NVO3 Working Group
INTERNET-DRAFT
Intended Status: Informational

Yizhou Li
Lucy Yong
Huawei Technologies
Lawrence Kreeger
Cisco
Thomas Narten
IBM
David Black
EMC

Expires: November 17, 2014
May 16, 2014

Hypervisor to NVE Control Plane Requirements
draft-yizhou-nvo3-hpvr2nve-cp-req-00

Abstract

This document describes the control plane protocol requirements when NVE is not co-located with the hypervisor on a server. A control plane protocol (or protocols) between a hypervisor and its associated external NVE(s) is used for the hypervisor to populate its virtual machines states to the NVE(s) for further handling. This document illustrates the functionalities required by such control plane signaling protocols and outlines the high level requirements to be fulfiled. Virtual machine states and state transitioning are summarized to help clarifying the needed requirements.

Status of this Memo

Copyright and License Notice

Table of Contents

1. Introduction

This document describes the control plane protocol requirements when NVE is not co-located with the hypervisor on a server. A control plane protocol (or protocols) between a hypervisor and its associated external NVE(s) is used for the hypervisor to populate its virtual machines states to the NVE(s) for further handling. This protocol is mentioned in NVO3 problem statement [I-D.ietf-nvo3-overlay-problem-statement] as the third work item. When TS and NVE are on the separate devices, we also call it split TS-NVE architecture and it is the primary interest in this document.

Virtual machine states and state transitioning are summarized in this document to illustrates the functionalities required by the control plane signaling protocols between hypervisor and the external NVE. Then the high level requirements to be fulfiled are outlined.

This document uses the term "hypervisor" throughout when describing the scenario where NVE functionality is implemented on a separate device from the "hypervisor" that contains a VM connected to a VN. In this context, the term "hypervisor" is meant to cover any device type where the NVE functionality is offloaded in this fashion, e.g., a Network Service Appliance.

This document often uses the term "VM" and "Tenant System" (TS) interchangeably, even though a VM is just one type of Tenant System that may connect to a VN. For example, a service instance within a Network Service Appliance may be another type of TS. When this document uses the term VM, it will in most cases apply to other types of TSs.

1.1 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].

This document uses the same terminology as found in [I-D.ietf-nvo3-framework] and [I-D.ietf-nvo3-nve-nva-cp-req]. This section defines additional terminology used by this document.

VN Profile: Meta data associated with a VN that is used by an NVE when ingressing/egressing packets to/from a specific VN. Meta data could include such information as ACLs, QoS settings, etc. The VN Profile contains parameters that apply to the VN as a whole. Control protocols could use the VN ID or VN Name to obtain the VN Profile.

VSI: Virtual Station Interface. [IEEE 802.1Qbg]

VDP: VSI Discovery and Configuration Protocol [IEEE 802.1Qbg]

1.2 Target Scenarios

In split TS-NVE architecture, an external NVE can provide an offload of the encapsulation / decapsulation function, network policy enforcement, as well as the VN Overlay protocol overheads. This offloading may provide performance improvements and/or resource savings to the End Device (e.g. hypervisor) making use of the external NVE.

The following figures give example scenarios where the Tenant System and NVE are on different devices.

```
           Hypervisor             Access Switch
      +------------------+       +-----+-------+
      | +--+   +-------+ |       |     |       |
      | |VM|---|       | | VLAN  |     |       |
      | +--+   |Virtual|---------+ NVE |       +--- Underlying
      | +--+   |Switch | | Trunk |     |       |    Network
      | |VM|---|       | |       |     |       |
      | +--+   +-------+ |       |     |       |
      +------------------+       +-----+-------+
```

Hypervisor with an External NVE

```
                                  Access
           Hypervisor             Switch         NVE
      +------------------+       +-----+       +-----+
      | +--+   +-------+ |       |     |       |     |
      | |VM|---|       | | VLAN  |     | VLAN  |     |
      | +--+   |Virtual|---------+     +-------+     +--- Underlying
      | +--+   |Switch | | Trunk |     | Trunk |     |    Network
      | |VM|---|       | |       |     |       |     |
      | +--+   +-------+ |       |     |       |     |
      +------------------+       +-----+       +-----+
```

Hypervisor with an External NVE across an Ethernet Access Switch

```
       Network Service Appliance         Access Switch
      +--------------------------+      +-----+-------+
      | +------------+    |\     |      |     |       |
      | |Net Service |----| \    |      |     |       |
      | |Instance    |    |  \   | VLAN |     |       |
      | +------------+    |   |---------+ NVE |       +--- Underlying
      | +------------+    |   |  | Trunk|     |       |    Network
      | |Net Service |----|  /   |      |     |       |
      | |Instance    |    | /    |      |     |       |
      | +------------+    |/     |      |     |       |
      +--------------------------+      +-----+-------+
           Physical Network Service Appliance with an External NVE
```

Figure 1 Split TS-NVE Architecture

Tenant Systems connect to NVEs via a Tenant System Interface (TSI). The TSI logically connects to the NVE via a Virtual Access Point (VAP) [I-D.ietf-nvo3-arch]. NVE may provide Layer 2 or Layer 3 forwarding. In split TS-NVE architecture, external NVE may be able to reach multiple MAC and IP addresses via a TSI. For example, Tenant Systems that are providing network services (such as firewall, load balancer, VPN gateway) are likely to have complex address hierarchy. It implies if a given TSI disassociates from one VN, all the MAC and IP addresses are also disassociated. There is no need to signal the deletion of every MAC or IP when the TSI is brought down or deleted. In the majority of cases, a VM will be acting as a simple host that will have a single TSI and single MAC and IP visible to the external NVE.

1.3 Motivations and Purpose

The problem statement [I-D.ietf-nvo3-overlay-problem-statement], discusses the needs for a control plane protocol (or protocols) to populate each NVE with the state needed to perform its functions.

In one common scenario, an NVE provides overlay encapsulation/decapsulation packet forwarding services to Tenant Systems (TSs) that are co-resident with the NVE on the same End Device (e.g. when the NVE is embedded within a hypervisor or a Network Service Appliance). In such cases, there is no need for a standardized protocol between the hypervisor and NVE, as the interaction is implemented via software on a single device. While in the split TS-NVE architecture scenarios, as shown in figure 1, some control plane signaling protocol needs to run between hypervisor and external NVE to pass the relevant state information. Such interaction is mandatory. This document will identify the requirements for such signaling protocol.

Section 2 describes VM states and state transitioning in its lifecycle. Section 3 introduces Hypervisor-to-NVE signaling protocol functionality derived from VM operations and network events. Section 4 outlines the requirements of the control plane protocol to achieve the required functionality.

2. VM Lifecycle

[I-D.ietf-opsawg-vmm-mib] shows the state transition of a VM in its figure 2. Some of the VM states are of the interest to the external NVE. This section illustrates the relevant phases or event in VM lifecycle. It should be noted that the following subsections do not give an exhaustive traversal of VM lifecycle state. They are intended as the illustrative examples which are relevant to split TS-NVE architecture, not as prescriptive text; the goal is to capture sufficient detail to set a context for the signaling protocol functionality and requirements described in the following sections.

2.1 VM Creation

VM creation runs through the states in the order of preparing, shutdown and running [I-D.ietf-opsawg-vmm-mib]. The end device allocates and initializes local virtual resources like storage in the VM preparing state. In shutdown state, VM has everything ready except that CPU execution is not scheduled by the hypervisor and VM's memory is not resident in the hypervisor. From the shutdown state to running state, normally it requires the human execution or system triggered event. Running state indicates the VM is in the normal execution state. Frame can be sent and received correctly. No ongoing migration, suspension or shutdown is in process.

In VM creation phase, tenant system has to be associated with the external NVE. Association here indicates that hypervisor and the external NVE have signaled each other and reached some agreement. Relevant parameters or information have been provisioned properly. External NVE should be informed with VM's MAC address and/or IP address. Another example is that hypervisor may use a locally significant VLAN ID to indicate the traffic destined to a specified VN. Both hypervisor and NVE sides should agree on that VID value for later traffic identification and forwarding.

External NVE needs to do some preparation work before it signals successful association with tenant system. Such preparation work may include locally saving the states and binding information of the tenant system and its VN, communicating with peer NVEs and/or NVA for network provisioning, etc.

Tenant System association should be performed before VM enters running state, preferably in shutdown state. If association with external NVE fails, VM should not go into running state.

2.2 VM Live Migration

Live migration is sometimes referred to as "hot" migration, in that from an external viewpoint, the VM appears to continue to run while being migrated to another server (e.g., TCP connections generally survive this class of migration). In contrast, suspend/resume (or "cold") migration consists of suspending VM execution on one server and resuming it on another. For simplicity, the following abstract summary about live migration assumes shared storage, so that the VM's storage is accessible to the source and destination servers. Assume VM migrates from hypervisor 1 to hypervisor 2. VM live migration involves the state transition on both hypervisors, source hypervisor 1 and destination hypervisor 2. VM state on source hypervisor 1 transits from running to migrating and then to shutdown [I-D.ietf-opsawg-vmm-mib]. VM state on destination hypervisor 2 transits from shutdown to migrating and then running.

External NVE connecting to destination hypervisor 2 has to associate the migrating VM with it by saving VM's MAC and/or IP addresses, its VN, locally significant VID if any, and provisioning other network related parameters of VM. The NVE may be informed about the VM's peer VMs, storage devices and other network appliances with which the VM needs to communicate or is communicating. VM on destination hypervisor 2 SHOULD not go to running state before all the network provisioning and binding has been done.

VM on source hypervisor and destination hypervisor SHOULD not be in running state at the same time during migration. VM on source hypervisor goes into shutdown state only when VM on destination hypervisor has successfully been entering the running state. It is possible that VM on the source hypervisor stays in migrating state for a while after VM on the destination hypervisor is in running state.

2.3 VM termination

VM termination is also referred to as "powering off" a VM. VM termination leads its state going to shutdown. There are two possible causes to terminate a VM [I-D.ietf-opsawg-vmm-mib], one is the normal "power off" of a running VM; the other is that VM has been migrated to other place and the VM image on the source hypervisor has to stop executing and to be shutdown.

In VM termination, the external NVE connecting to that VM needs to

deprovision the VM, i.e. delete the network parameters associated with that VM. In other words, external NVE has to de-associate the VM.

2.4 VM Pause, suspension and resumption

VM pause event leads VM transiting from running state to paused state. Paused state indicates VM is resident in memory but no longer scheduled to execute by the hypervisor [I-D.ietf-opsawg-vmm-mib]. VM can be easily re-activated from paused state to running state.

VM suspension leads VM to transit state from running to suspended and VM resumption leads VM to transit state from suspended to running. Suspended state means the memory and CPU execution state of the virtual machine are saved to persistent store. During this state, the virtual machine is not scheduled to execute by the hypervisor [I-D.ietf-opsawg-vmm-mib].

In split TS-NVE architecture, external NVE should keep any paused or suspended VM in association as VM can return to running state at any time.

3. Hypervisor-to-NVE Signaling protocol functionality

3.1 VN connect and disconnect

When an NVE is external, a protocol is needed between the End Device (e.g. Hypervisor) making use of the external NVE and the external NVE in order to make the NVE aware of the changing VN membership requirements of the Tenant Systems within the End Device.

A key driver for using a protocol rather than using static configuration of the external NVE is because the VN connectivity requirements can change frequently as VMs are brought up, moved and brought down on various hypervisors throughout the data center.

```
+---------------+   Recv VN_connect;        +-------------------+
|VN_Disconnected|   return Local_Tag value  |VN_Connected       |
+---------------+   for VN if successful;   +-------------------+
|VN_ID;         |-------------------------->|VN_ID;             |
|VN_State=      |                           |VN_State=connected;|
|disconnected;  |                           |Num_TSI_Associated;|
|               |<----Recv VN_disconnect----|Local_Tag;         |
+---------------+                           |VN_Context;        |
                                            +-------------------+
```

Figure 2 State machine of a VAP instance on an external NVE

Figure 2 show the state machine for a VAP on the external NVE. An NVE that supports the hypervisor to NVE signaling protocol should support one instance of the state machine for each active VN. The state transition on the external NVE is normally triggered by the hypervisor-facing side events and behaviors. Some of the interleaved interaction between NVE and NVA will be illustrated for better understanding of the whole procedures; while some of them may not be shown. More detailed information regarding that is available in [I-D.ietf-nvo3-nve-nva-cp-req].

The NVE must be notified when an End Device requires connection to a particular VN and when it no longer requires connection. In addition, the external NVE must provide a local tag value for each connected VN to the End Device to use for exchange of packets between the End Device and the NVE (e.g. a locally significant 802.1Q tag value). How "local" the significance is depends on whether the Hypervisor has a direct physical connection to the NVE (in which case the significance is local to the physical link), or whether there is an Ethernet switch (e.g. a blade switch) connecting the Hypervisor to the NVE (in which case the significance is local to the intervening switch and all the links connected to it).

These VLAN tags are used to differentiate between different VNs as packets cross the shared access network to the external NVE. When the NVE receives packets, it uses the VLAN tag to identify the VN of packets coming from a given TSI, strips the tag, and adds the appropriate overlay encapsulation for that VN and send to the corresponding VAP.

The Identification of the VN in this protocol could either be through a VN Name or a VN ID. A globally unique VN Name facilitates portability of a Tenant's Virtual Data Center. Once an NVE receives a VN connect indication, the NVE needs a way to get a VN Context allocated (or receive the already allocated VN Context) for a given VN Name or ID (as well as any other information needed to transmit encapsulated packets). How this is done is the subject of the NVE-to-NVA (called NVE-to-NVA in this document) protocol which are part of work items 1 and 2 in [I-D.ietf-nvo3-overlay-problem-statement].

VN_connect message can be explicit or implicit. Explicit means the hypervisor sending a message explicitly to request for the connection to a VN. Implicit means the external NVE receives other messages, e.g. very first TSI associate message for a given VN as in next subsection, to implicitly indicate its interest to connect to a VN.

A VN_disconnect message will make NVE release all the resources for that disconnected VN and transit to VN_disconnected state. The local

tag assigned for that VN can possibly be reclaimed by other VN.

3.2 TSI associate and activate

Typically, a TSI is assigned a single MAC address and all frames transmitted and received on that TSI use that single MAC address. As mentioned earlier, it is also possible for a Tenant System to exchange frames using multiple MAC addresses or packets with multiple IP addresses.

Particularly in the case of a TS that is forwarding frames or packets from other TSs, the NVE will need to communicate the mapping between the NVE's IP address (on the underlying network) and ALL the addresses the TS is forwarding on behalf of to NVA in each corresponding VN.

The NVE has two ways in which it can discover the tenant addresses for which frames must be forwarded to a given End Device (and ultimately to the TS within that End Device).

1. It can glean the addresses by inspecting the source addresses in packets it receives from the End Device.

2. The hypervisor can explicitly signal the address associations of a TSI to the external NVE. The address association includes all the MAC and/or IP addresses possibly used as source addresses in a packet sent from the hypervisor to external NVE. External NVE may further use this information to filter the future traffic from the hypervisor.

To perform the second approach above, the "hypervisor-to-NVE" protocol requires a means to allow End Devices to communicate new tenant addresses associations for a given TSI within a given VN.

Figure 3 shows the state machine for a TSI connecting to a VAP on the external NVE. An NVE that supports the hypervisor to NVE signaling protocol should support one instance of the state machine for each TSI connecting to a given VN.

```
                disassociate;  +--------+
              +--------------->|  Init  |<--------clear-------+
              |or keepalive    +--------+                     |
              |timer timeout;  |        |                     |
              |                |        |                     |
              |                +--------+                     |
              |                  |    |                       |
              |       associate  |    |  activate             |
              |      +-----------+    +-----------+           |
              |      |                            |           |
              |      |                            |           |
              |     \|/                          \|/          |
      +--------------------+                  +---------------------+
      |     Associated     |                  |       Activated     |
      +--------------------+                  +---------------------+
      |TSI_ID;             |                  |TSI_ID;              |
      |Port;               |-----activate---->|Port;                |
      |VN_ID;              |                  |VN_ID;               |
      |State=associated;   |                  |State=activated ;    |-+
    +-|Num_Of_Addr;        |<---deactivate;---|Num_Of_Addr;          | |
    | |List_Of_Addr;       |  or keepactive    List_Of_Addr;        | |
    | |ResetKeepaliveTimer;|  timer timeout;  |ResetKeepactiveTimer;| |
    | +--------------------+                  +---------------------+ |
    |                    /|\                     /|\                  |
    |                     |                       |                   |
    +---------------------+                       +-------------------+
     add/remove/updt addr;                        add/remove/updt addr;
     or update port; or                           or update port; or
     Recv keepalive pkt                           Recv keepactive pkt
     from TSI;                                    or data msg from TSI;
```

Figure 3 State machine of a TSI instance on an external NVE

Associated state of a TSI instance on an external NVE indicates all the addresses for that TSI have already associated with the VAP of the external NVE on port p for a given VN but no real traffic to and from the TSI is expected and allowed to pass through. NVE has reserved all the necessary resources for that TSI. NVE may report the mappings of NVE's underlay IP address and the associated TSI addresses to NVA and relevant network nodes may save such information to its mapping table but not forwarding table. NVE may create ACL or filter rules based on the associated TSI addresses on the attached port p but not enable them yet. Local tag for the VN corresponding to the TSI instance should be provisioned on port p to receive packets.

VM migration discussed section 2 may cause the hypervisor send associate message to the NVE connecting the destination hypervisor the VM migrates to. It is similar as the resource reservation request

to make sure the VM can be successfully migrated later. If such association fails, VM may choose another destination hypervisor to migrate to or alert with an administrative message. VM creation event may also lead to the same practice.

Activated state of a TSI instance on an external NVE indicates that all the addresses for that TSI functioning correctly on port p and traffic can be received from and sent to that TSI on NVE. The mappings of NVE's underlay IP address and the associated TSI addresses should be put into the forwarding table rather than the mapping table on relevant network nodes. ACL or filter rules based on the associated TSI addresses on the attached port p in NVE are enabled. Local tag for the VN corresponding to the TSI instance MUST be provisioned on port p to receive packets.

Activate message makes the state transit from Init or Associated to Activated. VM creation, VM migration and VM resumption events discussed in section 4 may trigger activate message to be sent from the hypervisor to the external NVE.

As mentioned in last subsection, associate or activate message from the very first TSI connecting to a VN on an NVE is also considered as the implicit VN_connect signal to create a VAP for that VN.

TSI information may get updated either in Associated or Activated state. Add or remove the associated addresses, update current associated addresses for example updating IP for a given MAC, update NVE port information from which the message receives are all considered as TSI information updating. Such update does not change the state of TSI. When any address associated to a given TSI changes, NVE should inform the NVA to update the mapping information on NVE's underlying address and the associated TSI addresses. NVE should also change its local ACL or filter settings accordingly for the relevant addresses. Port information update will cause the local tag for the VN corresponding to the TSI instance provisioned on new port p and removed from old port.

NVE keeps a timer for each TSI instance associated or activated on it. When NVE receives the keepalive or keepactive message for a TSI instance, it should reset the timer. Keepactive timer may also be reset by receiving the data packet from any associated address of the corresponding TSI instance. Keepactive timer times out leads the state transiting from Activated to Associated. Keepalive timer times out leads the state transiting from Associated to Init.

3.3 TSI disassociate, deactivate and clear

Disassociate and deactivate conceptually are the reverse behaviors of

associate and activate. From Activated state to Associated state, NVE needs to make sure the resources still reserved but the addresses associated to the TSI not functioning and no traffic to and from the TSI expected and allowed to pass through. For example, NVE needs to inform NVA to remove the relevant addresses mapping information from forwarding or routing table. ACL or filtering rules regarding the relevant addresses should be disabled. From Associated or Activated state to Init state, NVE will release all the resource relevant to TSI instances. NVE should also inform the NVA to remove the relevant entries from mapping table. ACL or filtering rules regarding the relevant addresses should be removed. Local tag provisioning on the connecting port on NVE should be cleared.

VM suspension discussed in section 2 may cause the relevant TSI instance(s) on NVE transit from Activated to Associated state. VM pause normally does not affect the state of the relevant TSI instance(s) on NVE as the VM is expected to run again soon. VM shutdown will cause the relevant TSI instance(s) on NVE transit to Init state from Activated state. All resources should be released.

VM migration will lead the TSI instance on the source NVE to leave Activated state. Such state transition on source NVE should not occur earlier than the TSI instance on the destination NVE transits to Activated state. Otherwise traffic interruption may occur. When a VM migrates to another hypervisor connecting to the same NVE, i.e. source and destination NVE are the same, NVE should use TSI_ID and incoming port to differentiate two TSI instance.

Although the triggering messages for state transition shown in Figure 3 does not indicate the difference between VM creation/shutdown and VM migration arrival/departure, the NVE can make optimizations if it is notified of such information. For example, if NVE knows the incoming activate message caused by migration rather than VM creation, some mechanisms may be employed or triggered to make sure the dynamic configurations or provisionings on the destination NVE same as those on the source NVE for the migrated VM, for example multicast group memberships.

4. Hypervisor-to-NVE Signaling Protocol requirements

Req-1: The protocol is able to run between the hypervisor and its associated external NVE which may directly connected or bridged in split-NVE architecture.

Req-2: The protocol MUST support the hypervisor initiating a request to its associated external NVE to be connected to a given VN.

Req-3: In response to the connection request to a given VN received

on NVE's port p as per Req-1, the protocol SHOULD support NVE replying a locally significant tag assigned, for example 802.1Q tag value, to each of the VN it is member of. NVE should keep the record of VN ID, local tag assigned and port p triplet.

Req-4: The protocol MUST support the hypervisor initiating a request to associate/disassociate, activate/deactive or clear a TSI instance to a VN on an NVE port. All requests should be logically consistent with text in section 5.2 & 5.3.

Req-5: The protocol MUST support the hypervisor initiating a request to add, remove or update addresses associated with a TSI instance which has associated or activated on the external NVE. Addresses can be expressed in different formats, for example, MAC, IP or pair of IP and MAC.

Req-6: When any request of the protocol fails, a reason code MUST be provided in the reply.

Req-7: The protocol MAY support the hypervisor explicitly informing NVE when a migration starts. It may help NVE to differentiate a new associated/activated TSI resulting from VM creation or VM migration.

Req-8: The protocol SHOULD be extensible to carry more parameters to meet future requirements, for example, QoS settings.

There are multiple candidate protocols probably with some simple extensions that can be used to exchange signaling information between hypervisor and external NVE. They include VDP [IEEE 802.1Qbg], LLDP, XMPP, and HTTP REST. Multiple factors influence the choice of protocol(s), for example, connection between hypervisor and external NVE is L2 or L3. Appendix A illustrates VDP for reader's information.

5. Security Considerations

NVEs must ensure that only properly authorized Tenant Systems are allowed to join and become a part of any specific Virtual Network. In addition, NVEs will need appropriate mechanisms to ensure that any hypervisor wishing to use the services of an NVE are properly authorized to do so. One design point is whether the hypervisor should supply the NVE with necessary information (e.g., VM addresses, VN information, or other parameters) that the NVE uses directly, or whether the hypervisor should only supply a VN ID and an identifier for the associated VM (e.g., its MAC address), with the NVE using that information to obtain the information needed to validate the hypervisor-provided parameters or obtain related parameters in a secure manner.

6. IANA Considerations

No IANA action is required. RFC Editor: please delete this section before publication.

7. Acknowledgements

This document was initiated and merged from the drafts draft-kreeger-nvo3-hypervisor-nve-cp, draft-gu-nvo3-tes-nve-mechanism and draft-kompella-nvo3-server2nve. Thanks to all the co-authors and contributing members of those drafts.

8. References

8.1 Normative References

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

8.2 Informative References

[I-D.ietf-nvo3-overlay-problem-statement] Narten, T., Gray, E., Black, D., Fang, L., Kreeger, L., and M. Napierala, "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-04 (work in progress), July 2013.

[I-D.ietf-nvo3-framework] Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-05 (work in progress), January 2014.

[I-D.ietf-nvo3-nve-nva-cp-req] Kreeger, L., Dutt, D., Narten, T., and D. Black, "Network Virtualization NVE to NVA Control Protocol Requirements", draft-ietf-nvo3-nve-nva-cp-req-01 (work in progress), October 2013.

[I-D.ietf-nvo3-arch] Black, D., Narten, T., et al, "An Architecture for Overlay Networks (NVO3)", draft-narten-nvo3-arch, work in progress.

[I-D.ietf-opsawg-vmm-mib] Asai H., MacFaden M., Schoenwaelder J., Shima K., Tsou T., "Management Information Base for Virtual Machines Controlled by a Hypervisor", draft-ietf-opsawg-vmm-mib-00 (work in progress), February 2014.

[IEEE 802.1Qbg] IEEE, "Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks - Amendment 21: Edge Virtual Bridging", IEEE Std 802.1Qbg, 2012

[8021Q] IEEE, "Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", IEEE Std 802.1Q-2011, August, 2011

Appendix A. IEEE 802.1Qbg VDP Illustration

VDP has the format shown in Figure A.1. Virtual Station Interface (VSI) is an interface to a virtual station that is attached to a downlink port of an internal bridging function in server. VSI's VDP packet will be handled by an external bridge. VDP is the controlling protocol running between the hypervisor and the external bridge.

```
+--------+--------+------+----+----+------+------+------+-----------+
|TLV type|TLV info|Status|VSI |VSI |VSIID | VSIID|Filter|Filter Info|
| 7b     |str len |      |Type|Type|Format|      | Info |           |
|        |  9b    | 1oct |ID  |Ver |      |      |format|           |
|        |        |      |3oct|1oct| 1oct |16oct |1oct  | M oct     |
+--------+--------+------+----+----+------+------+------+-----------+
|                 |      |                       |                  |
|                 |      |<--VSI type&instance-->|<----Filter------>|
|                 |      |<------------VSI attributes-------------->|
|<--TLV header--->|<-------TLV info string = 23 + M octets--------->|
```

Figure A.1: VDP TLV definitions

There are basically four TLV types.

1. **Pre-Associate: Pre-Associate is used to pre-associate a VSI instance** with a bridge port. The bridge validates the request and returns a failure Status in case of errors. Successful pre-association does not imply that the indicated VSI Type or provisioning will be applied to any traffic flowing through the VSI. The pre-associate enables faster response to an associate, by allowing the bridge to obtain the VSI Type prior to an association.

2. **Pre-Associate with resource reservation: Pre-Associate with Resource** Reservation involves the same steps as Pre-Associate, but on successful pre-association also reserves resources in the Bridge to prepare for a subsequent Associate request.

3. **Associate: The Associate creates and activates an association between** a VSI instance and a bridge port. The Bridge allocates any required

bridge resources for the referenced VSI. The Bridge activates the configuration for the VSI Type ID. This association is then applied to the traffic flow to/from the VSI instance.

4. **Deassociate: The de-associate is used to remove an association** between a VSI instance and a bridge port. Pre-Associated and Associated VSIs can be de-associated. De-associate releases any resources that were reserved as a result of prior Associate or Pre-Associate operations for that VSI instance.

Deassociate can be initiated by either side and the rest types of messages can only be initiated by the server side.

Some important flag values in VDP Status field:

1. **M-bit (Bit 5): Indicates that the user of the VSI (e.g., the VM) is** migrating (M-bit = 1) or provides no guidance on the migration of the user of the VSI (M-bit = 0). The M-bit is used as an indicator relative to the VSI that the user is migrating to.

2. **S-bit (Bit 6): Indicates that the VSI user (e.g., the VM) is** suspended (S-bit = 1) or provides no guidance as to whether the user of the VSI is suspended (S-bit = 0). A keep-alive Associate request with S-bit = 1 can be sent when the VSI user is suspended. The S-bit is used as an indicator relative to the VSI that the user is migrating from.

The filter information format currently supports 4 types as the following.

1. **VID Filter Info format**

```
   +---------+------+-------+--------+
   | #of     | PS   | PCP   | VID    |
   |entries  |(1bit)|(3bits)|(12bits)|
   |(2octets)|      |       |        |
   +---------+------+-------+--------+
             |<--Repeated per entry->|
```

Figure A.2 VID Filter Info format

2. MAC/VID filter format

```
   +---------+--------------+------+-------+--------+
   | #of     |  MAC address | PS   | PCP   | VID    |
   |entries  |  (6 octets)  |(1bit)|(3bits)|(12bits)|
   |(2octets)|              |      |       |        |
   +---------+--------------+------+-------+--------+
             |<--------Repeated per entry---------->|
```

Figure A.3 MAC/VID filter format

3. GroupID/VID filter format

```
   +---------+--------------+------+-------+--------+
   | #of     |  GroupID     | PS   | PCP   | VID    |
   |entries  |  (4 octets)  |(1bit)|(3bits)|(12bits)|
   |(2octets)|              |      |       |        |
   +---------+--------------+------+-------+--------+
             |<--------Repeated per entry---------->|
```

Figure A.4 GroupID/VID filter format

4. GroupID/MAC/VID filter format

```
+---------+----------+-------------+------+-----+--------+
| #of     | GroupID  | MAC address | PS   | PCP | VID    |
|entries  |(4 octets)| (6 octets)  |(1bit)|(3b )|(12bits)|
|(2octets)|          |             |      |     |        |
+---------+----------+-------------+------+-----+--------+
          |<-------------Repeated per entry------------->|
```

Figure A.5 GroupID/MAC/VID filter format

The null VID can be used in the VDP Request sent from the hypervisor to the external bridge. Use of the null VID indicates that the set of VID values associated with the VSI is expected to be supplied by the Bridge. The Bridge can obtain VID values from the VSI Type whose identity is specified by the VSI Type information in the VDP Request. The set of VID values is returned to the station via the VDP Response. The returned VID value can be a locally significant value. When GroupID is used, it is equivalent to the VN ID in NVO3. GroupID will be provided by the hypervisor to the bridge. The bridge will map GroupID to a locally significant VLAN ID.

The VSIID in VDP request that identify a VM can be one of the following format: IPV4 address, IPV6 address, MAC address, UUID or locally defined.

We compare VDP against the requirements in the following Figure A.6. It should be noted that the comparison is conceptual. Detail parameters checking is not performed.

<table>
<tr><th>Req</th><th>VDP supported?</th><th colspan="2">remarks</th></tr>
<tr><td>Req-1</td><td>partial</td><td colspan="2">support directly connected but not bridged</td></tr>
<tr><td>Req-2</td><td>Yes</td><td colspan="2">VN is represented by GroupID</td></tr>
<tr><td>Req-3</td><td>Yes</td><td colspan="2">VID=NULL in request and bridge returns the assigned value in response</td></tr>
<tr><td rowspan="4">Req-4</td><td rowspan="4">partial</td><td>requiments</td><td>VDP equivalence</td></tr>
<tr><td>associate/disassociate</td><td>pre-asso/de-asso</td></tr>
<tr><td>activate/deactivate</td><td>associate/nil</td></tr>
<tr><td>clear</td><td>de-associate</td></tr>
<tr><td>Req-5</td><td>partial</td><td colspan="2">VDP can handle MAC addresses properly. For IP addresses, it is not clearly specified.</td></tr>
<tr><td>Req-6</td><td>Yes</td><td colspan="2">Error type indicated in Status in response</td></tr>
<tr><td>Req-7</td><td>Yes</td><td colspan="2">M bit indicated in Status in request</td></tr>
<tr><td>Req-8</td><td>partial</td><td colspan="2">For certain information,e.g. new filter info format, VDP can easily be extended. For some, extensibility may be limited.</td></tr>
</table>

Figure A.6   Compare VDP with the requirements

Authors' Addresses

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Phone: +86-25-56625409
EMail: liyizhou@huawei.com

Lucy Yong
Huawei Technologies, USA

Email: lucy.yong@huawei.com

Lawrence Kreeger
Cisco

Email: kreeger@cisco.com

Thomas Narten
IBM

Email: narten@us.ibm.com
David Black
EMC

Email: david.black@emc.com