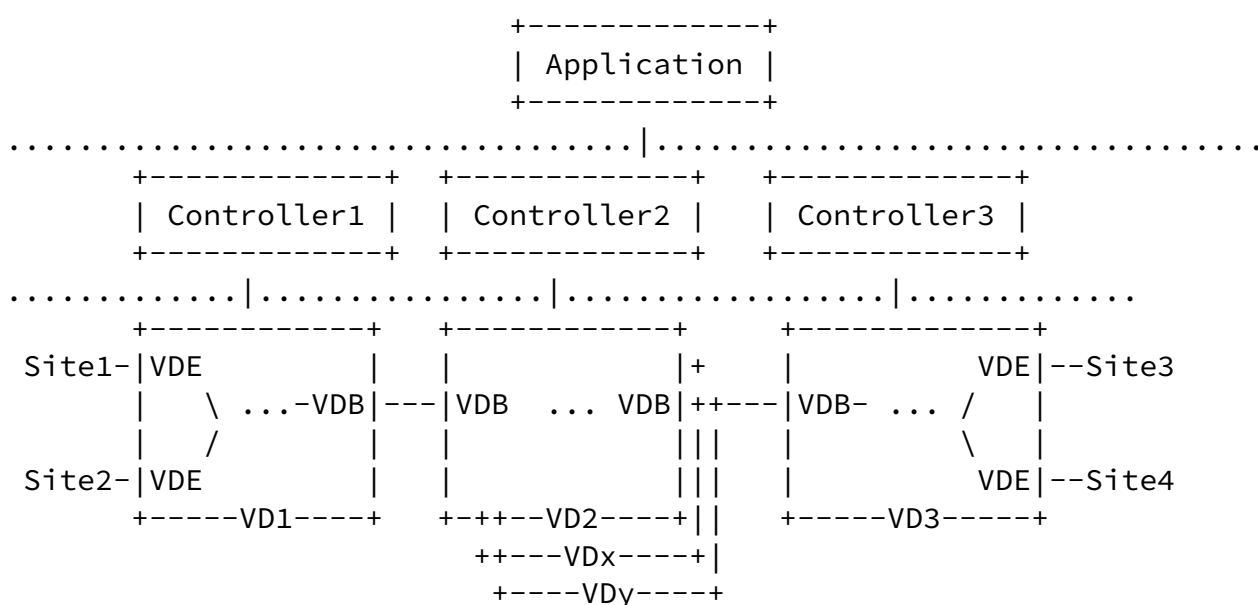
There are several kinds of network services: transport service (Optical, Ethernet, Internet Protocol (IP), Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Label Switched Path (LSP), etc.), Virtual Private Network (VPN) service(VLAN, VxLAN, L2/L3 VPN, etc.), policy related service (e.g., Access Control List (ACL)), security service, etc. This document mainly focuses on the transport

and VPN related services that can be abstracted and categorized into two types: Virtual Connection (VC) and Virtual Network(VN) services. Other services are basically dependent on these two services. For example, when you want to configure an ACL, load balance or security service, you must configure it on a specific connection or network. That is, they are the subordinate services of the connection and network services. The VC or VN is a kind of logical/abstract network services, where the VC can be mapped to a flow (as defined in Openflow), an optical path(a lambda, fiber, etc.), an E-line, a Pseudo Wire (PW), a TE LSP etc., and the VN can be mapped to a VLAN, a VxLAN, an L2/L3 VPN, etc. With the VC and VN concept, it could present a set of uniform abstract services and interfaces to the applications, independent of the underlying network diversity and complexity.

This document describes the use cases which leverage the SDN architecture and model to implement the Virtual Connection on Demand (VCoD) and Virtual Network on Demand (VNoD) services.

2. Use Cases

2.1. Reference Architecture



VD - Virtual Domain

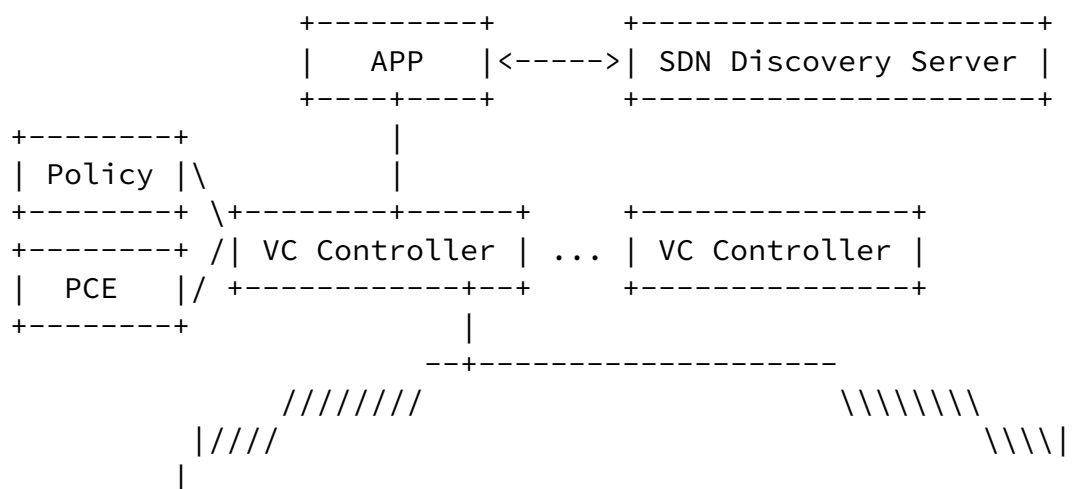
VDE - Virtual Domain Edge
VDB - Virtual Domain Boundary

Figure 1 SDN Reference Architecture

Figure 1 is a SDN reference architecture that consists of 3 tiers, application, controller and the physical network. For a large scale network, there normally should be multiple controllers and each of them is responsible for a specific domain of the network. Here, we introduce a concept of Virtual Domain (VD) that provides a way to abstract and split the physical network into controllable domains, where each domain has its own controller. Each VD has its own Virtual Domain Edges (VDE) that are the service access points for the VD, and the Virtual Domain Boundaries (VDB) are the boundaries between VDs.

There are many ways to split and plan the domains, for example, it could simply be based on the natural routing domains(e.g., per an area or AS) to split and form the VD . And it also could be based on the service to split the domain, meaning the same physical network could be abstracted to different VDs and each providing different services; e.g., an IP/MPLS metro network could be abstracted to an L2 VPN VD and an L3 VPN VD simultaneously.

[2.2.](#) VC on Demand



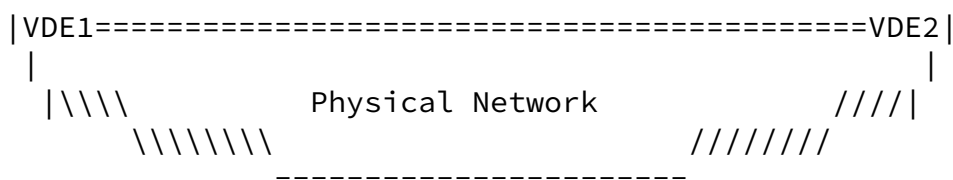


Figure 2 VC on Demand

Figure 2 illustrates the flow, involved with requesting a new Virtual Connection, which includes the following steps:

1. Service Discovery

First, the application needs to discover the VC controller and its capabilities and services. Typically, there will be multiple controllers. The locations, capabilities, and supplied services of the controllers may change over time. The SDN discovery server could present a relevant fixed access location to the application and in order to help make the model more robust.

The application sends a query message to the SDN discovery Server (similar to ALTO server discovery) to retrieve the candidate controllers and their relevant locations, capabilities, services, etc. The App needs to determine which controller is the correct one to use.

2. VC Request/Response

When the controller is determined, the application sends a VC request to the VC Controller to Create/Delete/Modify/Query a VC (e.g., request a new VC from VDE1 to VDE2). The VC controller could choose to respond to the application immediately, or wait for the VC to be successfully constructed and then respond to the application.

3. VC Path Compute

According to the requirements and constraints (e.g., bandwidth, delay, loss etc.) from the App, the VC controller needs to determine a specific network service to serve the VC request, map the VC to a specific network path (e.g., mapping the VC to an openflow-based flow/an E-line/an E-tree/a PW/a TE LSP...) and then compute/find the path. VC controller could finish the path computation by itself (for

example, the controller has an embedded path computation engine), or it may send a path compute request to another entity(e.g., Path Computation Element (PCE)) to finish the path computation.

4. VC Provisioning

When the VC path is determined, the VC controller will then send VC provisioning request to the related network devices to provision the path. For example, the request could be to install an entry in the openflow table, or trigger the ingress PE to signal a TE LSP or PW, etc. The communication channel between VC controller and the network devices could be either the existing protocols (e.g., Openflow, Netconf, Command Line Interface(CLI), etc.) or new APIs defined in the future.

5. VC Monitoring

Once a VC is established, the application may require the ability to monitor the status of the VC. The application data could be directed, according to the VC status, to switchover the data to a backup path when the master path is broken, or to switchover the data to a more cost-effective path, etc.

6. VC Policy

TBD

[2.3.](#) VN on Demand

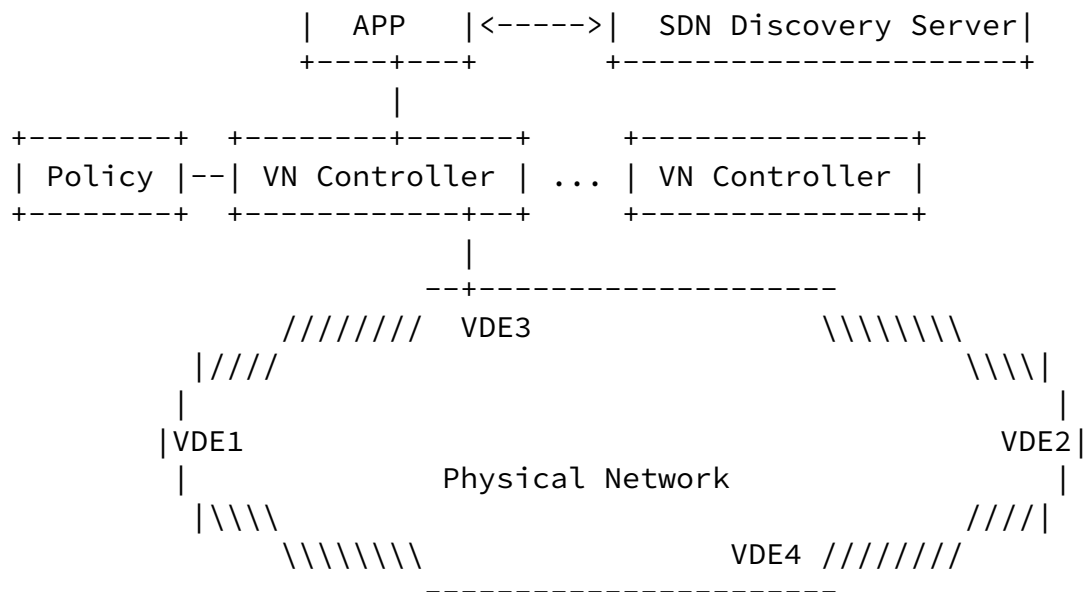


Figure 3 VN on Demand

Figure 3 illustrates the flow, involved with requesting a new Virtual Network:

1. Service Discovery

Similar to VC on Demand, the application first needs to find where the VN controller is located and the capabilities and services that the controller can provide. Then it can be determined who the proper controller is according to response from the SDN discovery server.

2. VN Request/Response

Once the controller is determined, the application sends a VN request to the VN Controller to Create/Delete/Modify/Query a VN. The VN controller could choose to respond to the application immediately, or wait for the VN to be successfully constructed and then respond to the application.

3. VN Orchestration

According to the requirements and constraints from the application, the VN controller needs to determine a specific network service to serve the VN request, map the VN to one or several specific physical or logical networks (e.g., mapping the VN to a VLAN/VxLAN/L2VPN/L3VPN, or a combination of them) and then finish the VN computation/orchestration.

4. VN Provisioning

When the VN orchestration is completed, the VN controller will then send a VN provisioning request to the related network devices to provision the VN.

5. VN Monitoring

Once a VN is constructed, the application may require the ability to monitor the status of the VN where it could steer the application data according to the VN status.

6. VN Policy

TBD

[3.](#) IANA Considerations

This document makes no request of IANA.

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

[4.](#) Security Considerations

TBD

[5.](#) Acknowledgements

[6.](#) Normative References

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

Authors' Addresses

Mach(Guoyi) Chen
Huawei Technologies Co., Ltd
No. 3 Xinxu Road, Shang-di, Hai-dian District
Beijing 100085
China

Email: mach@huawei.com

Internet-Draft

SDN Use Case for VC/VN on Demand

January 2013

Mike McBride
Huawei Technologies Co., Ltd
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: michael.mcbride@huawei.com

Liang Ou
China Telecom

Email: oul@gsta.com

