Network Working Group Y. Gu
Internet-Draft Huawei
Intended status: Standards Track July 6, 2012
Expires: January 7, 2013

The mechanism and protocol between VAP and TES to facilitate NVO3
draft-gu-nvo3-tes-nve-mechanism-00

Abstract

This draft provide requirements and rationale for the mechanism between Network Virtualization Edge (NVE) and Tenant End Systems (TES). It also provides analysis on candidate mechanisms to fulfill the requirements. These requirements is complementarity to the control plane requirements. In the subsequent versions of this draft, we will provide more technical details.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

This draft is not the first draft that discuss the requirements for a mechanism and protocol between NVE and TES. To simplify the description of this mechanism and protocol in this draft, we temporarily call it TES to NVE Notification mechanism and Protocol (TNP). The TNP is shown in the reference model in Fig1.

```
                       +------- L3 Network ------+
                       |                         |
                       |       Tunnel Overlay    |
          +------------+--------+       +--------+------------+
          | +----------+------+ |       | +------+----------+ |
          | | Overlay Module  | |       | | Overlay Module  | |
          | +--------+--------+ |       | +--------+--------+ |
          |          |VN Context|       |          |VN Context|
          |          |          |       |          |          |
          |  +-------+-------+  |       |   +------+-------+  |
          |  |    VNI        |  |       |   |     VNI      |  |
     NVE1 |  +-+-----------+-+  |       |   +-+----------+-+  | NVE2
          |    |   VAPs    |    |       |    |   VAPs    |    |
          +----+-----------+----+       +----+-----------+----+   /\
               |           |                 |           |        |
        -------+-----------+-----------------+-----------+------- |  TNP
               |           |     Tenant      |           |        |
               |           |   Service IF    |           |        \/
             Tenant End Systems                Tenant End Systems
```

Figure 1: Generic reference model for NV Edge

In Section 3 of , VN connect/disconnect notification have been introduced, which is one requirement for TNP. Some characteristic described in Section 4 of [overlay-cp]can also be extracted as requirements on TNP. [overlay-cp]Also in , the author's consideration on local VID issues makes us to think of how TNP can resolve this problem. [mobility]

Before list the requirements for TNP, some necessary functionality for network virtualization are introduced which are regarded, in this draft, as origins of TNP. Then specific requirements are listed.

In the mail list discussion, at least three mechanisms are mentioned, i.e. VDP (VSI Discovery and Configuration Protocol), existing IETF protocols, e.g. XMPP and OSPF, and centralized controller assistance. It's hard to compare all the possible protocols in one draft, so we classify the protocols and match each classification to the requirements and analyze whether and how the different classification can fulfill these requirements.

In the subsequent versions of this draft, we will provide more technical details for respective mechanism. We expect to keep updating the draft to reflect the relevant consensus made by working group and will define the protocol and mechanism in detail.

2. Terminologies and concepts

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 [RFC2119].

The document uses terms defined in [framework]and [overlay-cp].

VN: Virtual Network. This is a virtual L2 or L3 domain that belongs a tenant.

VNI: Virtual Network Instance. This is one instance of a virtual overlay network. Two Virtual Networks are isolated from one another and may use overlapping addresses.

Virtual Network Context or VN Context: Field that is part of the overlay encapsulation header which allows the encapsulated frame to be delivered to the appropriate virtual network endpoint by the egress NVE. The egress NVE uses this field to determine the appropriate virtual network context in which to process the packet. This field MAY be an explicit, unique (to the administrative domain) virtual network identifier (VNID) or MAY express the necessary context information in other ways (e.g. a locally significant identifier).

VNID: Virtual Network Identifier. In the case where the VN context has global significance, this is the ID value that is carried in each data packet in the overlay encapsulation that identifies the Virtual Network the packet belongs to.

NVE: Network Virtualization Edge. It is a network entity that sits on the edge of the NVO3 network. It implements network virtualization functions that allow for L2 and/or L3 tenant separation and for hiding tenant addressing information (MAC and IP addresses). An NVE could be implemented as part of a virtual switch within a hypervisor, a physical switch or router, a Network Service Appliance or even be embedded within an End Station.

Underlay or Underlying Network: This is the network that provides the connectivity between NVEs. The Underlying Network can be completely unaware of the overlay packets. Addresses within the Underlying Network are also referred to as "outer addresses" because they exist

in the outer encapsulation. The Underlying Network can use a completely different protocol (and address family) from that of the overlay.

Data Center (DC): A physical complex housing physical servers, network switches and routers, Network Service Appliances and networked storage. The purpose of a Data Center is to provide application and/or compute and/or storage services. One such service is virtualized data center services, also known as Infrastructure as a Service.

VM: Virtual Machine. Several Virtual Machines can share the resources of a single physical computer server using the services of a Hypervisor (see below definition).

Hypervisor: Server virtualization software running on a physical compute server that hosts Virtual Machines. The hypervisor provides shared compute/memory/storage and network connectivity to the VMs that it hosts. Hypervisors often embed a Virtual Switch (see below).

Virtual Switch: A function within a Hypervisor (typically implemented in software) that provides similar services to a physical Ethernet switch. It switches Ethernet frames between VMs' virtual NICs within the same physical server, or between a VM and a physical NIC card connecting the server to a physical Ethernet switch. It also enforces network isolation between VMs that should not communicate with each other.

Tenant: A customer who consumes virtualized data center services offered by a cloud service provider. A single tenant may consume one or more Virtual Data Centers hosted by the same cloud service provider.

Tenant End System: It defines an end system of a particular tenant, which can be for instance a virtual machine (VM), a non-virtualized server, or a physical appliance.

Virtual Access Points (VAPs): Tenant End Systems are connected to the Tenant Instance through Virtual Access Points (VAPs). The VAPs can be in reality physical ports on a ToR or virtual ports identified through logical interface identifiers (VLANs, internal VSwitch Interface ID leading to a VM).

VN Name: A globally unique name for a VN. The VN Name is not carried in data packets originating from End Stations, but must be mapped into an appropriate VN-ID for a particular encapsulating technology. Using VN Names rather than VN-IDs to identify VNs in configuration files and control protocols increases the portability of a VDC and

its associated VNs when moving among different administrative domains (e.g. switching to a different cloud service provider).

VSI: Virtual Station Interface. Typically, a VSI is a virtual NIC connected directly with a VM. [Qbg]

3. Origin of Requirements

The main target of NVO3 working group is to define mechanisms to support multi-tenancy in DC, including traffic isolation, address independence, and support the placement and migration of VMs. The NVE is the boundary between TES and VN, while VAP component in NVE is the point that directly access to TES. TES can be either VM or physical server, and NVE can be within external network entity or within Hypervisor.

NVE Location Cases

- (NVE Location 1) Physical server connects to external NVE: In this case, each physical server represents one end system of a tenant. There is no protocol or external assistance required since NVE can be notified by physical plugin and extraction. And no need to consider server mobility.

```
                         +------- L3 Network ------+
                         |                         |
                         |       Tunnel Overlay    |
           +-------------+--------+       +--------+-------------+
           | +------------------+ |       | +------------------+ |
           | |  Overlay Module  | |       | |  Overlay Module  | |
           | +---------+--------+ |       | +---------+--------+ |
           |           |VN context|       | VN context|          |
           |           |          |       |           |          |
           |  +--------+-------+  |       |  +--------+-------+  |
           |  |     VNI        |  |       |  |       VNI      |  |
      NVE1 |  +-+------------+-+  |       |  +-+-----------+--+  | NVE2
           |    |   VAPs     |    |       |    |    VAPs   |     |
           +----+------------+----+       +----+-----------+-----+
                |            |                 |           |
         -------+------------+-----------------+-----------+-------
                |            |     Tenant      |           |
                |            |   Service IF    |           |
              Tenant End Systems             Tenant End Systems
                Physical Server               Physical Server
```

Figure 2: NVE Location1: Physical server connects to external NVE

- (NVE Location 2) VM connects to NVE on Hypervisor: In this case, there should be some mechanism to assist Hypervisor know of VM changes, including adding, deleting and migration. Both VM and Hypervisor, as well as network service appliance, are controlled by VM Manager. VM Manager are aware of any VM identity and changes, hence can easily notify NVE about the information. So a protocol is not necessary in this case.

```
           +-------------+------------+
           |  +--------------------+  |
           |  |  +--------------+  |  |
           |  |  |Overlay Module|  |  |
           |  |  +----+---------+  |  |
           |  |        | VN context|  |
           |  |  +-----+-------+   |  |
           |  |  |      VNI    |   |  |
           |  |  +-+---------+-+   |  |
           |  |    |  VAPs   |     |  |
           |  +----+---------+-----+  |
           |       |         |        |
           |    +--+---------+---+    |
           |    |       VM       |    |
           |    +----------------+    |
           |                          |
           +--------------------------+
               Tenant End Systems
```

Figure 3

- (NVE Location 3) VM connects to NVE on external network entity: VM is controlled by VM Manager, while NVE is controlled by external controller. There is more than one way to enable NVE know of VM changes. One is enable protocol between TES and NVM, which indicates VM identity and changes. The another way is to enable NVE to get from or being notified by external controller that can communicate with VM Manager.

```
                        +------- L3 Network --------+
                        |                           |
                        |       Tunnel Overlay      |
           +------------+---------+       +---------+------------+
           | +----------+-------+ |       | +---------+--------+ |
           | |  Overlay Module  | |       | |  Overlay Module  | |
           | +---------+--------+ |       | +---------+--------+ |
           |           |VN context|       | VN context|          |
           |           |          |       |           |          |
           |  +--------+-------+  |       |  +--------+-------+  |
           |  |     VNI        |  |       |  |       VNI      |  |
      NVE1 |  +-+------------+-+  |       |  +-+-----------+--+  | NVE2
           |    |   VAPs     |    |       |    |    VAPs   |     |
           +----+------------+----+       +----+-----------+-----+
                |            |                 |           |
         -------+------------+-----------------+-----------+-------
                |            |     Tenant      |           |
                |            |   Service IF    |           |
           +----+------------+--------+    +---+-----------+-------+
           |   +----------------+     |    |  +---------------+    |
           |   |   Hypervisor   |     |    |  |  Hypervisor   |    |
           |   +--------+-------+     |    |  +-------+-------+    |
           |            |             |    |          |            |
           |    +-------+------+      |    |   +------+------+     |
           |    |       VM     |      |    |   |     VM      |     |
           |    +--------------+      |    |   +-------------+     |
           |                          |    |                       |
           +--------------------------+    +-----------------------+

               Tenant End Systems            Tenant End Systems
```

Figure 4: NVE Location3: VM connects to NVE on external network entity

In this draft, we focus on the requirements on the the third case, that is VM connects to NVE on external network entity.

3.1. Basic Functionality

3.1.1. Connectivity Notification

As described in , the basic functionality for VAP mechanism is connectivity notification. [overlay-cp]

VM has several status, including Created, Start up, Running, Shut Down, Removed, Emigration, Immigration.

- Created: A set of OS resources is assigned for a VM. The VM is ready to work, but not yet, like a well composed physical server with power off. The VM Manager has configured the VM-relevant information on Hypervisor

- Start up: The VM begins to work, just like a physical server is power on. The VM must be able to connect to the network and send packets at any time it wants. The Hypervisor should be aware of the VM's start up.

- Running: The VM is working, e.g. communicating with remote end systems, processing local applications, etc.

- Shut down: The VM is power off and no more network connection at this time, though the OS resources is still reserved for the VM, just like a physical server is power off.

- Removed: The set of OS resources assigned for this VM is released, which means the VM doesn't exist any more.

  - (Usually a full life of VM will pass through the following stages: Created--Start up--Running--Shut down--Removed.)

- Emigration: The VM is still alive. But the VM and the services running on the VM, if there is, migrate to another TES within or out of the DC. There is no Shut down and Remove stage for this VM on the original TES.

- Immigration: The VM is migrated from another TES within or out of the DC. It doesn't have to experience the stage of Created and Start up. There two kinds of Immigration VM, one is VM clean immigration which has no running service need to resume, and the other is VM live migration which has running service on it and the service should not be disrupted by the action of VM migration.

A Hypervisor is able to know every status of the VM, but the NVE needn't know each of them. In order not to interrupt the VM's network communication, NVE must be aware of these status: Start up, Shut down, Emigration and Immigration. And NVE must be aware of these status timely.

When it is notified of the VM's Start up and Immigration, NVE can get the specific VNID that VM is associated with, by using any information that can identify this VM and it's VN, e.g. VN Name, VSI ID, MAC address, etc.. We call all the above information as VN Clue. And then NVE can get the globally unique VNID and timely created local mapping, and forwarding table. The way to get these table information is discussed in dedicated drafts.

When it is notified of the VM's Shut down and Emigration, NVE can timely remove the local mapping and forwarding information to reduce the chance for mis-forwarding and to save the space on the NVE.

3.1.2. Inner Address Notification

One important feature of multi-tenant DC is to support overlapped IP address among different Tenant. In , the requirements for inner to outer address mapping is described. In order to make this mapping, NVE must be aware of the inner address, e.g. the MAC address of VM's virtual NIC or VM's IP address. [overlay-cp]

3.1.3. Local Tag Feedback

After NVE get the VNID associated with a VM, it needs to assign a local tag for the VNID. The local tag could be an IEEE802.1Q tag if the TES access to NVE via L2. In every data frame sent from TES to NVE, the local tag is included and NVE can easily define which VNID the data frame belong to. The data frame sent from VM is usually untagged, and Hypervisor in TES will add the local tag to the data frame. Since local tag is assigned by NVE, it's necessary to find a way to notify Hypervisor of the local tag.

```
VM          Hypervisor                       NVE             Controller
|               |                             |
|---Start up--->|                             |
                |--Start up (with VN Clue)--->|-------Get VNID-->|
                                              |<---VNID----------|
                |<--------local tag-----------|
|-data frame--->|
                |-data frame(with local tag)->|
                                              |--encapsulated VN traffic->
```

Figure 5: Local Tag Feedback

But it's also possible that the data frame from VM is already tagged with VID, then adjacent bridge, either virtual or physical, need to conver this VID to local tag. In our case, i.e. NVE location 3, the adjacent bridge is a virtual bridge on Hypervisor, and Hypervisor has to convert the VID to the local tag provided by NVE.

NVE should also support untagged traffic from TES. In this case, NVE should be able to get the specific VNID from its controller for untagged data frames arriving at its Virtual Access Points.

3.2. Extension Functionality

3.2.1. PCP notification

In typical DC, where physical server connects to adjacent bridge, the data frame from server can be tagged with PCP or untaggged. If a data frame is untagged, it can be tagged with PCP on adjacent bridge. While in virtualized DC, the adjacent bridge is Hypervisor. There are two options to deal with PCP tag, 1) data frame is tagged with PCP by VM, 2)data frame is tagged with PCP by Hypervisor and 3) data frame is tagged with PCP by NVE.

In cloud service, the VM can be anybody and it may want a higher priority than it should have. The VM can tag it's data frame with higher PCP value and get better service. Based on the assumption that PCP provided by VM is not reliable, it's more reasonable to let the network to define the PCP value based on VM's priority, and enable bridges to tag the PCP value, as 2) or 3).

This problem is similar to local VID, which can be tagged either by Hypervisor or by NVE. The benefit to tag PCP by Hypervisor is to reduce the load on NVE.

4. Requirement

4.1. Basic Requirements

REQUIREMENT-1: The TNP (TES to NVE notification mechanism and protocol) MUST support TES to notify NVE about the VM's status, including but not limited to Start up, Shut down, Emigration and Immigration.

REQUIREMENT-2: The TNP MUST support TES to notify NVE about the VM's VN Clue, which can be one identifier or a combination of several indentifier.

REQUIREMENT-3: The TNP MUST support TES to notify NVE about the VM's inner address. The inner address MUST include one or both of MAC address of VM's virtual NIC and VM's IP address. And it SHOULD be extensible to carry new address type.

REQUIREMENT-4: The TNP MUST support NVE to notify TES about the VM's local tag. The local Tag type supported by TNP MUST include IEEE 802.1Q tag. And it SHOULD be extensible to carry other type of local tag.

4.2. Extension Requirements

REQUIREMENT-5: The TNP SHOULD support NVE to notify TES about the VM's traffic PCP value.

5. Mechanism Classification

5.1. Mechanism Classification

At least three kinds of mechanisms are discussed in the mail list.

- o IEEE 802.1Qbg- VDP(VSI discovery and configuration protocol): VDP is a new protocol defined in IEEE802.1, the original intention is to enable adjacent bridge to discover the connection/remove/ migration of VM's. And it also enable configuration between adjacent bridge and Hypervisors. We will disclose more in the following sections.

- o Assist by external controller: The external controller can communicate with VM Manager, who is totally aware of the VM's status and private information, including VSI ID, MAC address, IP address and etc. The controller can get enough information from the VM Manger and then configure and control NVE.

- o Existing IETF protocols, e.g. XMPP, OSPF and BGP: The basic idea is to reuse these protocol to transfer the VN Clue to NVE. It's not necessary to implement the complete protocols. The usage of these protocols doesn't belong to their original definitions, but only utilize their ability to carry additional information.

In this section, we will analyze each mechanism classification, how they match the requirements listed in previous section.

5.2. IEEE 802.1Qbg

5.2.1. Brief Introduction

VDP has four basic TLV types.

- o 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 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.

- o 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.

- o Associate: The Associate TLV Type 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.

- o Deassociate: The de-associate TLV Type 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.

```
|1        |2       |3       |4       |7       |8     |9      |25         |
26          |25+M
|---------+--------+--------+--------+--------+------+-------+-----------
+------------|
|TLV type|TLV info | Status |VSI Type|VSI Type|VSIID |VSIID  |Filter Info|
Filter Infor|
|(7bits) |strlength|(1octet)|  ID    |version |format|(16oct)|   format  | (M
octets) |
|        | (9bits) |        |(3oct)  |(1oct)  |(1oct)|       |  (1
octet)|            |
|--------+---------+--------+--------+--------+------+-------+-----------
+------------|
|                            |<-------VSI type&instance------>|<-------
Filter----------|
|                            |<--------------------VSI
attibutes----------------------->|
|<----TLV header--><--------------TLV information string =
23+Moctets---------------->|
```

Figure 6: VDP TLV definitions

Some important flag values in VDP request:

- o 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.

- o 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 field supports the following format:

- VID

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

Figure 7

o MAC/VID

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

Figure 8

o GroupID/VID

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

Figure 9

o GroupID/MAC/VID

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

Figure 10

In each format, the null VID can be used in the VDP Request. In this case, the Bridge is expected to supply the corresponding local VID value in the VDP Response.

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.

VDP features	Requirements Matching
Pre-Associate/ Pre-Associate with resource reservation/ Associate/ Deassociate	Requirement-1
M-bit/S-bit	Requirement-1
VSI type&instance in VDP request	Requirement-2
Filter Infor	Requirement-3
VID infor in VDP response	Requirement-4
PCP in VDP response	Requirement-5

VDP TLV types

5.3. External Controller

5.4. Reuse existing protocols

6. Security Considerations

There are some considerations on security in . Most of the considerations are about mechanism between NVE and external controller, and the attack on underlying networks, which can not be resolved only by the mechanism between TES and NVE. One security issue related to the mechanism between TES and NVE is about the authentication of VM who announces to associate with a particular VN. There is a hypervisor between VMs and NVEs, and both VMs and hypervisor are not always reliable. For example, a poisoned hypervisor may modify the VN Name, or identification for similar intention, in order to associate with a VN that it doesn't belong to. [overlay-cp]

7. References

7.1. Normative Reference

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

[Qbg] "IEEE P802.1Qbg Edge Virtual Bridging".

7.2. Informative Reference

[framework]
Marc Lasserre, Marc., Balus, Florin., Morin, Thomas.,

Bitar, Nabil., and Yakov. Rekhter, "draft-lasserre-nvo3-framework-02", June 2012.

[overlay-cp]
Kreeger, L., Dutt, D., Narten, T., Black, D., and M. Sridharan, "draft-kreeger-nvo3-overlay-cp-00", Jan 2012.

[mobility]
Dunbar, L., "draft-dunbar-nvo3-overlay-mobility-issues-00", June 2012.

Author's Address

Gu Yingjie
Huawei
No. 101 Software Avenue
Nanjing, Jiangsu Province 210001
P.R.China

Phone: +86-25-56625392
Email: guyingjie@huawei.com