Network Working Group Y. Gu
Internet-Draft W. Hao
Intended status: Standards Track Huawei
Expires: January 10, 2013 July 9, 2012

Analysis of external assistance to NVE and consideration of architecture draft-gu-nvo3-overlay-cp-arch-00

Abstract

Draft [overlay-cp] has introduced some control plan requirements and characteristics. From NVE's perspective, this draft describes what assistance is needed to make NVE satisfy the requirements and characteristics introduce in [overlay-cp]. Not all of these assistance is necessarily achieved by an external controller. Some of the assistance requirements can be regarded as a complementarity requirements to [overlay-cp] . while others are requirements to an assistance Database. This draft also provide considerations on how the network virtualization architecture should be like and how these assistance can be fulfilled. The target is to help the working group to figure out the architecture of overlay control plane, instead of providing solutions.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 10, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

Draft [overlay-cp] has introduced some control plan requirments and characteristics. From NVE's perspective, this draft describes what assistance is needed to make NVE statisfy the requirements and characteristics introduce in [overlay-cp]. Not all of these assistance is necessarily acheived by an external controller. Some of the assistance requirements can be regarded as a complementarity requirements to [overlay-cp] . while others are requirements to an assistance Database. This draft also provide considerations on how the network virtualization architecture should be and how these assistance can be fulfilled. The target is to help the working group to figure out the architecture of overlay control plane, instead of providing solutions.

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. The fundamental requirements and characteristics

In this section, we make a summary of the fundamental requirements and characteristics made in [overlay-cp].

Summary of requirements:

- Inner to Outer address mapping
- Underlying Network Multi-Destination Delivery Address(es)
- VN Connect/Disconnect Notification
- VN Name to VN-ID Mapping

Summary of characteristics:

- As few local caching state as better
- Fast acquisition of needed state
- Fast detection/update of stale cached state information
- Minimize processing overhead
- Highly scalable
- Minimize the complexity of the implementation
- Extensible

- Simple protocol configuration
- Do not rely on IP Multicast
- Flexible mapping sources

3.1. Assistance to NVE

In this section, we describe the assistance to NVE as an addition to the requirements enumerated in the above section. Meanwhile the additional requirements must satisfy the required characteristic. We call it assistance, instead of control plane requirements, since the assistance can be achieved by a controller, or a database, which is not traditionally in concept of control plane.

In following section, more than one options to enable these assistance are introduced. No matter what kind of control plane components are finally adopted by the working, the assistance requirements must be satisfied.

3.1.1. Assistance from TES

In draft [tes-nve-mechanism], some requirements and possible mechanisms to enable the requirements are described. These requirements are the assistance that TES can provides, maybe together with external entities, e.g. controllers or profile Database. A summary is enumerated here.

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.

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

The following sections are the assistance the NVE needs but can be provided by entities other than TES, e.g. by an external controller or a database. These assistance requirements are complementarity to those introduced in . [overlay-cp]

3.2. Access Control List

While VAP identify the a new membership, be a VM or a physical server, NVE needs to get the Access Control List to the member. The ACL maybe associate with a specific member or associate with a specific VNI. If the ACL is associate with a specific VNI, NVE only needs to get the ACL at the first time the NVE is associate with the VNI.

If the ACL changes, e.g. rules change or deleting, the assistance subject must be able to notify NVE to update the ACL.

While the member migrates to a new NVE, the NVE must be able to get the ACL as soon as possible.

3.3. QoS

Similar to ACL, NVE needs to get the QoS policies while a new member is associated with the NVE. In order to achieve QoS policies, not only the NVE but also the network devices on traffic path other than NVE need to be aware of the QoS policies. But in the NVO3 working group, we only focus on NVE.

While the member migrates to a new NVE, the NVE must be able to get the QoS policies as soon as possible.

3.4. DHCP Snooping

While DHCP Snooping function is enabled on NVE, a DHCP snooping table item is created by the access NVE. While VM migrates to a new NVE, the VM may not resend a DHCP request since the migration is transparent to the VM and the IP address must be the same. In this case, the new NVE must be able to get the DHCP Snooping information created by the original NVE by some way. And the original NVE must be able to delete the DHCP Snooping information timely.

3.5. NVE to VNI Registration

While the first membership to a specific VNI is created on NVE, NVE need to register the association to an external entity. The reason

for this is to enable an a global view of which NVEs belongs to a

specific VNI. Every NVE must be aware of NVE to VNI mapping for multicast in a single VNI or to update the QoS/ACL policies. For example, all NVEs responsible to at least one member belong to a particular VNI have to be notified of updated ACL or QoS policies related to this VNI.

3.6. VNI to Multicast Addr Mapping

NVE can get the inner to outer address mapping through control plane assistance or through data plane learning. In the case of latter, NVE must be able to learn the VNI to Multicast address mapping in order to forward unknown unicast and broadcast traffic.

3.7. Synchronization

This assistance a general requirement. For whatever information NVE get from external entity, while the origin of the information is changing, all relevant NVE who have local copy of the information must be able to synchronize with the origin. Some examples of the information are ACL, QoS, Inner to Outer address mapping, VN Name to VNID mapping, and NVEs to VNI global view.

4. Implementation Options and Architecture considerations

The combination of requirements in Section 3 and Section 4 are the assistance that NVE need in order to fulfill the overlay forwarding in a way satisfying the characteristic in Section 3. Not all of the assistance is necessarily regarded as requirements to an external controller. In fact, there are more than one way to enable these requirements. In this section, we introduce 2 kinds of assistance subject to enable the above requirements. These should not be regarded as solution proposals, but considerations on overlay control plan components.

In this draft, we only consider the situation where external NVE is embedded on network devices and VMs access to NVE via hypervisor. But for other cases, the mechanism introduced here can also be used, with necessary prune.

Two assistance subjects are introduced, including external controller and centralized database. It's not feasible to use only database, e.g. it's hard for database to synchronize mapping and QoS/ACL polices among all VNI-relevant NVEs. But a centralized database can offload much work from controller.

4.1. Exclusively using External Controller

Only an external controller is used to assist NVE for virtualization network forwarding. The controller might have a database on it or directly attached.

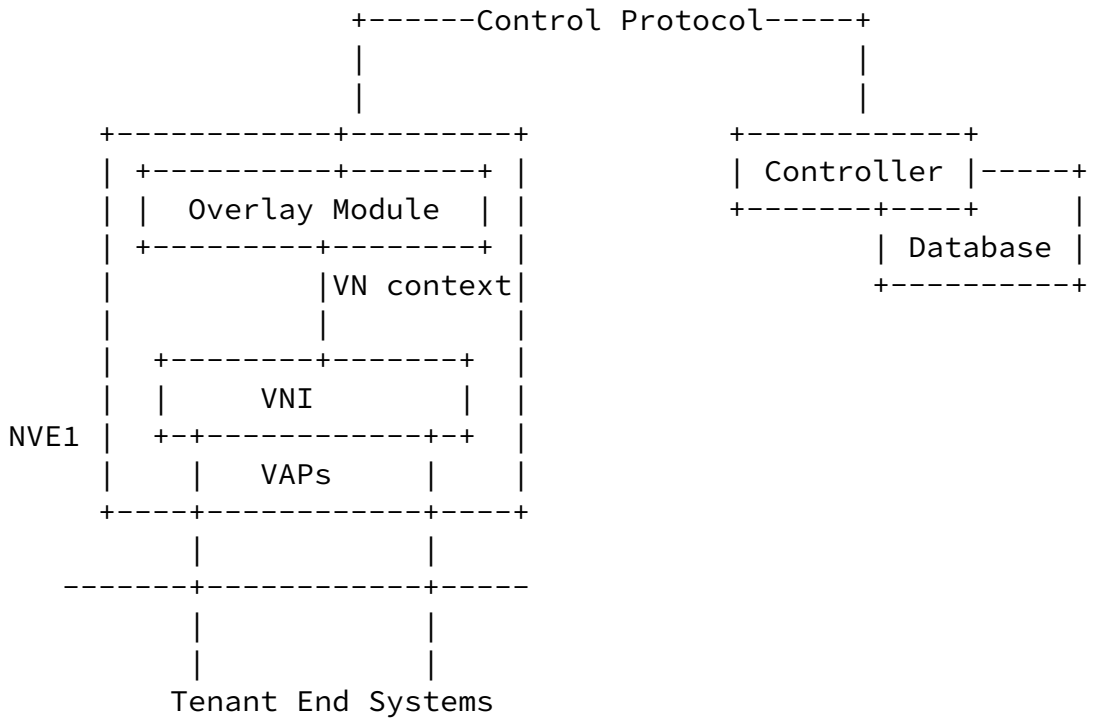


Fig1. Architecture with only controller

The working flow is as follows.

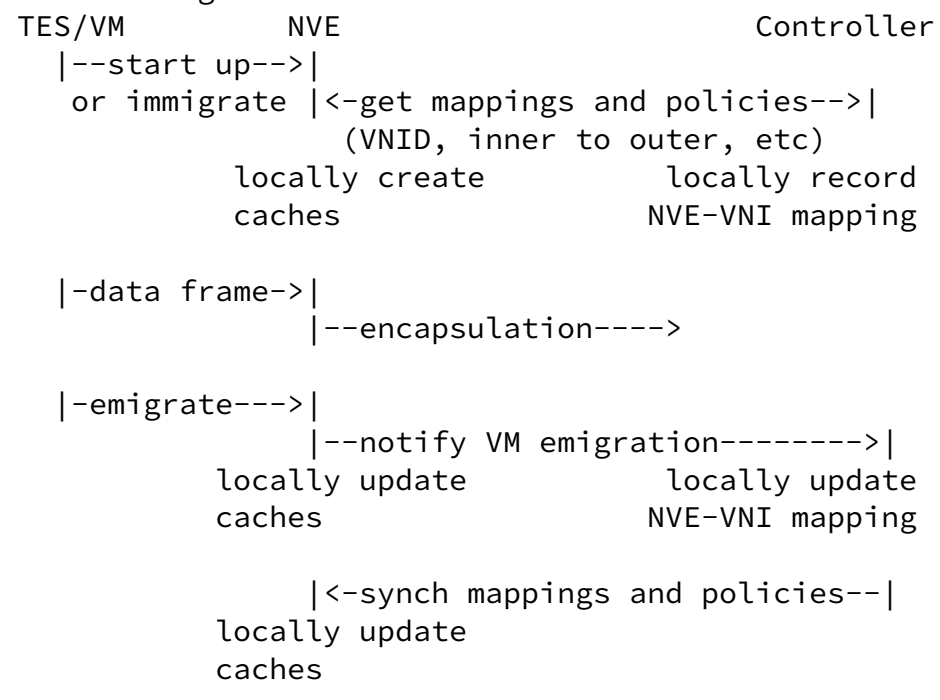


Fig2. Work flow with controller assists NVE

4.2. Hybrid of External Controller and Centralized Database

4.2.1. Brief introduction of VDP profile database and work flow

Take Profile Database introduced in IEEE 802.1Qbg as an example of the Centralized Database. In IEEE 802.1Qbg, a database is mentioned on how to assist the VDP protocol. It's not standardized in IEEE 802.1Qbg, but is a fundamental knowledge while VDP is defined. Please refer to to find out the brief protocol introduction of VDP. The following figure shows what is profile database and how it works. [tes-nve-mechanism]

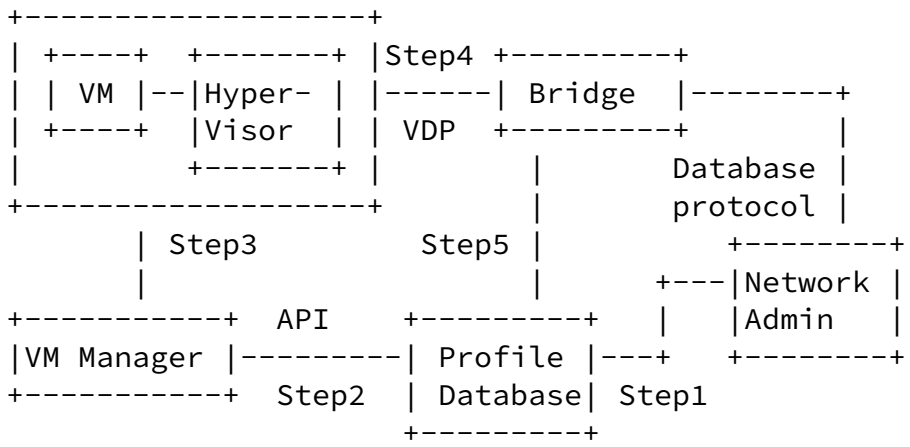


Fig3. VDP Profile Database

A profile database is a centralized database, which is used to store profile of VSI type and VM. A VSI type is a set of policies or resource definition that can be shared by all VMs that choose to use this VSI type. VSI type can be regarded as an instance of Virtual Network. The profile is quite flexible, and it can be organized in a way shown in the following figure and include one or more of the following information. There can be other kind of profile organization format. The profile is very easy to extend to include more information.

VSI type	Profile type	description
VN1	Priority	The priority of traffic
	QoS	QoS policies for the VSI type
	ACL	ACL rules for the VSI type
	Bandwidth	Bandwidth of the traffic
	Multicast Addr	The multicast addr for all VMs belong to the VN
	VNID	A global unique ID for this VN
VN2	Priority	The priority of traffic
	QoS	QoS policies for the VSI type
	ACL	ACL rules for the VSI type
	Bandwidth	Bandwidth of the traffic
	Multicast Addr	The multicast addr for all VMs belong to the VSI type
	VNID	A global unique ID for this VN

Fig4. Profile organization example

A mapping between VSI type and VM is also managed on the database.

VSI type	VM list	Profile type	description
VN1	VM1	MAC Addr	The MAC Addr of VM's vNIC.
		VID	The VID to which the VM is associated.
		Inner Addr	The inner addr of the VM,

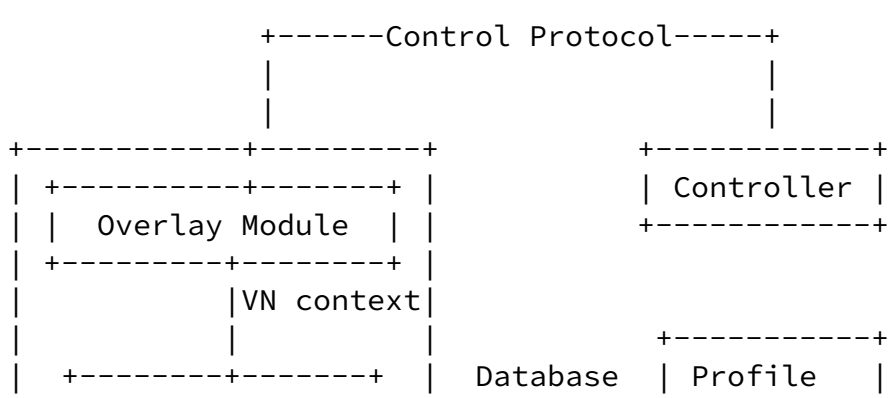
```
|        |       |             | which can be IPv4/v6 addr.|
|        |       |Outer Addr   | The outer addr of the VM, |
|        |       |             | which can be IPv4/v6 addr.|
|        |-------+-------------+---------------------------+
|        |VM2    |MAC Addr     | The MAC Addr of VM's vNIC.|
|        |       |VID          | The VID to which the VM is|
|        |       |             | associated.               |
|        |       |Inner Addr   | The inner addr of the VM, |
|        |       |             | which can be IPv4/v6 addr.|
|        |       |Outer Addr   | The outer addr of the VM, |
|        |       |             | which can be IPv4/v6 addr.|
+--------+-------+-------------+---------------------------+
```

Fig5. VSI type to VM mapping

The work flow of VDP with profile database is as follows.

- Step1: Network Administrator creates VSI type database.
- Step2: VM Manager query available VSI type and obtain a VSI type instance.
- Step3: VM Manager creat a VM on physical server and push VSI type information to Hypervisor
- Step4: While VM is in start up/shut down/emigrate/immigrate status, VDP messages are exchanged between hypervisor and bridge.
- Step5: Bridge retrieve VSI type information from profile database.

4.2.2. Example Architecture and Work Flow



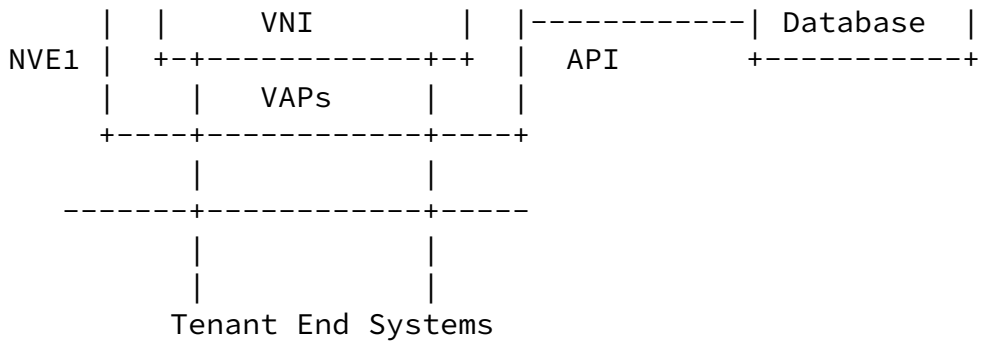
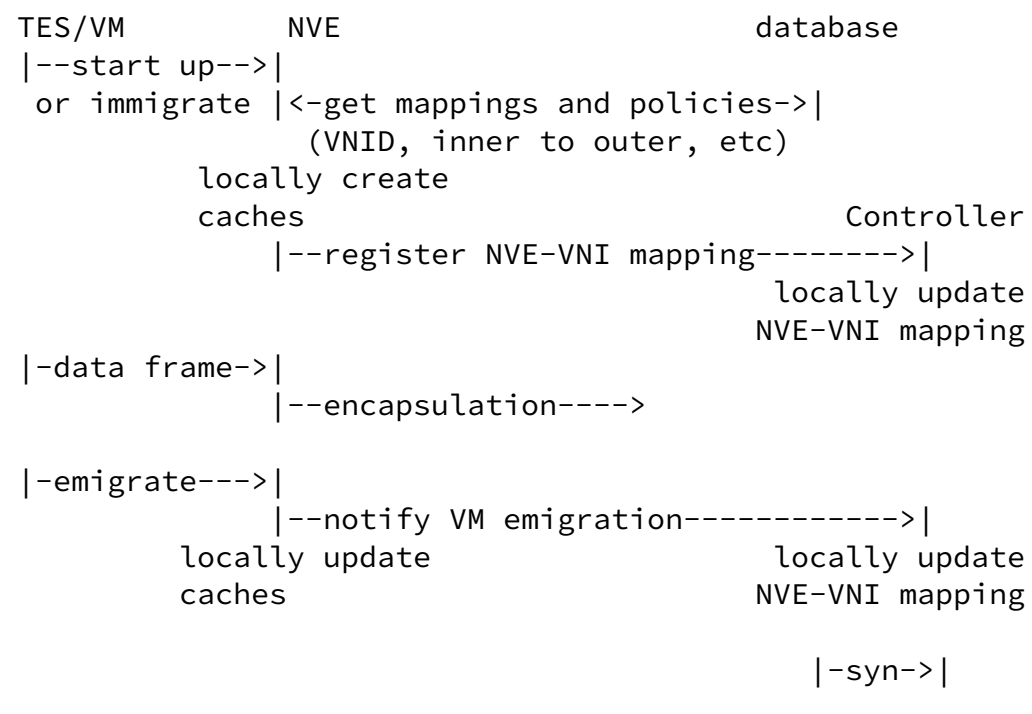


Fig6. Example architecture



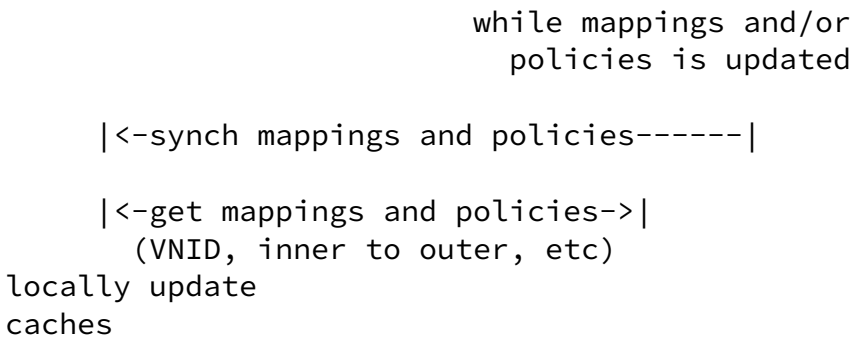


Fig7. Example work flow

5. Summary

Compared the mechanism in Sec 4.1 and 4.2, we can get the following results. From architecture view, exclusive controller has simpler architecture with few interaction requirements, and simpler work flow.

From performance view and reusing of existed protocols, hybird mechanism is able to offload the query of static information to database, which can optimize the performance of controller and make the system more extensible.

6. Security Considerations

TBA

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.

[tes-nve-mechanism]
Gu, Y., "The mechanism and protocol between TES and NVE to facilitate NVO3", July 2012.

Authors' Addresses

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

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

Weiguo Hao
Huawei