Network Working Group P. Pan
Internet-Draft
Intended status: Standards Track T. Nadeau
Expires: September 12, 2012
March 11, 2012

Software-Defined Network (SDN) Problem Statement and Use Cases for Data Center Applications

draft-pan-sdn-dc-problem-statement-and-use-cases-02.txt

Abstract

Software Defined Network (SDN) is an overlay architecture that presents the underlying transport network to the applications and services for monitoring, and provisioning at abstraction level.

In this memo, we outline some of the problems, and present an architecture outline. We will present a few applications to validate the problems and the architecture framework.

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

Copyright Notice

Table of Contents

1. Introduction

Service providers and enterprises are increasingly offering services and applications from data centers. When the applications and services are offered from a group of data centers, they would require extensive support from the underlying IP networks.

Software Defined Network (SDN) is an overlay architecture that presents the underlying transport network to the applications and services for monitoring, and provisioning at abstraction level.

The concept of SDN has been controversial: some envision that the deployment of SDN would simplify network operation and system requirements, and thereby reduce overall networking cost. On the other hand, some argue that SDN is adding nothing new in terms of network operation, as SNMP, NetConf and many existing protocols could be equally effective.

We argue that the value in SDN is above and beyond network operation and equipment cost reduction. SDN is a part of network service evolution.

Traditionally, telephone companies have retained the control of voice, video and leased line services. With the emergence of the Internet, the ISPs can trump all those services by leveraging the technology advancement in Ethernet, packet switching and IP routing, and offer VoIP, CDN and VPN services to the end-users at much reduced cost.

The emergence of cloud-based computing has once again changed the networking service models. Essentially, cloud computing is to utilize centralized processing, storage and computing to create a new service layer on top of the network. The services can be modeled as IaaS, PaaS and SaaS etc. Interactive services (which combines voice, video and data) could replace separated voice (e.g. phone) and video (e.g. TV) services. Enterprise services through private cloud may replace the traditional ISP-initiated VPN's. Much of the mobile services, which have traditionally been run by the network service providers, could be realized by application providers over IP/LTE networks. This change has been validated by service offerings from Google, Amazon and Apple in recent years, and likely will continue to proliferate in years to come.

Consequently, the cloud-based services are driving the networking traffic demand. It requires the networking resources to be available in anywhere and anytime. As far as the cloud service providers are concerned, networking is an an utility business. It makes no difference on network types (circuit or packet) and networking

technologies (Ethernet, IP or MPLS), as long as the network can reliably transport user data at competitive price.

Within this framework, SDN plays the role of enabling cloud service providers to have an uniformed application interface to the underlying networks, so that they can optimize the use of the networking resources.

Note that, SDN does not imply the following:

1. Data center management: There has been multiple approaches in constructing data centers network fabric (e.g. Q-Fabric, PBB E-VPN, etc.). SDN does not define and dictate the data center interior architecture and management. Instead, SDN is to interface with the data centers to make the use of the network resources.

2. Storage and computing management: Cloud services can be roughly divided in storage, computing and networking. SDN is only responsible for the networking portion.

3. Direct network operation: SDN is a technical solution that enables the cloud service providers to provision network resources from the application layer. The operation itself must subject to proper business arrangement between data center and network service providers. The SDN resource programming can only take place on abstract level.

In this document, we will articulate the issues and present a general architecture in SDN through a number of user cases.

2. Related Work

There has been much work in this area over the years.

OpenFlow has pioneered the concept of software-defined network via FlowVisor. It has introduced a new packet forwarding methodology to be applied on hardware or software L2 switches. OpenFlow Version 1.0 have been in deployment in VM hypervisor environment. OpenFlow Version 1.2 has been recently released where it has introduced the concept of "virtual interface" for aggregating multiple packet flows. The subsequent new versions will address issues such as extendibility, modularity and carrier-grade.

NETCONF/YANG provides a XML-based solution for network device configuration. It has been in wide-deployment. By definition, it supports server-to-client configuration, but not client-to-server

alarms or feedback.

ALTO is a server solution designed to gather network abstraction information and interface with applications (such as P2P) for more efficient traffic distribution. It does not require configuring the underlying network devices.

PCE is a client-server protocol that operates in MPLS networks that enables the network operators to compute and potentially provision optimal point-to-point and point-to-multipoint connections. However, PCE does not interface with applications to optimize traffic from user applications.

3. The Problem Definition

As mentioned before, SDN is an overlay architecture that presents the underlying transport network to the applications and services for monitoring, and provisioning at abstraction level.

The the existing network does not have a clean interface to the applications. Figure 1 illustrates the relationship between application and network today, where the applications have little or fragmented knowledge, control of or visibility of underlying networks and resources.

```
      +-------------+    +-------------+    +-------------+
      | Application |    | Application |    | Application |
      |     #1      |    |     #2      |    |     #3      |
      +-------------+    +-------------+    +-------------+
             |                  |                  |
             |                  |                  |
      +-----------------------------------------------------+
      |                  Physical Network                   |
      +-----------------------------------------------------+
```

Figure 1: Application to network relationship today

This presents a number of challenges and problems.

First, due to the lack of correlation, it becomes difficult to provide service guarantees at network-level (in particular, delay) to the applications. The operators may over-provision network links to overcome to potential network congestion and packet drop within data centers. However, such practice may become unpredictable and costly in many networking scenarios.

Second, many services require the interface and interaction with 3rd party back-end applications that may operate from remote locations (such as ads networks). This requires the service operators to constantly monitor the SLA conditions with remote applications, and adjust the network resources if necessary.

Third, many data center applications (such as VM) require massive user data replication on different sites for performance and redundancy purposes. Also, due to the limitation in routing and load balancing, much user traffic may be routed between data centers. As such, the inter-data center data transport need to be efficient, which requires the proper interface between applications and network.

Finally, to scale up enterprise applications on data centers, the VM's may locate on different data centers, and mirage between data centers depending on capacity and other constraints. This requires the collaboration between VM applications and the underlying networks.

SDN is to solve the above inefficiency, as envisioned in Figure 2. It is to enable the applications to visualize the traffic flows at IP network layer, and manage the mapping or binding between user traffic flows to the network connections from the edge of the networks.

```
      +-------------+    +-------------+    +-------------+
      | Application |    | Application |    | Application |
      |     #1      |    |     #2      |    |     #3      |
      +-------------+    +-------------+    +-------------+
             |                  |                  |
             |                  |                  |
      +-----------------------------------------------------+
      |                      SDN Layer                      |
      |     (Network virtualization, programmability       |
      |                  and Monitoring)                    |
      +-----------------------------------------------------+
             |                  |                  |
             |                  |                  |
      +-----------------------------------------------------+
      |                   Physical Network                  |
      +-----------------------------------------------------+
```

Figure 2: SDN-enabled network

In summary, SDN is to provide the applications with the following capabilities:

1. The ability to retrieve the underlying topology.

2. The ability to monitor underlying network conditions, such as failure etc.

3. The ability to initiate and adjust network connections/tunnels.

4. The ability for the applications to create services on top of the provisioned network connections directly

3.1. The Architecture

Specifically, the SDN architecture may be constructed as the following:

```
              +---------------------------+
              | Applications and Services |
              +---------------------------+
                            |
                            |
                 +----------------------+        +---------------------+
                 | SDN Service Mediator |<------>|   Network Topology  |
                 +----------------------+        | & Resource Database |
                   ^        |      ^             +---------------------+
                   |        |      |
        +----------+        |      +-------------+
        |                   |                    |
        |                  \|/                   |
   +-----------+      +--------------+      +------------+
   | Discovery |      |   Network    |      | Monitoring |
   +-----------+      | Provisioning |      +------------+
                      +--------------+
```

Figure 3: Proposed SDN Architecture

o SDN Service Mediator: This is a logical server that coordinates applications and networks. It may expand into multiple physical servers in different locations. One may imagine the Service Mediator as an Apache-based web servers, with task scheduling and redundancy functions. The Service Mediator may be owned by the network service providers to serve application providers. Or the application providers may deploy Service Mediator to control traffic over multiple networks.

o Discovery: This is a process controlled by Service Mediator, but deployed to users or devices, in the form of applications or VM's.

It enables the SDN users to discover and register to the Service Mediator. As a part of the discovery process, the SDN users may negotiate capabilities with the service mediator. Some of the common discovery mechanisms could be a DNS extension (in which case, Service Mediator is simply a url for the users to contact), or as simple as IMPP messages.

- o Network Provisioning: This the process that allows the Service Mediator to provision the underlying network resources. There are many ways to provision the networks, depending on the applications. For simple VLAN switching and aggregation, OpenFlow 1.2 may be sufficient. But for more sophisticated networking technologies, such as MPLS and GMPLS, the Service Mediator needs to input a range of attributes to the network (edge) devices in the form of NetConf or other protocols to create or adjust traffic engineering connections.

- o Monitoring: This is an important function in SDN. This allows the Service Mediator to interface with the underlying network to gather network topology information at abstract level, and detect the network failures that may impact the applications and services.

- o Network Topology Database: This is a part of the inventory management that service/application providers maintain. This is an important part of the SDN-enabled network operation.

- o SDN North-Bound Interface: SDN Service Mediator is to provide the RESTful API's to the applications/services. The API interface should be at abstract level and application-friendly.

3.2. Use Case Summary

In the remaining of the document, we will outline a number of potential SDN applications that can be implemented with the architecture outlined above.

1. Bandwidth on Demand: In this application, the applications will provision one or multiple physical links, and construct an overlay network for one or a group of users. The provisioning sequence requires the setup, change or deletion of network connections/tunnels on network devices. This application is also known as 'network slicing'.

2. Virtual Data Center: The data center servers may virtualize applications, processors and devices for the end users. The virtual machines may be located on multiple data centers over IP networks. For performance and reliability reasons, the traffic

from the virtual machines need to be inter-connected or aggregated over the network connections/tunnels that have been discovered and provisioned through SDN.

3. L3 Virtualization: L2 virtualization does not scale. Through the coordination of SDN Service Mediator, the virtual machines may interconnect each other through IP routing protocols directly.

4. Use Cases

4.1. Bandwidth on Demand

In this use case, we show the relevant SDN components for clarity, and suggest some of the possible implementation approaches.

```
              +---------------------------+
              | Applications and Services |
              +---------------------------+
                            |
                        (1) |
                 +----------------------+  (2)   +---------------------+
                 | SDN Service Mediator |<------>|   Network Topology  |
                 +----------------------+        | & Resource Database |
                            |      ^             +---------------------+
                            |      |
                        (3) |      +-------------+
                            |                    | (7)
                           \|/                   |
                      +--------------+      +------------+
                      |   Network    |      | Monitoring |
                      | Provisioning |      +------------+
                      +--------------+            ^
                            |          (6)        |
                         (4)|   +------------------+
                            |   |
                      +--------------+
                      |  Router or   |   (5)
                      |   Switch     |=========( IP Networks )
                      +--------------+
```

Figure 4: Support of Bandwidth-on-Demand

As an illustration, shown in Figure 4, the application needs to create a MPLS connection with certain bandwidth on one of the inter-data center links. Since it is aggregating traffic from a large

number of virtual machines, it needs to be notified in event of network failure.

Here could be the sequence of events:

1. Through the RESTful API interface, the Service Mediator receives the request from applications.

2. The Service Mediator will query the network inventory database to select the proper link and network device for provisioning. For argument sake, the Service Mediator could act as a PCE server, and compute the proper MPLS-TE ERO for the connection.

3. Upon the completion of the path computation, the Service Mediator will initiate the provisioning commands to the Provisioning Engine.

4. Through provisioning protocols (such as, PCE or NetConf or OpenFlow), the Provisioning Engine propagates the commands to the actual switches and routers.

5. The routers (or switches) will utilize the MPLS control-plane protocols to interface with other nodes in the network and complete the connection setup.

6. At some point, a unrecoverable failure has occurred in the network. The Monitoring Engine will pick up the failure condition and compress the alarms.

7. The Monitoring Engine will notify the Service Mediator, which in turn will inform the corresponding applications and services.

There are a number of variations here>:

- The underlying network could be an optical network running GMPLS. The same sequence would apply for setting up large inter-data center links.

- The enterprise network may not use IP routers. A similar sequence could apply on multiple network nodes to perform manual VLAN cross-connects. The provisioning protocol could be OpenFlow.

4.2. Virtual Data Center

The idea here is to construct an overlay network for a group of users. Each user is represented by an individual virtual machine. All users in the same group will share a common network identification (such as VLAN).

When the users communicate among each other, they should not be aware of the underlying networks. This requires the SDN to aggregate the user traffic over a selected set of network connections, which should provide adequate delay and bandwidth guarantees.

In Figure 5, we will illustrate a simple use case:

```
             +---------------------------+
             | Applications and Services |
             +---------------------------+
                           |
                           |
                +----------------------+        +---------------------+
                | SDN Service Mediator |<------>|   Network Topology  |
                +----------------------+        | & Resource Database |
                  ^        |      ^             +---------------------+
                  |        |      |
       +----------+        |      +-------------+
       |                   |                    |
       |                  \|/                   |
  +-----------+      +--------------+      +------------+
  | Discovery |      |   Network    |      | Monitoring |
  +-----------+      | Provisioning |      +------------+
                     +--------------+
                      |  | | |
                 +----+  | | +----------------------------+
                 |       | +---------------------+        |
                 |       |         ^^^^^^        |        |
                 |       |        (      )       |        |
    +------+   +---+   +----+    (        )    +----+   +---+   +------+
    |Server|---|TOR|---|Core|---( Backbone )---|Core|---|TOR|---|Server|
    +------+   +---+   +----+    (        )    +----+   +---+   +------+
                                  (      )
             Data Center           vvvvvv             Data Center
                 #1                                         #2
```

Figure 5: Support of Virtual Data Center

This use case is very similar to that of Bandwidth-on-Demand. The general operation sequence would be:

- o The Service Mediator interfaces with network edge nodes, and provisions the network tunnels.

- o When the application is to setup a logical connection between two virtual machines, the Service Mediator will identify the user

network identification (e.g. VLAN or VXLAN), select the network tunnel to use.

- Fianlly, through Service Mediator, the application is to program the ToR switches and initiate the logical connections. All virtual machine traffic will be aggregated through selected network connections for latency and bandwidth guarantees.

This application is new, and much new signaling protocols have been proposed and studied.

4.3. Virtual Data Center

Due to historic reasons, much of the networking virtualization is through the use of L2 technology. Consequently, the applications are experiencing sever scalability problems. Many recent proposals have been focused in making the L2 technology more scalable, including extending the VLAN range. Many hypervisors are positioned to run L3-over-L2-over-L3.

We argue that to solve the virtual machine scaling problems, we should introduce L3 virtualization.

Here, we introduce a method in supporting L3 virtualization using SDN:

```
                   +---------------------------+
                   | Applications and Services |
                   +---------------------------+
                                 |
                                 |
                      +----------------------+
                      | SDN Service Mediator |
                      +----------------------+
                          ^        |      ^
                          |        |      |
               +----------+        |      +-------------+
               |                   |                    |
               |                  \|/                   |
         +-----------+      +-------------+      +------------+
         |   BGP     |      |   BGP       |      |     BGP    |
         | Signaling |      |  Signaling  |      | Signaling  |
         |   GW      |      |    GW       |      |    GW      |
         +-----------+      +-------------+      +------------+
             |    |             |       |            |    |
             |    +---+         |       |        +---+    |
             |        |         |       |        |        |
           +----+   +----+   +----+  +----+   +----+   +----+
           | VM |   | VM |   | VM |  | VM |   | VM |   | VM |
           +----+   +----+   +----+  +----+   +----+   +----+
```

Figure 6: Support of L3 Virtualization

The idea is that the users (in VMs) may trigger the discovery process, and connect to BGP gateways when connecting to other users. The SDN Service Mediator is to correlate the routing information among BGP GW's.

Much of the processing details can be found in 'BGP-signaled end-system IP/VPNs' (http://www.ietf.org/id/draft-marques-l3vpn-end-system-05.txt)

There are a number of key advantages in this approach.

1. It scales: compare with the existing L2 solutions, this runs at IP layer.

2. Reuse much of the existing and proven routing techniques. The underlying solution is identical to that of MPLS VPN that has been in wide deployment for years.

3. Application-friendly interface through SDN

5. Security Consideration

6. IANA Considerations

7. Acknowledgments

This work is based on the interaction with many people. Thanks Pedro, Lyndon, Shane and Matt.

8. Normative References

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

[RFC2234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

Authors' Addresses

Ping Pan
(Infinera)

Email: ppan@infinera.com

Thomas Nadeau
(CA)

Email: thomas.nadeau@ca.com