Abstract

The document "Problem Statement: Overlays for Network Virtualization" discusses the needs for network virtualization using overlay networks in highly virtualized data centers. The problem statement outlines a need for control protocols to facilitate running these overlay networks. This document outlines the high level requirements related to the interaction between hypervisors and the Network Virtualization Edge device when the two entities are not co-located on the same physical device.

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 April 19, 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
Table of Contents

1. Introduction ......................................................... 3
2. Terminology .......................................................... 3
3. Hypervisor-NVE Control Plane Protocol Functionality ................ 4
   3.1. VN Connect/Disconnect Notification .............................. 6
   3.2. VN Profile .......................................................... 7
4. Security Considerations .............................................. 7
5. Informative References .............................................. 7
Authors’ Addresses ....................................................... 8
1. Introduction

Note: the contents of this document were originally in [I-D.kreeger-nvo3-overlay-cp]. The content has been pulled into its own document because the problem area covered is distinct and different from what most folk think of as a "control protocol" for NVO3. Other related documents on this same general topic include [I-D.kompella-nvo3-server2nve], [I-D.gu-nvo3-overlay-cp-arch], and [I-D.gu-nvo3-tes-nve-mechanism].

"Problem Statement: Overlays for Network Virtualization" [I-D.ietf-nvo3-overlay-problem-statement] discusses the needs for network virtualization using overlay networks in highly virtualized data centers and provides a general motivation for building such networks. "Framework for DC Network Virtualization" [I-D.ietf-nvo3-framework] provides a framework for discussing overlay networks generally and the various components that must work together in building such systems. The reader is assumed to be familiar with both documents.

Section 3.5 of [I-D.ietf-nvo3-overlay-problem-statement] describes three separate work areas that fall under the general category of a control protocol for NVO3. This document focuses entirely on the control protocol related to the hypervisor-NVE interaction, labeled as the "third work item" in [I-D.ietf-nvo3-overlay-problem-statement]. Requirements for the other control protocol components and work areas are described in [I-D.kreeger-nvo3-overlay-cp].

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

2. Terminology

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

Network Service Appliance: A stand-alone physical device or a virtual device that provides a network service, such as a firewall, load balancer, etc. Such appliances may embed Network Virtualization Edge (NVE) functionality within them in order to more efficiently operate as part of a virtualized network.
VDC: Virtual Data Center. A container for virtualized compute, storage and network services. Managed by a single tenant, a VDC can contain multiple VNs and multiple Tenant End Systems that are connected to one or more of these VNs.

VN Alias: A string name for a VN as used by administrators and customers to name a specific VN. A VN Alias is a human-usable string that can be listed in contracts, customer forms, email, configuration files, etc. and that can be communicated easily vocally (e.g., over the phone). A VN Name is independent of the underlying technology used to implement a VN and will generally not be carried in protocol fields of control protocols used in virtual networks. Rather, a VN Alias will be mapped into a VN Name where precision is required.

VN Name: A globally unique identifier for a VN suitable for use within network protocols. A VN Name will usually be paired with a VN Alias, with the VN Alias used by humans as a shorthand way to name and identify a specific VN. A VN Name should have a compact representation to minimize protocol overhead where a VN Name is carried in a protocol field. Using a Universally Unique Identifier (UUID) as discussed in RFC 4122, may work well because it is both compact and a fixed size and can be generated locally with a very high likelihood of global uniqueness.

VN ID: A unique and compact identifier for a VN within the scope of a specific NVO3 administrative domain. It will generally be more efficient to carry VN IDs as fields in control protocols than VN Aliases. There is a one-to-one mapping between a VN Name and a VN ID within an NVO3 Administrative Domain. Depending on the technology used to implement an overlay network, the VN ID could be used as the Context Identifier in the data plane, or would need to be mapped to a locally-significant Context Identifier.

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

3. Hypervisor-NVE Control Plane Protocol Functionality

The problem statement [I-D.ietf-nvo3-overlay-problem-statement], discusses the needs for a control plane protocol (or protocols) to populate each NVE with the state needed to perform its functions.
In one common scenario, an NVE provides overlay encapsulation/decapsulation packet forwarding services to Tenant End Systems (TESs) that are co-resident with the NVE on the same End System (e.g. when the NVE is embedded within a hypervisor or a Network Service Appliance). In such cases, there is no need for a standardized protocol between the hypervisor and NVE, as the interaction is implemented via software on a single device. Alternatively, a TES may use an externally connected NVE (e.g. an NVE residing on a physical Network Switch connected to the hypervisor via an access network). The following figures give example scenarios where the NVE and hypervisor are on different devices separated by an access network.

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

Hypervisor with an External NVE.

Figure 1

Network Service Appliance | Access Switch
-------------------------|----------------
| +--------+ | \|
| Net Service | Instance | NVE ++ Underlying
| +--------+ | \ | Trunk | Network

Physical Network Service Appliance with an External NVE.

Figure 2

In the examples above, the physical VLAN Trunk connecting the Hypervisor or Network Services Appliance to the external NVE only
needs to carry locally significant (e.g. link-local) VLAN tag values. These tags are only used to differentiate two different VNs as packets cross the (shared) access network to the external NVE. When the NVE receives packets, it uses the VLAN tag to identify the VN the TES belongs to, strips the tag, and adds the appropriate overlay encapsulation for that VN.

On the hypervisor-facing side of the NVE, a control plane protocol is necessary to provide an NVE with the information it needs to connect a given TES to its associated Virtual Network. Specifically, the hypervisor (or Network Service Appliance) utilizing an external NVE needs to "attach to" and "detach from" an NVE. Thus, they will need a protocol that runs across the access network between the two devices that identifies the TES and VN Name for which the NVE is providing service. In addition, such a protocol will identify a locally significant tag (e.g., an 802.1Q VLAN tag) that can be used to identify the data frames that flow between the TES and VN.

3.1. VN Connect/Disconnect Notification

In the previous figures, NVEs reside on an external networking device (e.g. an access switch). Using an external network device as the NVE can provide an offload of the encapsulation / decapsulation function and the protocol overheads which may provide performance improvements and/or resource savings to the client End Device making use of the external NVE.

When an NVE is external, a protocol is needed between a client End Device making use of the external NVE and the NVE itself in order to make the NVE aware of the changing VN membership requirements of the client End Device. A key driver for using a protocol rather than using static configuration of the external NVE is because the VN connectivity requirements can change frequently as VMs are brought up, moved and brought down on various hypervisors throughout the data center.

The NVE must be notified when an End Device requires connection to a particular VN and when it no longer requires connection. This protocol should also provide the inner TES addresses within the VN that the End Device contains (e.g. the virtual MAC address of a VMs virtual NIC) to the external NVE. In addition, the external NVE must provide a local tag value for each connected VN to the End Device to use for exchange of packets between the End Device to the NVE (e.g. a locally significant 802.1Q tag value).

The Identification of the VN in this protocol could either be through a VN Name or a VN ID. A globally unique VN Name facilitates portability of a Tenant’s Virtual Data Center. When a VN within a
VDC is instantiated within a particular administrative domain, it can be allocated a VN Context which only the NVE needs to use. An End Device that is making use of an offloaded NVE only needs to communicate the VN Name to the NVE, and get back a locally significant tag value.

3.2. VN Profile

Once an NVE (embedded or external) receives a VN connect indication with a specified VN Name or ID, the NVE must determine the VN Context value to encapsulate packets with as well as other information that may be needed (e.g., QoS settings). The NVE serving that hypervisor needs a way to get a VN Context allocated or receive the already allocated VN Context for a given VN Name or ID (as well as any other information needed to transmit encapsulated packets). A protocol for an NVE to get this mapping may be a useful function, but would be the subject of work items 1 and 2 in [I-D.ietf-nvo3-overlay-problem-statement].

4. Security Considerations

Editor’s Note: This is an initial start on the security considerations section; it will need to be expanded, and suggestions for material to add are welcome.

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

5. Informative References

[I-D.gu-nvo3-overlay-cp-arch]
Yingjie, G. and W. Hao, "Analysis of external assistance to NVE and consideration of architecture",
draft-gu-nvo3-overlay-cp-arch-00 (work in progress), July 2012.
[I-D.gu-nvo3-tes-nve-mechanism] Yingjie, G., "The mechanism and protocol between VAP and TES to facilitate NVO3", draft-gu-nvo3-tes-nve-mechanism-00 (work in progress), July 2012.


Authors’ Addresses

Lawrence Kreeger
Cisco

Email: kreeger@cisco.com

Thomas Narten
IBM

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

Email: david.black@emc.com