Internet Engineering Task Force (IETF)
Request for Comments: 8394
Category: Informational
ISSN: 2070-1721

Y. Li
D. Eastlake 3rd
Huawei Technologies
L. Kreeger
Arrcus, Inc.
T. Narten
IBM
D. Black
Dell EMC

May 2018

Split Network Virtualization Edge (Split-NVE) Control-Plane Requirements

Abstract

In the Split Network Virtualization Edge (Split-NVE) architecture, the functions of the NVE are split across a server and a piece of external network equipment that is called an "External NVE". The server-resident control-plane functionality resides in control software, which may be part of hypervisor or container-management software; for simplicity, this document refers to the hypervisor as the "location" of this software.

One or more control-plane protocols between a hypervisor and its associated External NVE(s) are used by the hypervisor to distribute its virtual-machine networking state to the External NVE(s) for further handling. This document illustrates the functionality required by this type of control-plane signaling protocol and outlines the high-level requirements. Virtual-machine states as well as state transitioning are summarized to help clarify the protocol requirements.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

In the Split Network Virtualization Edge (Split-NVE) architecture shown in Figure 1, the functionality of the NVE is split across an end device supporting virtualization and an external network device that is called an "External NVE". The portion of the NVE functionality located on the end device is called the "tNVE" (terminal-side NVE), and the portion located on the External NVE is called the "nNVE" (network-side NVE) in this document. Overlay encapsulation/decapsulation functions are normally offloaded to the nNVE on the External NVE.

```
                       +------------ Split-NVE ---------+
                       |                                |
                       |                                |
     +-----------------|-----+                          |
     | +---------------|----+|                          |
     | | +--+         \|/   ||                          |
     | | |V |TSI  +-------+ ||                   +------|-------------+
     | | |M |-----+       | ||                   |     \|/            |
     | | +--+     |       | ||                   |+--------+          |
     | | +--+     | tNVE  | ||-------------------||        |          |
     | | |V |TSI  |       | ||                   || nNVE   |          |
     | | |M |-----|       | ||                   ||        |          |
     | | +--+     +-------+ ||                   |+--------+          |
     | |                    ||                   +--------------------+
     | +-----Hypervisor-----+|
     +-----------------------+
            End Device                               External NVE
```

Figure 1: Split-NVE Structure

The tNVE is normally implemented as a part of a hypervisor or container and/or a virtual switch in a virtualized end device. This document uses the term "hypervisor" throughout when describing the Split-NVE scenario where part of the NVE functionality is offloaded to a separate device from the "hypervisor" that contains a VM (Virtual Machine) connected to a VN (Virtual Network). In this context, the term "hypervisor" is meant to cover any device type where part of the NVE functionality is offloaded in this fashion, e.g., a Network Service Appliance or Linux Container.

The Network Virtualization over Layer 3 (NVO3) problem statement [RFC7364] discusses the need for a control-plane protocol (or protocols) to populate each NVE with the state needed to perform the required functions. In one scenario, an NVE provides overlay encapsulation/decapsulation packet-forwarding services to Tenant Systems that are co-resident within 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 the NVE, as the interaction is implemented via software on a single device. However, in the Split-NVE architecture scenarios shown in Figures 2 through 4 (see Section 1.2), one or more control-plane protocols between a hypervisor and its associated External NVE(s) are required for the hypervisor to distribute the VM's networking states to the NVE(s) for further handling. The protocol is an NVE-internal protocol and runs between tNVE and nNVE logical entities. This protocol is mentioned in the "third work area" text in Section 4.5 of the NVO3 problem statement [RFC7364].

VM states and state transitioning are summarized in this document, showing events where the NVE needs to take specific actions. Such events might correspond to actions that the control-plane signaling protocol or protocols need to take between the tNVE and the nNVE in the Split-NVE scenario. The high-level requirements to be fulfilled are listed in Section 4.

To describe the requirements, this document uses VMs as an example of Tenant Systems, 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 is another type of Tenant System, as are systems running on OS-level virtualization technologies like containers. The fact that VMs have lifecycles (e.g., can be created and destroyed, can be moved, and can be started or stopped) results in a general set of protocol requirements, most of which are applicable to other forms of Tenant Systems, although not all of the requirements are applicable to all forms of Tenant Systems.

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

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses the same terminology as the terminology found in [RFC7365]. This section defines additional terminology used by this document.

Split-NVE: A type of NVE (Network Virtualization Edge) where the functionalities are split across an end device supporting virtualization and an external network device.

tNVE: Terminal-side NVE. The portion of Split-NVE functionalities located on the end device supporting virtualization. The tNVE interacts with a Tenant System through an internal interface in the end device.

nNVE: Network-side NVE. The portion of Split-NVE functionalities located on the network device that is directly or indirectly connected to the end device that contains the corresponding tNVE. The nNVE normally performs encapsulation to and decapsulation from the overlay network.

External NVE: The physical network device that contains the nNVE.

Hypervisor: The logical collection of software, firmware, and/or hardware that allows the creation and running of server or service appliance virtualization. The tNVE is located under a hypervisor. The term "hypervisor" is loosely used in this document to refer to the end device supporting the virtualization. For simplicity, we also use the term "hypervisor" to represent both the hypervisor and the container.

Container: Please see "Hypervisor:" above.

VN Profile: Metadata that is associated with a VN and applied to any attachment point to the VN (i.e., VAP (Virtual Access Point) properties that are applied to all VAPs associated with a given VN and used by an NVE when ingressing/egressing packets to/from a specific VN). Metadata could include such information as Access Control Lists (ACLs) and QoS settings. The VN Profile contains parameters that apply to the VN as a whole. Control protocols between the NVE and the NVA (Network Virtualization Authority) could use the VN ID or VN Name to obtain the VN Profile.

VSI: Virtual Station Interface. See [IEEE802.1Q].

VDP: VSI Discovery and Configuration Protocol. See [IEEE802.1Q].

1.2. Target Scenarios

In the Split-NVE architecture, an External NVE can provide offloading of the encapsulation/decapsulation functions and network policy enforcement as well as offloading of overhead from the VN overlay protocol. This offloading may improve performance and/or save resources in the end device (e.g., hypervisor) using the External NVE.

Figures 2 through 4 give example scenarios for the Split-NVE architecture.

```
              Hypervisor             Access Switch
         +------------------+       +-----+-------+
         | +--+   +-------+ |       |     |       |
         | |VM|---|       | | VLAN  |     |       |
         | +--+   | tNVE  |---------+ nNVE|       +--- Underlying
         | +--+   |       | | Trunk |     |       |    Network
         | |VM|---|       | |       |     |       |
         | +--+   +-------+ |       |     |       |
         +------------------+       +-----+-------+
```

Figure 2: Hypervisor with an External NVE

```
             Hypervisor       L2 Switch
         +---------------+     +-----+     +----+---+
         | +--+   +----+ |     |     |     |    |   |
         | |VM|---|    | |VLAN |     |VLAN |    |   |
         | +--+   |tNVE|-------+     +-----+nNVE|   +--- Underlying
         | +--+   |    | |Trunk|     |Trunk|    |   |    Network
         | |VM|---|    | |     |     |     |    |   |
         | +--+   +----+ |     |     |     |    |   |
         +---------------+     +-----+     +----+---+
```

Figure 3: Hypervisor with an External NVE
Connected through an Ethernet Access Switch

```
        Network Service Appliance          Access Switch
     +-----------------------------+      +-----+-------+
     | +---------------+    | \    |      |     |       |
     | |Network Service|----|  \   |      |     |       |
     | |Instance       |    |   \  | VLAN |     |       |
     | +---------------+    |tNVE| |------+nNVE |       +--- Underlying
     | +---------------+    |    | | Trunk|     |       |    Network
     | |Network Service|----|   /  |      |     |       |
     | |Instance       |    |  /   |      |     |       |
     | +---------------+    | /    |      |     |       |
     +-----------------------------+      +-----+-------+
```

Figure 4: Physical Network Service Appliance with an External NVE

Tenant Systems connect to External NVEs via a Tenant System Interface (TSI). The TSI logically connects to the External NVE via a VAP [RFC8014]. The External NVE may provide Layer 2 or Layer 3 forwarding. In the Split-NVE architecture, the External NVE may be able to reach multiple Media Access Control (MAC) addresses and IP addresses via a TSI. An IP address can be in either IPv4 or IPv6 format. For example, Tenant Systems that are providing network services (such as a transparent firewall, load balancer, or VPN gateway) are likely to have a complex address hierarchy. This implies that if a given TSI de-associates from one VN, all the MAC and/or IP addresses are also de-associated. There is no need to signal the deletion of every MAC or IP address 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 as well as a single MAC and IP address visible to the External NVE.

Figures 2 through 4 show the use of VLANs to separate traffic for multiple VNs between the tNVE and the nNVE; VLANs are not strictly necessary if only one VN is involved, but multiple VNs are expected in most cases. Hence, this document assumes the presence of VLANs.

2. VM Lifecycle

Figure 2 of [RFC7666] shows the states and transitions of a VM. Some of the VM states are of interest to the External NVE. This section illustrates the relevant phases and events in the VM lifecycle. Note that the following subsections do not give exhaustive descriptions of VM lifecycle states. Rather, they are intended as illustrative examples that are relevant to the Split-NVE architecture and 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 Event

The VM creation event causes the VM state to transition from the "preparing" state to the "shutdown" state and then to the "running" state [RFC7666]. The end device allocates and initializes local virtual resources like storage in the VM's preparing state. In the shutdown state, the VM has everything ready, except that CPU execution is not scheduled by the hypervisor and the VM's memory is not resident in the hypervisor. The transition from the shutdown state to the running state normally requires human action or a system-triggered event. The running state indicates that the VM is in the normal execution state. As part of transitioning the VM to the running state, the hypervisor must also provision network connectivity for the VM's TSI(s) so that Ethernet frames can be sent and received correctly. Initially, when in the running state, no ongoing migration, suspension, or shutdown is in process.

In the VM creation phase, the VM's TSI has to be associated with the External NVE. "Association" here indicates that the hypervisor and the External NVE have signaled each other and reached some form of agreement. Relevant networking parameters or information have been provisioned properly. The External NVE should be informed of the VM's TSI MAC address and/or IP address. In addition to external network connectivity, the hypervisor may provide local network connectivity between the VM's TSI and TSIs for other VMs that are co-resident on the same hypervisor. When the intra- or inter-hypervisor connectivity is extended to the External NVE, a locally significant tag, e.g., VLAN ID, should be used between the hypervisor and the External NVE to differentiate each VN's traffic. Both the hypervisor and External NVE sides must agree on that tag value for traffic identification, isolation, and forwarding.

The External NVE may need to do some preparation before it signals successful association with the TSI. Such preparation may include locally saving the states and binding information of the TSI and its VN or communicating with the NVA for network provisioning.

A TSI association should be performed before the VM enters the running state, preferably in the shutdown state. If the association with an External NVE fails, the VM should not go into the running state.

2.2. VM Live Migration Event

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, "cold" migration consists of shutting down VM execution on one server and restarting it on another. For simplicity, the following abstract summary of live migration assumes shared storage, so that the VM's storage is accessible to the source and destination servers. Assume that the VM "live migrates" from hypervisor 1 to hypervisor 2. Such a migration event involves state transitions on both source hypervisor 1 and destination hypervisor 2. The VM state on source hypervisor 1 transitions from the running state to the "migrating" state and then to the shutdown state [RFC7666]. The VM state on destination hypervisor 2 transitions from the shutdown state to the migrating state and then to the running state.

The External NVE connected to destination hypervisor 2 has to associate the migrating VM's TSI with itself (i.e., the External NVE) by discovering the TSI's MAC and/or IP addresses, discovering its VN, discovering its locally significant VLAN ID (if any), and provisioning other network-related parameters of the TSI. The External 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. The migrated VM on destination hypervisor 2 should not go to the running state until all the network provisioning and binding have been done.

The VM state on both the source hypervisor and the destination hypervisor will be the migrating state during the transfer of VM execution. The migrating VM should not be in the running state at the same time on the source hypervisor and destination hypervisor during migration. The VM on the source hypervisor does not transition to the shutdown state until the VM successfully enters the running state on the destination hypervisor. It is possible that the VM on the source hypervisor stays in the migrating state for a while after the VM on the destination hypervisor enters the running state.

2.3. VM Termination Event

A VM termination event is also referred to as "powering off" a VM. A VM termination event leads to the VM's transition to the shutdown state. Per [RFC7666], there are two possible causes of VM termination:

1. A running VM has undergone a normal "power-off".

2. The VM has been migrated to another hypervisor, and the VM image on the source hypervisor has to stop executing and be shut down.

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, the External NVE has to de-associate the VM’s TSI.

2.4. VM Pause, Suspension, and Resumption Events

A VM pause event leads to the VM transitioning from the running state to the "paused" state. The paused state indicates that the VM is resident in memory but that CPU execution is not scheduled by the hypervisor [RFC7666]. The VM can be easily reactivated from the paused state to the running state.

A VM suspension event leads to the VM transitioning from the running state to the "suspended" state. A VM resumption event leads to the VM transitioning from the suspended state to the running state. In the suspended state, the memory and CPU execution state of the VM are saved to persistent storage. During this state, CPU execution for the VM is not scheduled by the hypervisor [RFC7666].

In the Split-NVE architecture, the External NVE should not de-associate the paused or suspended VM, as the VM can return to the running state at any time.

3. Hypervisor-to-NVE Control-Plane Protocol Functionality

The following subsections show illustrative examples of the state transitions of an External NVE that are relevant to hypervisor-to-NVE signaling-protocol functionality. Note: This is not prescriptive text for the full state machine.

3.1. VN_Connect and VN_Disconnect

In the Split-NVE scenario, a protocol is needed between the end device (e.g., hypervisor) and the External NVE it is using, in order to make the External 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 that the VN connectivity requirements can change frequently as VMs are brought up, moved, and brought down on various hypervisors throughout the data center or external cloud.

Figure 5 shows the state transition for a VAP on the External NVE. An NVE that supports the hypervisor-to-NVE control-plane protocol should support one instance of the state machine for each active VN. The state transition on the External NVE is normally triggered by events and behaviors on the hypervisor-facing side. Some of the interleaved interactions between the NVE and the NVA will be illustrated to better explain the whole procedure, while other interactions will not be shown.

```
+----------------+   Receive VN_Connect;     +----------------------+
|VN_Disconnected |   return Local_Tag value  |VN_Connected          |
+----------------+   for VN if successful;   +----------------------+
|VN_ID;          |-------------------------->|VN_ID;                |
|VN_State=       |                           |VN_State=VN_Connected;|
|VN_Disconnected;|                           |Num_TSI_Associated;   |
|                |<--Receive VN_Disconnect---|Local_Tag;            |
+----------------+                           |VN_Context;           |
                                             +----------------------+
```

Figure 5: State Transition Example of a VAP Instance on an External NVE

The External NVE must be notified when an end device requires a connection to a particular VN and when it no longer requires a connection. Connection cleanup for the failed devices should be employed. Note that this topic is out of scope for the protocol specified in this document.

In addition, the External NVE should provide a local tag value for each connected VN to the end device to use for exchanging packets between the end device and the External NVE (e.g., a locally significant tag value per [IEEE802.1Q]). How "local" the significance is depends on whether

1. the hypervisor has a direct physical connection to the External NVE (in which case the significance is local to the physical link) or

2. 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 External NVE receives packets, it uses the VLAN tag to identify their VN coming from a given TSI, strips the tag, adds the appropriate overlay encapsulation for that VN, and sends it towards the corresponding remote NVE across the underlying IP network.

The Identification of the VN in this protocol could be through either a VN Name or a VN ID. A globally unique VN Name facilitates portability of a tenant’s virtual data center. Once an External NVE receives a VN_Connect message, the NVE needs a way to get a VN_Context allocated (or to receive the already-allocated VN_Context) for a given VN Name or VN ID (as well as any other information needed to transmit encapsulated packets). How this is done is the subject of the NVE-to-NVA protocol; see the "first two areas of work" text in Section 4.5 of [RFC7364]. The External NVE needs to synchronize the mapping information of the local tag and VN Name or VN ID with the NVA.

The VN_Connect message can be explicit or implicit. "Explicit" means that the hypervisor sends a request message explicitly for the connection to a VN. "Implicit" means that the External NVE receives other messages, e.g., the very first TSI Associate message (see the next subsection) for a given VN, that implicitly indicate its interest in connecting to a VN.

A VN_Disconnect message indicates that the NVE can release all the resources for that disconnected VN and transition to the VN_Disconnected state. The local tag assigned for that VN can possibly be reclaimed for use by another 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 Tenant System that is forwarding frames or packets from other Tenant Systems, the External NVE will need to communicate the mapping between the NVE's IP address on the underlying network and ALL the addresses the Tenant System is forwarding on behalf of the corresponding VN to the NVA.

The NVE has two ways it can discover the tenant addresses for which frames are to be forwarded to a given end device (and ultimately to the Tenant System 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. An address association includes all the MAC and/or IP addresses possibly used as source addresses in a packet sent from the hypervisor to the External NVE. The External NVE may further use this information to filter the future traffic from the hypervisor.

To use the second approach above, the control-plane protocol running between the hypervisor and the NVE must support end devices communicating new tenant-address associations for a given TSI within a given VN.

Figure 6 shows an example of a state transition for a TSI connecting to a VAP on the External NVE. An NVE that supports the hypervisor-to-NVE control-plane protocol may support one instance of the state machine for each TSI connecting to a given VN.

```
                De-Associate   +--------+     De-Associate
              +--------------->|  Init  |<--------------------+
              |                +--------+                     |
              |                |        |                     |
              |                |        |                     |
              |                +--------+                     |
              |                  |    |                       |
              |       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;       |                  |List_Of_Addr;        | |
    | +--------------------+                  +---------------------+ |
    |                    /|\                     /|\                  |
    |                     |                       |                   |
    +---------------------+                       +-------------------+
     add/remove/updt addr;                        add/remove/updt addr;
     or update port;                              or update port;
```

Figure 6: State Transition Example of a TSI Instance on an External NVE

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

The VM migration event (discussed in Section 2) may cause the hypervisor to send an Associate message to the NVE connected to the destination hypervisor of the migration. A VM creation event may also trigger the same scenario.

The Activated state of a TSI instance on an External NVE indicates that all the addresses for that TSI are functioning correctly on a given port, e.g., port p, and traffic can be received from and sent to that TSI via the NVE. The mappings of the NVE’s underlay IP address and the associated TSI addresses should be added to the forwarding table rather than the mapping table on relevant network nodes. ACLs or filter rules based on the associated TSI addresses on the attached port p on the NVE are enabled. The local tag for the VN corresponding to the TSI instance must be provisioned on port p to receive packets.

The Activate message makes the state transition from Init or Associated to Activated. VM creation, VM migration, and VM resumption events (discussed in Section 2) may trigger sending the Activate message from the hypervisor to the External NVE.

TSI information may get updated in either the Associated state or the Activated state. The following are considered updates to the TSI information: add or remove the associated addresses, update the current associated addresses (for example, update the IP address for a given MAC address), and update the NVE port information based on where the NVE receives messages. Such updates do not change the state of the TSI. When any address associated with a given TSI changes, the NVE should inform the NVA to update the mapping information for the NVE’s underlying address and the associated TSI addresses. The NVE should also change its local ACLs or filter settings accordingly for the relevant addresses. Port information updates will cause the provisioning of the local tag for the VN corresponding to the TSI instance on the new port and removal from the old port.

3.3. TSI De-Associate and Deactivate

De-Associate and Deactivate behaviors are conceptually the reverse of Associate and Activate.

From the Activated state to the Associated state, the External NVE needs to make sure the resources are still reserved but the addresses associated with the TSI are not functioning. No traffic to or from the TSI is expected or allowed to pass through. For example, the NVE needs to tell the NVA to remove the relevant information regarding address mapping from the forwarding and routing tables. ACLs and filter rules regarding the relevant addresses should be disabled.

From the Associated or Activated state to the Init state, the NVE releases all the resources relevant to TSI instances. The NVE should also inform the NVA to remove the relevant entries from the mapping table. ACLs or filter rules regarding the relevant addresses should be removed. Local tag provisioning on the connecting port on the NVE should be cleared.

A VM suspension event (discussed in Section 2) may cause the relevant TSI instance(s) on the NVE to transition from the Activated state to the Associated state.

A VM pause event normally does not affect the state of the relevant TSI instance(s) on the NVE, as the VM is expected to run again soon.

A VM shutdown event will normally cause the relevant TSI instance(s) on the NVE to transition to the Init state from the Activated state. All resources should be released.

A VM migration will cause the TSI instance on the source NVE to leave the Activated state. When a VM migrates to another hypervisor connecting to the same NVE, i.e., the source and destination NVE are the same, the NVE should use the TSI_ID and the incoming port to differentiate two TSI instances.

Although the triggering messages for the state transition shown in Figure 6 do not indicate the difference between a VM creation/shutdown event and a VM migration arrival/departure event, the External NVE can make optimizations if it is given such information. For example, if the NVE knows that the incoming Activate message is 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 are the same as those on the source NVE for the migrated VM. For example, an IGMP query [RFC2236] can be triggered by the destination External NVE to the migrated VM so that the VM is forced to send an IGMP report to the multicast router. The multicast router can then correctly route the multicast traffic to the new External NVE for those multicast groups the VM joined before the migration.

4. Hypervisor-to-NVE Control-Plane Protocol Requirements

Req-1: The protocol MUST support a bridged network connecting end devices to the External NVE.

Req-2: The protocol MUST support multiple end devices sharing the same External NVE via the same physical port across a bridged network.

Req-3: The protocol MAY support an end device using multiple External NVEs simultaneously, but only one External NVE for each VN (active-standby External NVE case for a VN).

Req-4: The protocol MAY support an end device using multiple External NVEs simultaneously for the same VN (active-active External NVE case for a VN).

Req-5: The protocol MUST allow the end device to initiate a request to its associated External NVE to be connected/disconnected to a given VN.

Req-6: The protocol MUST allow an External NVE initiating a request to its connected end devices to be disconnected from a given VN.

Req-7: When a Tenant System attaches to a VN, the protocol MUST allow for an end device and its External NVE to negotiate one or more locally significant tags for carrying traffic associated with a specific VN (e.g., tags per [IEEE802.1Q]).

Req-8: The protocol MUST allow an end device initiating a request to associate/de-associate and/or activate/deactivate some or all addresses of a TSI instance to a VN on an NVE port.

Req-9: The protocol MUST allow the External NVE initiating a request to de-associate and/or deactivate some or all addresses of a TSI instance to a VN on an NVE port.

Req-10: The protocol MUST allow an end device initiating a request to add, remove, or update address(es) associated with a TSI instance on the External NVE. Addresses can be expressed in different formats -- for example, MAC, IP, or IP-MAC pair.

Req-11: The protocol MUST allow the External NVE and the connected end device to authenticate each other.

Req-12: The protocol MUST be able to run over Layer 2 links between the end device and its External NVE.

Req-13: The protocol SHOULD support an end device that indicates that an Associate or Activate request from the end device is the result of a VM hot migration event.

5. VDP Applicability and Enhancement Needs

The Virtual Station Interface (VSI) Discovery and Configuration Protocol (VDP) [IEEE802.1Q] can be the control-plane protocol running between the hypervisor and the External NVE. Appendix A provides informative VDP illustrations for the reader.

VDP facilitates the automatic discovery and configuration of Edge Virtual Bridging (EVB) stations and EVB bridges. An EVB station is normally an end station running multiple VMs. In this document, it is considered conceptually equivalent to a hypervisor. An EVB bridge is conceptually equivalent to the External NVE.

VDP is able to pre-associate/associate/de-associate a VSI on an EVB station with a port on the EVB bridge. In the context of this document, a VSI is conceptually approximate to a virtual port by which a VM connects to the hypervisor. The EVB station and the EVB bridge can reach agreement on VLAN ID(s) assigned to a VSI via a VDP message exchange. Other configuration parameters can be exchanged via VDP as well. VDP is carried over the Edge Control Protocol (ECP) [IEEE802.1Q], which provides reliable transportation over a Layer 2 network.

VDP needs some extensions to fulfill the requirements listed in Section 4 of this document. Table 1 shows the needed extensions and/or clarifications in the NVO3 context.

<table>
<tr><th>Req</th><th>Supported by VDP?</th><th>Remarks</th></tr>
<tr><td>Req-1</td><td>Partially</td><td>Needs extension. Must be able to send to a specific unicast MAC, and should be able to send to a non-reserved well-known multicast address other than the nearest customer bridge address.</td></tr>
<tr><td>Req-2</td><td></td><td></td></tr>
<tr><td>Req-3</td><td></td><td></td></tr>
<tr><td>Req-4</td><td></td><td></td></tr>
<tr><td>Req-5</td><td>Yes</td><td>The VN is indicated by GroupID.</td></tr>
<tr><td>Req-6</td><td>Yes</td><td>The bridge sends a De-Associate.</td></tr>
<tr><td>Req-7</td><td>Yes</td><td>VID==NULL in the request. The bridge returns the assigned VLAN ID (VID) value in the response. GroupID, which is optionally present in the request, is equivalent to the VN ID in the context of NVO3. Multiple VLANs per group are allowed.</td></tr>
<tr><td>Req-8</td><td>Partially</td><td><table><tr><th>Requirements</th><th>VDP Equivalent</th></tr><tr><td>Associate/De-Associate
Activate/Deactivate</td><td>Pre-Assoc/De-Associate
Associate/De-Associate</td></tr></table>Needs extension to allow Associate->Pre-Assoc.</td></tr>
<tr><td>Req-9</td><td>Yes</td><td>The VDP bridge initiates a De-Associate.</td></tr>
<tr><td>Req-10</td><td>Partially</td><td>Needs extension for an IPv4/IPv6 address. Add a new "filter information format" type.</td></tr>
<tr><td>Req-11</td><td>No</td><td>An out-of-band mechanism is preferred, e.g., MACsec or 802.1X. Implicit authentication based on control of physical connectivity exists in VDP when the External NVE connects to the end device directly and is reachable with the nearest customer bridge address.</td></tr>
<tr><td>Req-12</td><td>Yes</td><td>VDP naturally runs on the Layer 2 protocol.</td></tr>
</table>

Req-13	Partially	A migration event may cause the M-bit to be set to 1 in the VDP request to the migration destination hypervisor and the S-bit to be set to 1 in the VDP request to the migration source hypervisor. However, a setting of M-bit = 0 or S-bit = 0 can indicate that no information is available regarding migration or that the events in question are not caused by migration. To fully meet the requirement, this ambiguity would need to be fixed so that migration or no migration could be safely inferred from the M-bit or S-bit settings.

Table 1: Comparison of Split-NVE Requirements and VDP Capabilities

By simply adding the ability to carry Layer 3 addresses as per Req-10, VDP can provide most of the hypervisor-to-NVE control-plane functionality required.

6. Security Considerations

External NVEs must ensure that only properly authorized Tenant Systems are allowed to join and become a part of any particular VN. In some cases, the tNVE may want to connect to the nNVE for provisioning purposes. This may require that the tNVE authenticate the nNVE in addition to the nNVE authenticating the tNVE. If a secure channel is required between the tNVE and the nNVE to carry the encrypted Split-NVE control-plane protocol, then existing mechanisms such as MACsec [IEEE802.1AE] can be used. In some deployments, authentication may be implicit, based on control of physical connectivity, e.g., if the nNVE is located in the bridge that is directly connected to the server that contains the tNVE. The use of the "nearest customer bridge address" in VDP [IEEE802.1Q] is an example of where this sort of implicit authentication is possible, although explicit authentication also applies in that case.

As the control-plane protocol results in configuration changes for both the tNVE and the nNVE, tNVE and nNVE implementations should log all state changes, including those described in Section 3. Implementations should also log significant protocol events, such as the establishment or loss of control-plane protocol connectivity between the tNVE and the nNVE, as well as authentication results.

In addition, External NVEs will need appropriate mechanisms to ensure that any hypervisor wishing to use the services of an NVE is properly authorized to do so. One design point is whether the hypervisor should

1. supply the External NVE with necessary information (e.g., VM addresses, VN information, or other parameters) that the External NVE uses directly or

2. only supply a VN ID and an identifier for the associated VM (e.g., its MAC address), with the External NVE using that information to obtain the information needed to validate the hypervisor-provided parameters or obtain related parameters in a secure manner.

The former approach can be used in a trusted environment so that the External NVE can directly use all the information retrieved from the hypervisor for local configuration. It relieves the External NVE side of effort related to information retrieval and/or validation. The latter approach gives more reliable information, as the External NVE needs to retrieve it from a management-system database. In particular, some network-related parameters, such as VLAN IDs, can be passed back to the hypervisor to be used as a form of provisioning that is more authoritative. However, in certain cases it is difficult or inefficient for an External NVE to be granted rights to access or query information on those management systems. The External NVE then has to obtain the information from the hypervisor.

7. IANA Considerations

This document has no IANA actions.

8. References

8.1. Normative References

[IEEE802.1Q]
IEEE, "IEEE Standard for Local and metropolitan area networks--Bridges and Bridged Networks", IEEE Standard 802.1Q-2014, DOI 10.1109/IEEESTD.2014.6991462.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC7365] Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for Data Center (DC) Network Virtualization", RFC 7365, DOI 10.17487/RFC7365, October 2014, <https://www.rfc-editor.org/info/rfc7365>.

[RFC7666] Asai, H., MacFaden, M., Schoenwaelder, J., Shima, K., and T. Tsou, "Management Information Base for Virtual Machines Controlled by a Hypervisor", RFC 7666, DOI 10.17487/RFC7666, October 2015, <https://www.rfc-editor.org/info/rfc7666>.

[RFC8014] Black, D., Hudson, J., Kreeger, L., Lasserre, M., and T. Narten, "An Architecture for Data-Center Network Virtualization over Layer 3 (NVO3)", RFC 8014, DOI 10.17487/RFC8014, December 2016, <https://www.rfc-editor.org/info/rfc8014>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

8.2. Informative References

[IEEE802.1AE]
IEEE, "IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Security", IEEE Standard 802.1AE-2006, DOI 10.1109/IEEESTD.2006.245590.

[NVO3-HYPERVISOR-NVE-CP]
Kreeger, L., Narten, T., and D. Black, "Network Virtualization Hypervisor-to-NVE Overlay Control Protocol Requirements", Work in Progress, draft-kreeger-nvo3-hypervisor-nve-cp-01, February 2013.

[NVO3-TES-NVE]
Yingjie, G. and L. Yizhou, "The mechanism and signalling between TES and NVE", Work in Progress, draft-gu-nvo3-tes-nve-mechanism-01, October 2012.

[NVO3-VM-NVE]
Kompella, K., Rekhter, Y., Morin, T., and D. Black, "Signaling Virtual Machine Activity to the Network Virtualization Edge", Work in Progress, draft-kompella-nvo3-server2nve-02, April 2013.

[RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, DOI 10.17487/RFC2236, November 1997, <https://www.rfc-editor.org/info/rfc2236>.

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, DOI 10.17487/RFC4122, July 2005, <https://www.rfc-editor.org/info/rfc4122>.

[RFC7364] Narten, T., Ed., Gray, E., Ed., Black, D., Fang, L., Kreeger, L., and M. Napierala, "Problem Statement: Overlays for Network Virtualization", RFC 7364, DOI 10.17487/RFC7364, October 2014, <https://www.rfc-editor.org/info/rfc7364>.

Appendix A. VDP Illustrations (per IEEE 802.1Q) (for Information Only)

VDP (the VSI Discovery and Discovery and Configuration Protocol; see Clause 41 of [IEEE802.1Q]) can be considered as a controlling protocol running between the hypervisor and the external bridge. The VDP association TLV structure is formatted as shown in Figure 7.

```
+--------+--------+------+-----+--------+------+------+------+------+
|TLV Type|TLV Info|Status|VSI  |VSI Type|VSI ID|VSI ID|Filter|Filter|
|        |String  |      |Type |Version |Format|      |Info  |Info  |
|        |Length  |      |ID   |        |      |      |Format|      |
+--------+--------+------+-----+--------+------+------+------+------+
|                 |      |<--VSI Type and instance--->|<--Filter--->|
|                 |      |<-------------VSI attributes------------->|
|<--TLV header--->|<-----------TLV information string ------------->|
```

Figure 7: VDP Association TLV

There are basically four TLV types.

1. Pre-Associate: The 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 the case of errors. A successful Pre-Associate does not imply that the indicated VSI Type or provisioning will be applied to any traffic flowing through the VSI. By allowing the bridge to obtain the VSI Type prior to an association, the Pre-Associate enables faster response to an Associate.

2. Pre-Associate with Resource Reservation: The Pre-Associate with Resource Reservation involves the same steps as those for the Pre-Associate, but on success it also reserves resources in the bridge to prepare for a subsequent Associate request.

3. Associate: The Associate request creates and activates an association between a VSI instance and a bridge port. A 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. De-Associate: 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. The De-Associate releases any resources that were reserved as a result of prior Associate or Pre-Associate operations for that VSI instance.

The De-Associate can be initiated by either side, and the other types can only be initiated by the server side.

Some important flag values in the VDP Status field are as follows:

1. M-bit (Bit 5): M-bit = 1: indicates that the user of the VSI (e.g., the VM) is migrating. M-bit = 0: no indication of whether the VSI user is migrating. The M-bit is used as an indicator relative to the VSI to which the user is migrating.

2. S-bit (Bit 6): S-bit = 1: indicates that the VSI user (e.g., the VM) is suspended. S-bit = 0: no indication of whether the VSI user is suspended. 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 from which the user is migrating.

The filter information format currently defines four types. Information for each of these types is shown in detail in Figures 8 through 11. "PCP" stands for Priority Code Point [IEEE802.1Q]. The PCP value, if specified, is used by the EVB station as the default PCP value associated with the VSI and VID. The filter information contains a PCP Significant (PS) bit associated with each PCP field, indicating whether the PCP field carries a PCP value (binary 1) or does not carry a PCP value (binary 0).

```
                 +----------+-------+--------+--0------+
                 | # of     | PS    | PCP    | VID     |
                 |entries   |(1 bit)|(3 bits)|(12 bits)|
                 |(2 octets)|       |        |         |
                 +----------+-------+--------+---------+
                            |<---Repeated per entry--->|
```

Figure 8: VID Filter Information Format

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

Figure 9: MAC/VID Filter Information Format

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

Figure 10: GroupID/VID Filter Information Format

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

Figure 11: GroupID/MAC/VID Filter Information Format

The null VID can be used in the VDP Request sent from the station to the external bridge. The null VID indicates that the set of VID values associated with the VSI is expected to be supplied by the bridge. The set of VID values is returned to the station via the VDP Response. The returned VID values can be locally significant values. When GroupID is used, it is equivalent to the VN ID in NVO3. GroupID will be provided by the station to the bridge. The bridge maps GroupID to a locally significant VLAN ID.

The VSI ID in the VDP association TLV that identifies a VM can be in one of the following formats: IPv4 address, IPv6 address, MAC address, Universally Unique Identifier (UUID) [RFC4122], or locally defined.

Acknowledgements

This document was initiated based on the merger of the following documents: [NVO3-HYPERVISOR-NVE-CP], [NVO3-TES-NVE], and [NVO3-VM-NVE]. Thanks to all the coauthors and contributing members of those documents.

The authors would like to specially thank Lucy Yong and Jon Hudson for their generous help in improving this document.

Authors' Addresses

Yizhou Li
Huawei Technologies
101 Software Avenue
Nanjing 210012
China

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

Donald Eastlake 3rd
Huawei R&D USA
155 Beaver Street
Milford, MA 01757
United States of America

Phone: +1-508-333-2270
Email: d3e3e3@gmail.com

Lawrence Kreeger
Arrcus, Inc.

Email: lkreeger@gmail.com

Thomas Narten
IBM

Email: narten@us.ibm.com

David Black
Dell EMC
176 South Street
Hopkinton, MA 01748
United States of America

Email: david.black@dell.com