IETF
Internet Draft

Ping Pan
(Infinera)
Thomas Nadeau
(CA)

Expires: January 31, 2012

October 31, 2011

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

draft-pan-sdn-dc-problem-statement-and-use-cases-01.txt

Status of this Memo

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 January 31, 2012.

Abstract

Service providers and enterprises are increasingly offering services and applications from data centers. Subsequently, data centers originate significant amount of network traffic. Without proper network provisioning, user applications and services are subject to congestion and delay.

In this document, we argue the necessity in providing network information to the applications, and thereby enabling the applications to directly provision network edge devices and relevant applications.

Table of Contents

1. Introduction

Service providers and enterprises are increasingly offering services and applications from data centers. Subsequently, data centers originate significant amount of network traffic. On contrast to end-to-end user applications, much of the inter-data center traffic is aggregated over a finite number of links over the backbone network. As such, without proper network provisioning, user applications and services are subject to congestion and delay.

Further, many web applications would require the interaction between multiple servers in the networks. Without adequate level of monitoring and provisioning on the network, the users may experience unacceptable services.

In this document, we argue the necessity in providing network information to the applications, and thereby enabling the applications to provision the underlying network edge devices and relevant applications directly.

Here are some of the 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 RFC-2119 [RFC2119].

2. Related Work

There has been much work in this area in recent years.

OpenFlow [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 and 1.1 have been in deployment in VM hypervisor environment. The new versions will address issues such as extendibility, modularity and carrier-grade. Currently, OpenFlow does not support a mechanism to interface with network devices through the existing IP/MPLS control-plane protocols.

NETCONF/YANG provides a XML-based solution for network device configuration. It has been in wide-deployment. By definition, it supports client-to-server configuration, and server-to-client alarms or feedback (The servers are the devices/systems to be configured; the clients are the network configuration/management systems). NETCONF provides support for executing configuration change transactions over multiple devices.

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.

DMTF is a cloud computing standardization organization, which have defined many virtualization management interfaces using Restful API. However, it does not include any interface to the underlying networks.

3. Problem Definition

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.

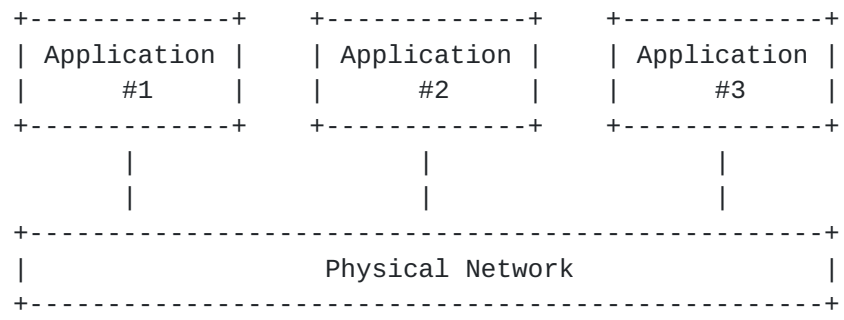

Figure 1: Application to network relationship today

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 too 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.

4. The Role of SDN Layer

To solve the above problem, one simple way is to introduce a software-define network (SDN) layer (as shown in Figure 2), that is responsible for network virtualization, programmability and monitoring, between applications and network.

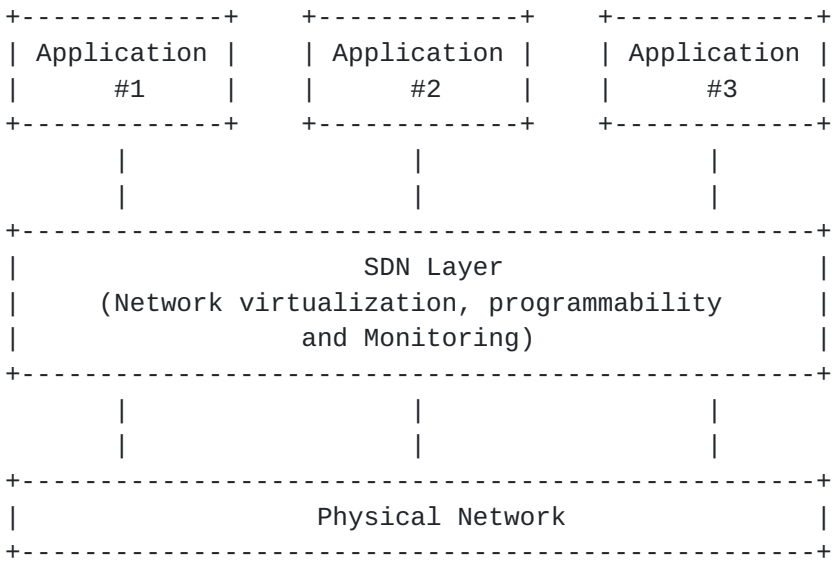

Figure 2: Application to network relationship today

The purpose of the SDN Layer 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.

There are multiple ways in implementing the SDN Layer. There have been multiple proprietary solutions in the area of interfacing Virtual Machines (VM) to the underlying network interfaces. In particular, solutions such as OpenFlow support such vision by directly programming the underlying network interface via a new protocol.

The implementation of SDN Layer involves the interfacing among applications, storage and network devices, which implies that there is a need for having a standardized interface.

Further, we recommend of utilizing the existing technologies and protocols to provision, manage and monitor network connections. The focus in realizing the SDN Layer is in optimizing the application-to-network workflow. The associated SDN protocols need to be modular, scalable and simple in design.

5. Use Cases

5.1. Data Center Network Interface

Figure 3 illustrates the data flow in data centers.

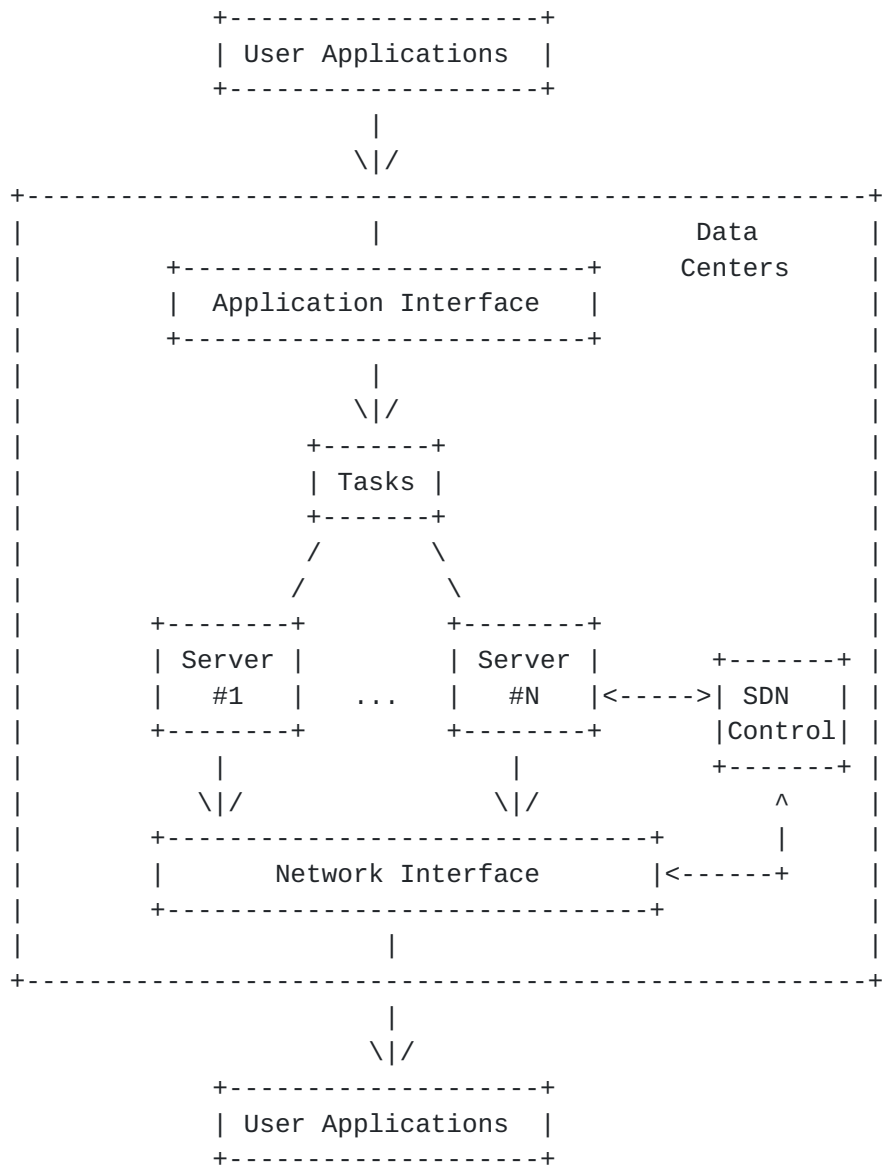

Figure 3: Data Center Traffic Flow

The data centers are designed to scale up to handle a large volume of user requests. To handle the user requests, the application interface would process and bundle the requests to different servers. Depending on the application, the data may flow between the servers or be forwarded to the users through network interface.

Note that when the servers transmit data, they typically do not have the knowledge on network connection bandwidth, delay and distance information. For intra-data center communication, this can be compensated by over-provisioning local networks. However, for transferring data between two remotely located data centers, the applications have no control of the data transmission.

Further, today, when setting up VM's over different servers, extensive manual configuration may be required. For example, all the traffic belongs to the same group/enterprise must share the same VLAN over all involved servers. This can potentially handicap the usability of the applications.

In this case, it would be desirable to have a standardized SDP protocol that can be used by the applications to interface with the networks. Through this protocol, the applications should be able to assign VLAN values to the appropriate VM sessions over all servers, and interface with the connected networks to balance the traffic load if necessary.

5.2. Inter-data center transport

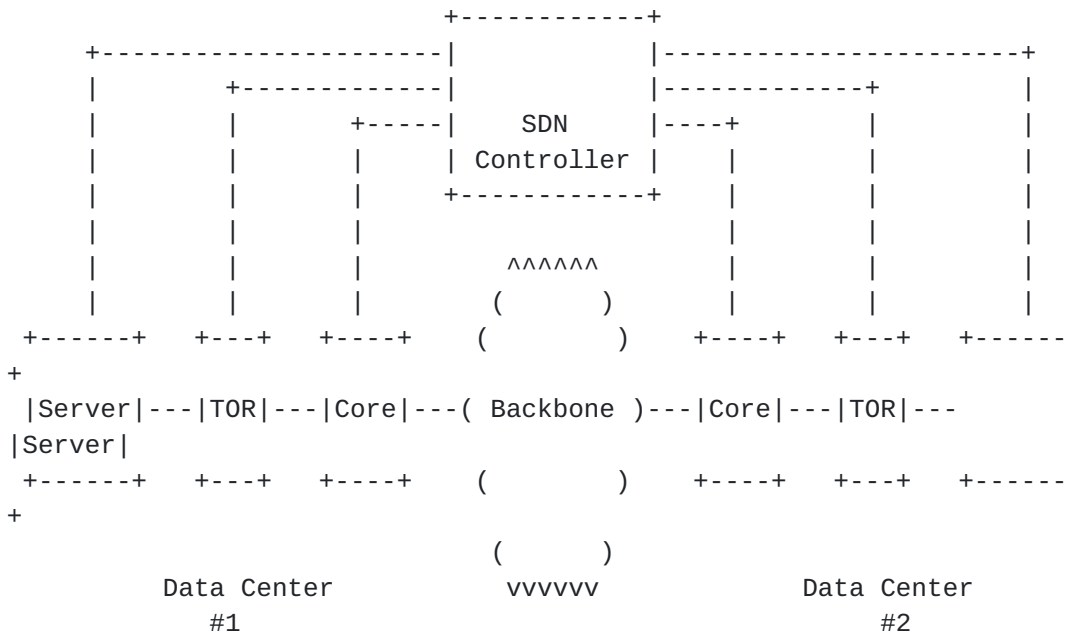

Figure 4: Inter-data center transport

When transporting data between data centers, the packets will be encapsulated into one or multiple tunnels before sending over the Internet. Traffic engineering is typically applied at tunnel-level. For instance, user IP packets at servers may be encapsulated first into a VLAN tunnel, and then aggregated into MPLS LSP's at the core node.

In this case, it would be desirable of having a SDN controller to coordinate the aggregation procedure. The controller is responsible for determining the mapping of the VLAN's to the MPLS LSP's. Further, it is possible that the controller can interface with the core node to adjust the LSP bandwidth.

5.3. VPN

Another use case is VPN, as shown in Figure 5.

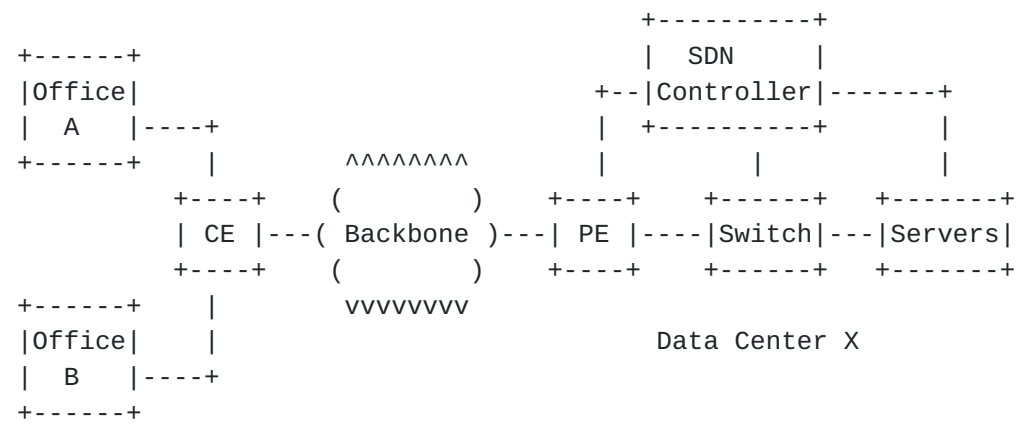

Figure 5: VM Groups to MPLS VPN Mapping

At application level, the service providers may initiate a set of VM's for a specific enterprise. For service guarantee, it requires all the VM's that may be distributed on various servers and data centers to be mapped to the same MPLS (L2)VPN.

There are multiple ways in achieving this goal. One is to utilize a centralized SDN Controller to coordinate the mapping.

5.4. VM Mobility

VM mobility is to move one or multiple VM instances from one data centers to another without disturbing the user existing setting and applications. Once again, there are many ways to achieve such objective.

Traditionally, the operators may create network tunnels between routers/switches in the data centers, and switch VM traffic over the tunnels. However, a more effective solution is to create a logical controller between hypervisors to perform the VM mobility operator.

In the context of SDN, SDN controller can be used to coordinate the hypervisors and the underlying network. It can provide the efficiency in managing the VM's, while making the best use of the underlying network resources.

6. Security Consideration

TBD

7. IANA Considerations

This document has no actions for IANA.

8. Normative References

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

[2] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.

9. Acknowledgments

This work is based on the conversation with many people, including Thomas Nadeau, Lydon Ong and Benson Schliesser.

Authors' Addresses

Ping Pan
Email: ppan@infinera.com

Thomas Nadeau
Email: Thomas.nadeau@ca.com