

INTERNET-DRAFT
Intended Status: Standards Track
Expires: Feb 2016

Congjie Chen
Dan Li
Tsinghua University
Jun Li
University of Oregon
August 2015

SVDC: Software Defined Data Center Network Virtualization Architecture
[draft-chen-svdc-00](#)

Abstract

This document describes SVDC, a highly-scalable and low-overhead virtualization architecture designed for large layer-2 data center networks. By leveraging the emerging software defined network framework, SVDC decouples the global identifier of a virtual network from the identifier carried in the packet header. Hence, SVDC can scale to a large scale of virtual networks with a very short tag in the packet header, which is never achieved by previous network virtualization solutions. SVDC enhances MAC-in-MAC encapsulation in a way that packets with overlapped MAC addresses are correctly forwarded even without in-packet global identifiers to differentiate the virtual networks they belong to. Besides, scalable and efficient layer-2 multicast and broadcast within virtual networks are also supported in SVDC. This document also introduces a basic framework to illustrate SVDC deployment.

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/lid-abstracts.html>

INTERNET DRAFT

SVDC

August 2015

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

Copyright and License Notice

Copyright (c) 2015 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.

This Internet-Draft will expire on January, 2016.

Table of Contents

1	Introduction	3
1.1	Terminology	4
2	SVDC Architecture	5
2.1	Virtual Switch	8
2.2	Edge switches	9
2.3	SVDC Controller	10
3	Packet Forwarding	11
3.1	Unicast Traffic	11
3.2	Multicast/Broadcast Traffic	12
3.3	SVDC Frame Format	13
4	SVDC Deployment Considerations	14
4.1	VM Migration	14
4.2	Fault Tolerance	15
5	Security Considerations	15
6	IANA Considerations	16
7	References	16
7.1	Normative References	16
7.2	Informative References	16
	Authors' Addresses	17

INTERNET DRAFT

SVDC

August 2015

1 Introduction

Due to the simplicity and easiness to manage, large layer-2 network is widely accepted as the fabric to build a data center network. Scalable layer-2 architectures, for example, TRILL [[RFC6325](#)] and SPB [[802.1aq](#)] are proposed as industry standards. A large layer-2 network segment can even cross the Internet via virtualization services such as VPLS [[RFC4762](#)]. However, this kind of layer-2 network fabric design mainly focus on routing/forwarding rules in the network, and it is still an open issue how to run a multi-tenant network virtualization scheme on top of the large layer-2 network fabrics. Existing network virtualization solutions, including VLAN [[802.1q](#)], VXLAN [[RFC7348](#)] and [[NVGRE](#)] either face severe scalability problem or are not specifically designed for layer-2 networks. Particularly, designing a virtualization solution for large layer-2 network needs to address following challenges.

For a large-scale, geographically distributed layer-2 network operated by a cloud provider, the potential number of tenants and virtual networks can be huge. Network virtualization based on VLAN can support at most 4094 virtual networks, which is obviously not enough. Although VXLAN [[RFC7348](#)] and [[NVGRE](#)] can support 16,777,216 virtual networks, they are at the cost of using much more bits in the packet header. The fundamental issue is, in existing network virtualization proposals, the number of virtual networks that can be differentiated depends on the number of bits used in the packet header.

Given the possible overlapped MAC addresses for VMs in different virtual networks and the limited forwarding table size in data center switches, it is inevitable to encapsulate the original MAC address of a packet when transmitting it in the core network. MAC-in-UDP encapsulation used in VXLAN [[RFC7348](#)] incur unnecessary packet header overhead for a layer-2 network. MAC-in-MAC encapsulation framework is more applicable in the multi-tenant large layer-2 network where MAC addresses of VMs largely overlap.

Multicast service is common in data center networks, but how to support scalable multicast service in a multi-tenant virtualized large layer-2 network is still open. A desired capability with a layer-2 network virtualization framework is to support efficient and scalable layer-2 multicast as well as broadcast.

This document describes SVDC, which leverages the framework of [[SDN](#)] to address the challenges above, and achieves the goal of a high scalability and low overhead large layer-2 network virtualization architecture. It decouples the global identifier of a virtual network and the in-packet tag to encompass a great scale of virtual networks

with a minimal tag length in the packet header. The global identifier is maintained in the SVDC controller while the in-packet identifier is only used to differentiate virtual networks residing in the same server. To mask the VM MAC address overlap in the core network, SVDC uses MAC-in-MAC encapsulation in ingress edge switches and employs two techniques to guarantee correct packet forwarding in the first hop and last hop without in-packet global virtual network identifier. What's more, SVDC can efficiently support up to tens of billions of multicast and broadcast groups with possible overlapping multicast or broadcast addresses in different virtual networks in a layer-2 network by the same framework as in unicast.

[1.1](#) Terminology

This document uses the following 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 [RFC 2119](#) [[RFC2119](#)].

Virtual Network (VN): A VN is a logical abstraction of a physical network that provides L2 network services to a set of Tenant Systems.

Virtual Machine (VM): It is an instance of OS's running on top of hypervisor over a physical machine or server. Multiple VMs can share the same physical server via the hypervisor, yet are completely isolated from each other in terms of compute, storage, and other OS resources.

Virtual Switch (vSwitch): A function within a hypervisor (typically implemented in software) that provides similar forwarding services to a physical Ethernet switch. A vSwitch forwards Ethernet frames between VMs running on the same server or between a VM and a physical Network Interface Card (NIC) connecting the server to a physical Ethernet switch or router. A vSwitch also enforces network isolation between VMs that by policy are not permitted to communicate with each other. (e.g., by honoring VLANs).

Global Tenant Network Identifier (GTID): A GTID is a global identifier of a virtual network. It is never carried in packets that VMs send out but maintained in the SVDC controller.

Local Tenant Network Identifier (LTID): A LTID is a local identifier that is used to differentiate virtual networks on the same server. For the same virtual network, its LTID in different servers can either be different or the same. When a new virtual network is created, it will be assigned a LTID in each server that hosts its VMs.

Global Identifier of a Multicast/Broadcast Group (Group-G): It is used to denote the address of a multicast/broadcast group that can be used in the physical network in the SVDC architecture. When a new multicast/broadcast group wants to send traffic across the core network, an available Group-G will be assigned to it. When all the receivers of a group leave a multicast group, or a broadcast group lacks of activity for a long duration, the corresponding Group-G will be removed.

Local Identifier of a Multicast/Broadcast Group (Group-L): It is used to denote the address of a multicast/broadcast group within a virtual network. Group-L in different virtual networks can be overlapped.

Edge Switch Identifier (EID): It is used to denote the identifier of an edge switch. Any identifier of a switch such as the MAC address of a switch can be represented as it.

Server Identifier (SID): It is used to denote the identifier of a physical server just like EID.

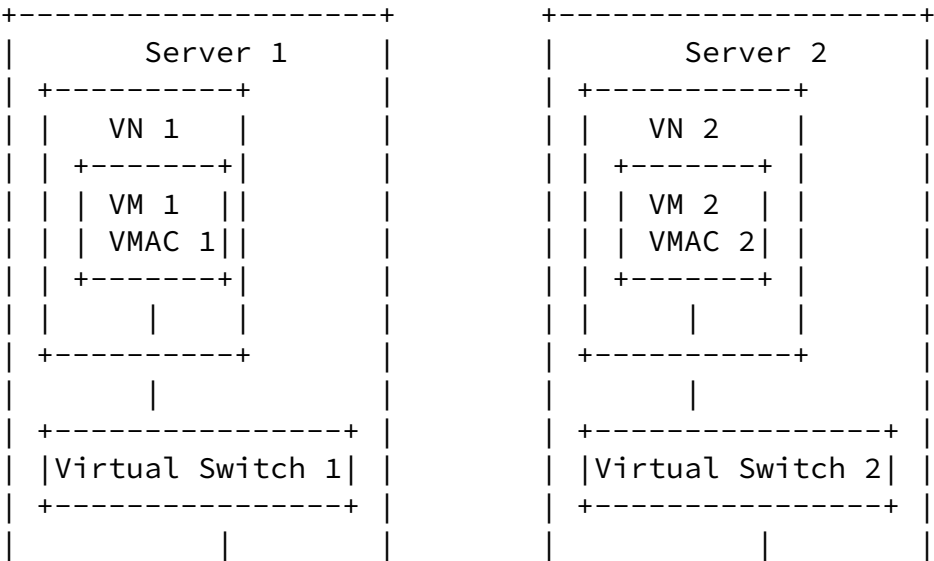
Virtual Machine MAC Address (VMAC): This is the MAC address assigned to the virtual NIC of each VM. It is visible to VMs and applications

running within VMs.

Egress Port Identifier (p-ID): It denotes the outgoing port to which the egress edge switch should forward the packet.

2. SVDC Architecture

The basic architecture of SVDC is depicted in Figure 1.



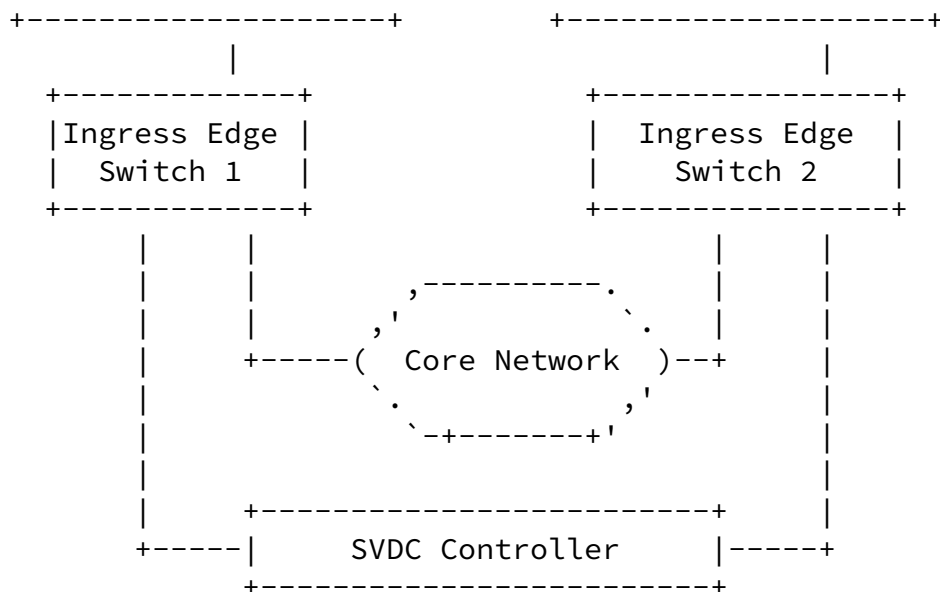


Figure 1 SVDC Architecture

In minimum configuration, the SVDC architecture only contains an SVDC controller and the updated edge switches. The controller interacts with the edge switches using an SDN protocol like [\[OPENFLOW\]](#). A very light-weight modification on the virtual switch is required to fill the server-local identifier of a virtual network into the packet. Core switches and VMs just run legacy protocols, and can be unaware of SVDC.

In the core network, any kind of layer-2 forwarding schemes can be used, for example, Spanning Tree Protocol (STP) [\[802.1D\]](#), TRILL protocol [\[RFC6325\]](#) and Shortest Path Bridging protocol [\[802.1aq\]](#) for unicast, while global multicast tree formation protocol for multicast. However, up to the operator's configuration, the SVDC controller can also use [\[OPENFLOW\]](#) to configure the unicast/multicast

forwarding entries in the core network. SVDC can seamlessly coexist with any forwarding fabric in the core network, either SDN or non-SDN.

Every virtual switch maintains a local FIB table with entries destined to VMs on the local server, while packets sent to all the other VMs are simply forwarded to the edge switch it connects to. An edge switch maintains both a unicast encapsulation table and a

multicast encapsulation table, used in MAC-in-MAC encapsulation for every packet. When the first packet of a flow arrives at an ingress edge switch, the encapsulation table lookup will fail and then the packet is directed to the SVDC controller. The SVDC controller then looks up its mapping tables which maintain the global information of the network, and responds to the ingress switch with the information to update its encapsulation table. Subsequent packets of the flow will be directly encapsulated by looking up the encapsulation table, without interrupting the SVDC controller again. Multicast group join requests are also directed to the SVDC controller, and then the controller updates the multicast decapsulation table in corresponding egress switches with group membership.

SVDC supports a great scale of virtual networks by maintaining a global identifier for every virtual network in the SVDC controller, but never carrying the identifier in the packet. Instead, a server-local identifier is carried in the packet header to identify a virtual network on a certain physical server. The SVDC controller maintains the mapping relationship between the global and local identifiers, and is responsible for the translation when the first packet of a flow is directed to the SVDC controller. The translation includes both mapping a server-local virtual network identifier to the global identifier, and vice-versa. SVDC reuses the 12-bit VLAN [802.1q] field as the in-packet server-local virtual network identifier, which should be adequate since the number of virtual networks in a physical server cannot exceed 4096.

To minimize the packet header overhead introduced due to encapsulating the original Ethernet packets from VMs in a layer-2 network, SVDC uses MAC-in-MAC encapsulation in ingress switches. It not only masks the MAC address overlap from VMs in different virtual networks, but also minimizes the number of forwarding entries in core switches. The key point here is how to guarantee correct packet forwarding in the first hop and last hop, since no information is carried in the packet to globally differentiate the virtual networks in a direct way. SVDC has two approaches to deal with these problems.

First, for the ingress switch to identify the virtual network an incoming packet belongs to, only the server-local identifier carried in the VLAN field is not enough. But the VLAN field together with the

incoming port of the switch are just enough for the identification,

since the incoming port of the switch can uniquely identify the physical server where the packet is sent from.

Second, when the egress switch decapsulates the outer MAC header, it needs a way to correctly forward the packet to an outgoing port. Local table lookup cannot help because the in-packet virtual network identifier is not the global one and thus can overlap. The way we come up with is to reuse the VLAN field of the outer MAC header to indicate the forwarding port in the egress switch. The field is filled in the ingress switch for a unicast packet by looking up the unicast encapsulation table, and filled in the egress switch for a multicast packet by looking up the multicast decapsulation table. The 12-bit VLAN tag is also more than enough to identify different servers connecting the egress switch, unless the egress switch has more than 4096 ports, which cannot happen in practice.

SVDC encompasses multicast and broadcast within each virtual network with possible overlapping group addresses. In order to avoid traffic leakage among virtual networks, the SVDC controller maps each multicast group or broadcast in a virtual network to a global multicast group, which can be identified by the global multicast group address, composed of 23-bit multicast MAC address and 12-bit VLAN field. This 35-bit global multicast group address is enough to support a potentially huge number of multicast/broadcast groups within virtual networks and can be carried in the outer Ethernet header.

The following sections will describe the design detail of each component in SVDC architecture.

[2.1](#) Virtual Switch

Every virtual switch configures its FIB table entries towards VMs in the local server, and sets the forwarding port of the default entry towards the edge switch connecting to the server it resides in. The key of the FIB table entry in virtual switch is a tuple (LTID,VMAC), which uniquely identifies a VM in a physical server. Note that in SVDC, VMs are not aware of the virtualized network infrastructure, and thus the Ethernet header sent by a VM does not contain any LTID.

When a virtual switch receives an Ethernet packet, it first determines whether it is from a local VM or from the outbound port. If from a local VM, the virtual switch adds the LTID in the VLAN field of the Ethernet header based on the incoming port and then forwards it out. If from the outbound port, operations on it depend on whether it is a unicast packet or a multicast/broadcast packet. For a unicast packet, the virtual switch directly looks up the FIB

table and forwards it to a certain VM in the local server; for a broadcast packet, the virtual switch forwards it to all VMs within the same virtual network on the local server; while for a tenant-defined multicast packet, the virtual switch forwards it towards VMs that are interested in it, which can be learned by snooping the multicast group join message sent by VMs.

[2.2](#) Edge switches

Edge switches bear most intelligence of the data plane in SVDC. It is responsible for rewriting VLAN field in the inner Ethernet packet header and encapsulating/decapsulating the original Ethernet packets.

Every ingress edge switch maintains a unicast encapsulation table which maps from (in-port, LTID-s, VM-d) to (LTID-d, ES-d, p-ID), where in-port is the incoming port of the packet, LTID-s is the LTID of the virtual network in the source server, VM-d is the MAC address of the destination VM in the original Ethernet header, LTID-d is the LTID of the virtual network in the destination server, ES-d is the MAC address of the egress edge switch, and p-ID is the outgoing port to which the egress edge switch should forward the packet. If the lookup hits, the ingress edge switch will do the following operations. First, it rewrites LTID-s in the VLAN field of the original Ethernet header as LTID-d. Second, it encapsulates the packet by adding an outer Ethernet header, with ES-d as the destination MAC address, its own MAC address (ES-s) as the source MAC address, and p-ID as the VLAN field. Third, it forwards the encapsulated packet by looking up the forwarding table. However, if the lookup fails, the ingress edge switch will direct the packet to the SVDC controller with incoming port of the packet, which helps the controller obtain the information required to install an encapsulation entry in the unicast encapsulation table.

A multicast encapsulation table is also maintained, which maps from the tuple (in-port, LTID-s, Group-L) to the global multicast group address Group-G to fill in the outer Ethernet header. If the lookup hits, it encapsulates the multicast/broadcast packets with Group-G as the destination MAC address and VLAN ID while ES-s as the source MAC address. If the lookup misses, it will send this packet to the SVDC controller to update the multicast encapsulation table.

Since VMs of a certain group can have different LTIDs in different servers, egress edge switches should rewrite LTID in the inner Ethernet header for each packet duplication destined to different servers. Thus, every egress edge switch maintains a multicast decapsulation table, which maps from Group-G to multiple (Out-PORT,

LTID-d) tuples, where Out-PORT is an output port of a multicast/broadcast packet duplication and LTID-d is the LTID of the

virtual network in the destination server connecting to the Out-PORT. Entries in this table are inserted by the SVDC controller when the multicast group join message sent by a VM is directed to it. When an egress edge switch receives a multicast packet, it first duplicates this packet as the number of (Out-PORT,LTID-d) tuples. Then, it decapsulates each packet duplication, rewrites the LTID in the inner Ethernet header of each packet duplication as indicated by LTID-d and sends each packet duplication towards the destination server as indicated by the Out-PORT.

[2.3](#) SVDC Controller

The SVDC controller keeps several groups of mapping tables based on its global knowledge of the network.

- LT-GT MAP: (SID, LTID) is mapped to GTID.
It is used to identify the global identifier of a virtual network based on a physical server identifier and its local virtual network identifier.
- VM-LT MAP: (GTID, VMAC) is mapped to (SID,LTID).
Based on the global identifier of a virtual network and a certain MAC address, we can uniquely identify the physical server a VM resides in as well as the local identifier of the virtual network on that server.
- SID-ES MAP: (EID, port) is mapped to SID and vice versa.
This mapping table can be directly obtained from the network topology and it is used to identify the server connected to a certain port of an edge switch or vice versa.
- GL-GG MAP: (GTID,Group-L) is mapped to Group-G.
It is used to map a multicast group or broadcast address within a virtual network to its global multicast group address.

The main function of the SVDC controller is to respond to requests from edge switches with information they need, which helps install the encapsulation/decapsulation table entries in the ingress/egress edge switches. When an ingress edge switch receives the first packet

of a flow, it directs the packet to the controller with the incoming port of the packet and queries the controller for the information required.

If it is a unicast data packet, the controller first uses SID-ES MAP to get the SID of the source server. By source server's SID and LTID in the original packet, the controller then identifies GTID of the virtual network by LT-GT MAP. Based on the GTID and the destination MAC address of the original packet, the controller can use VM-LT MAP

to further identify the destination SID and LTID of the virtual network in the destination server. Finally, the controller depends on the SID-ES MAP again to get the MAC address of the egress edge switch as well as the port number of the egress edge switch connecting to the destination server. Now, the SVDC controller can return all the information needed by the ingress edge switch to construct an unicast encapsulation table entry.

If it is a multicast data packet, the controller uses SID-ES MAP and LT-GT MAP sequentially to get the GTID of the virtual network as aforementioned. Then, if the controller can find a corresponding entry in GL-GG MAP to get Group-G, it returns Group-G to the ingress switch to build the multicast encapsulation table. If not, it will find an available global multicast group address Group-G, insert a new entry to GL-GG MAP, and return the new Group-G to the ingress edge switch.

If it is a multicast group join request, the SVDC controller first gets the GTID of the virtual network by using SID-ES MAP and LT-GT MAP sequentially. Then, it looks up the GL-GG MAP to find the corresponding Group-G. If the SVDC controller can find one, it just responds to the edge switch with this information. If not, the SVDC controller will find an available Group-G and insert a new entry to the GL-GG MAP before it responds it to the edge switch. After the edge switch gets the Group-G from the SVDC controller, it inserts a new entry into the multicast decapsulation table with Out-PORT as the incoming port of the multicast group join request and LTID-d as the LTID of it.

If the cloud provider's layer-2 data center networks are geographically distributed across the Internet, the SVDC controller needs to maintain the information of all cloud data center networks

of this cloud provider. In practice, each data center network has a controller and the global information is synchronized among the controllers periodically.

[3.](#) Packet Forwarding

[3.1](#) Unicast Traffic

When a unicast packet is generated by a VM and sent out to the local virtual switch, it carries the destination MAC address (VM-d), the source MAC address (VM-s), and leaves the VLAN field empty.

The virtual switch then adds the local LTID (LTID-s) into the VLAN field of the packet and looks up the local FIB table for forwarding.

If the destination VM is within the local server, the packet will be directly forwarded to it. Otherwise, the packet is delivered to the ingress edge switch ES-s.

Next, the ingress edge switch ES-s looks up its encapsulation table using (in-port, LTID-s, VM-d) as key. If missed, the ingress edge switch directs the packet to the controller and the controller installs the encapsulation entry for the flow. If hit, the ingress edge switch obtains the tuple (LTID-d, ES-d, p-ID). Then VLAN field of the original Ethernet header is changed from LTID-s to LTID-d, and an outer Ethernet header is added. The ingress edge switch immediately looks up the FIB table to forward the packet.

After that, the packet is delivered by core switches towards the egress edge switch ES-d. The egress edge switch gets the VLAN field of the outer Ethernet header p-ID, decapsulates the outer Ethernet header, and forwards it to the port p-ID.

Finally the packet arrives at the destination virtual switch. The virtual switch looks up the FIB table based on LTID-d and VM-d, and delivers it to the destination VM.

[3.2](#) Multicast/Broadcast Traffic

When a VM generates a multicast packet, the destination address field of the Ethernet header is filled with the layer-2 multicast group address, denoted as Group-L. This packet then goes to the virtual switch, which inserts LTID-s into the VLAN field and forwards it towards the ingress edge switch.

The ingress edge switch ES-s looks up its multicast encapsulation table using (in-port, LTID-s, Group-L) as key. If missed, the ingress edge switch directs the packet to the controller. Then, the controller installs the multicast encapsulation entry into the ingress edge switch and the multicast decapsulation entries into the egress edge switches. If hit, the ingress edge switch gets the global multicast group address Group-G to fill in the outer Ethernet header.

This packet is then forwarded towards the egress edge switches along the multicast tree. When an egress edge switch receives this packet, it takes Group-G filled in the outer Ethernet header as key and gets multiple (Out-PORT,LTID-d) tuples. It then duplicates the packet as the number of the tuples, decapsulates each packet duplication, rewrites the LTID of it and forwards it towards the Out-PORT.

Finally, the packet arrives at the destination virtual switch and is forwarded towards VMs which have joined the multicast group in the

virtual network.

[3.3](#) SVDC Frame Format

To mask the overlapped VM MAC addresses and mitigate the limitation of the forwarding table size in switches. SVDC enhances MAC-in-MAC encapsulation to guarantee correct packet forwarding. Figure 2 demonstrates the packet format of the MAC-in-MAC encapsulation used in SVDC.

Outer Ethernet Header:

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Outer Destination MAC Address                               |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Outer Destination MAC Address | Outer Source MAC Address                               |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Outer Source MAC Address                               |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```



Figure 2. SVDC MAC-in-MAC Packet Format

The outer Ethernet header: The source Ethernet address in the outer Ethernet header is set to the MAC address of the ingress edge switch. The destination Ethernet address is either set to the MAC address of

the egress edge switch (in unicast traffic) or set to the first 48 bits of the Global-G assigned to the virtual network (in multicast/broadcast). To distinguish SVDC packets, the Ethertype of the outer Ethernet header needs to be set to a specific SVDC Ethertype. The outer VLAN information is used to indicate either the egress port of the packet in the egress edge switch (in unicast traffic) or the last 12 bits of the Global-G of the virtual network.

The Inner Ethernet header: The source and destination Ethernet address in the inner Ethernet header is set to the MAC address of the source and destination VM, respectively. Value of the VLAN tag is

used to indicate the LTID of the virtual network this packet belongs to in the destination server. The payload of the inner Ethernet header includes the Ethertype of the original payload and the original Ethernet payload.

[4.](#) SVDC Deployment Considerations

[4.1](#) VM Migration

To handle VM migration, a central VM manager which can communicate with all hosts needs to be deployed in the network. The SVDC controller needs to be co-located with this central VM manager. In this scenario, when a VM is about to migrate, the VM manager will notify the SVDC controller about the destination server ID, the IP address and the GTID of this VM.

SVDC controller needs to check whether a LTID is assigned to the virtual network of this VM in the destination server before VM migration starts. If not, a LTID will be created and the virtual switch on the destination server will be configured.

After VM migration completes, a gratuitous ARP message is sent from the destination server to announce the new location of the VM. This ARP message is directed to SVDC controller for broadcast entries query when it arrives at the edge switch. In this way, SVDC can confirm VM migration completion and update the location information of this VM in its mapping tables.

To maintain the communication states destined for the migrated VM in edge switches, SVDC controller broadcasts an entry update message to all edge switches immediately after it receives the gratuitous ARP message. This message contains the (LTID, ES, p-ID) tuple the migrated VM uses after migration. All edge switches that maintain encapsulation table entries toward the migrated VM update their encapsulation tables and keep the communication states towards the

migrated VM. The gratuitous ARP message is then sent to VMs within the same virtual networks to update the ARP tables of them.

[4.2](#) Fault Tolerance

An important aspect of large virtualized data center network is the increased likelihood of failures. SVDC tolerates server failures as well as edge switch failures, because no "hard state" is associated with a specific virtual switch or edge switch. In large virtualized data center, it is rational to assume that there are virtual network and physical network management systems which are responsible for detecting failed virtual switches or edge switches.

However, it is necessary for SVDC to handle failures of controller instances or control links between controller instances and edge switches. To handle failures of controller instances, more than one controller instances can be used to manage each network element. All controller instances will synchronize network information periodically. They can work in hot backup or cold backup mode. When one controller instance fails, another instance can replace it in time. To handle failures of control links, traditional routing protocols that are fault-tolerant, e.g. Spanning-Tree protocol [[802.1D](#)], can be applied to the out-band management network deployment. For in-band management network deployment, we assume the layer-2 routing scheme in the core network can take the responsibility to handle link failures.

[5](#) Security Considerations

Since SVDC enhances MAC-in-MAC technique to implement network virtualization, it faces several security challenges that traditional Ethernet network also faces, such as layer-2 traffic snooping, packet flooding causing denial of service attack and MAC address spoofing. In SVDC, malicious end-point can choose to attack the SVDC controller by forging a great number of communication request with different source and destination pairs or hijack the MAC address of the edge switch to interfere the normal communication between the SVDC controller and the edge switches.

Traditional layer-2 technique can be deployed in SVDC to handle these problems, for example, IEEE 802.1 port admission control mechanism [[802.1X](#)] can be used to mitigate the spoofing problem. The security of the communication channel between edge switches and the SVDC controller relies on security mechanism in transport layer.

[6](#) IANA Considerations

This document has no actions for IANA, but SVDC needs to be assigned a new ethertype.

[7](#) References

[7.1](#) Normative References

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

[7.2](#) Informative References

[802.1aq] IEEE, "Standard for Local and metropolitan area networks -- Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks -- Amendment 20: Shortest Path Bridging", IEEE P802.1aq-2012, 2012.

[802.1D] IEEE, "Draft Standard for Local and Metropolitan Area Networks/ Media Access Control (MAC) Bridges", IEEE P802.1D-2004, 2004.

[802.1Q] IEEE, "Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks.", IEEE Standard 802.1Q, 2005 Edition, May 2006.

[802.1X] IEEE, "IEEE Standard for Local and Metropolitan area networks -- Port-Based Network Access Control", IEEE Std 802.1X-2010, February 2010.

[RFC4762] Lasserre, M. and Kompella, V., "Virtual private LAN service (VPLS) using label distribution protocol (LDP) signaling", [RFC 4762](#), January 2007.

[RFC6325] Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (Rbridges): Base Protocol Specification", [RFC 6325](#), July 2011.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., and Agarwal, P., "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", [RFC 7348](#), August 2014.

[NVGRE] Sridharan, M., A. Greenberg, N. Venkataramiah, Y. Wang, K.

INTERNET DRAFT

SVDC

August 2015

Tumuluri. "NVGRE: Network virtualization using generic routing encapsulation." IETF draft, April, 2015.

[SDN] Open Networking Foundation White Paper, "Software-Defined Networking: The New Norm for Networks", April 2012.

[OPENFLOW] McKeown, N., T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. "OpenFlow: enabling innovation in campus networks (OpenFlow White Paper)." Online: <http://www.openflowswitch.org> 2008.

Authors' Addresses

Congjie Chen
4-104, FIT Building,
Tsinghua University,
Hai Dian District,
Beijing, China

EMail: ccjguangzhou@gmail.com

Dan Li
4-104, FIT Building,
Tsinghua University,
Hai Dian District,
Beijing, China

EMail: toolidan@tsinghua.edu.cn

Jun Li
Network and Security Research Laboratory,
Department of Computer and Information Science,

University of Oregon,
1585 E 13th Ave.
Eugene, OR 97403

EMail: lijun@cs.uoregon.edu

Chen, et al.

Expires Feb 2016

[Page 17]