Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: August 2, 2012

L. Kreeger
D. Dutt
Cisco
T. Narten
IBM
D. Black
EMC
M. Sridharan
Microsoft

January 30, 2012

Network Virtualization Overlay Control Protocol Requirements
draft-kreeger-nvo3-overlay-cp-00

Abstract

The document draft-narten-nvo3-overlay-problem-statement-01 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 to be fulfilled by the control protocols.

Table of Contents

1. Introduction

The document draft-narten-nvo3-overlay-problem-statement-01 discusses the needs for network virtualization using overlay networks in highly virtualized data centers. It focuses the problem less on the particular encapsulation, or even what address families are carried inside/outside the overlay, but instead on the control protocol issues that need to be addressed in order to provide a solution. The problem statement discusses the use of virtual network overlays where the encapsulation/decapsulation is performed by the first hop switch in the data center, which could be either a virtual switch residing in the hypervisor, or a physical access switch connected to a server or Network Service Appliance.

2. Terminology

This document uses the following terminology:

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

VN-ID: Virtual Network Identifier. 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. It should be a large enough ID space to not be a limiting factor within an administrative domain managing the ID space. There are several technologies which encapsulate using a 24 bit ID value, e.g. PBB, SPBM, LISP, OTV, TRILL Fine-grained labels, VXLAN, NVGRE.

OBP: Overlay Boundary Point. This is a network entity that is on the edge boundary of the overlay. It performs encapsulation to send packets to other OBPs across an Underlying Network for decapsulation. An OBP 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.

Underlying Network: This is the network that provides the connectivity between the OBPs. The Underlying Network can be completely unaware of the VN of packets carried within the encapsulation. 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: 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.

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 OBP functionality within them in order to more efficiently operate as part of a virtualized network.

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.

End Station: This is an end device which connects to a VN. The End Station is unaware of how the VN is implemented. OBPs encapsulate/decapsulate on the behalf of these End Stations. An End Station can be a VM, a physical server, or a Network Service Appliance. End Station addresses are also referred to as "inner addresses" because they exist inside of the overlay encapsulation payload.

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.

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 End Stations that are connected to one or more of these VNs.

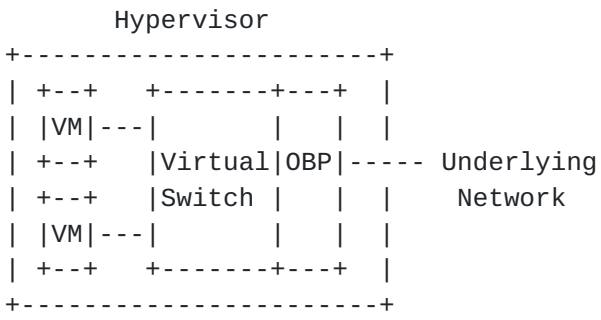
VN Name: A globally unique name for a VN. The VN Name is not carried in data packets originating from End Stations, but must be mapped into an appropriate VN-ID for a particular encapsulating technology. Using VN Names rather than VN-IDs to identify VNs in configuration files and control protocols increases the portability of a VDC and its associated VNs when moving among different administrative domains (e.g. switching to a different cloud service provider).

3. Control Plane Protocol Functionality

The problem statement (draft-narten-nvo3-overlay-problem-statement-01), discusses the needs for a control plane protocol (or protocols) to populate each OBP with the state needed to peform its functions.

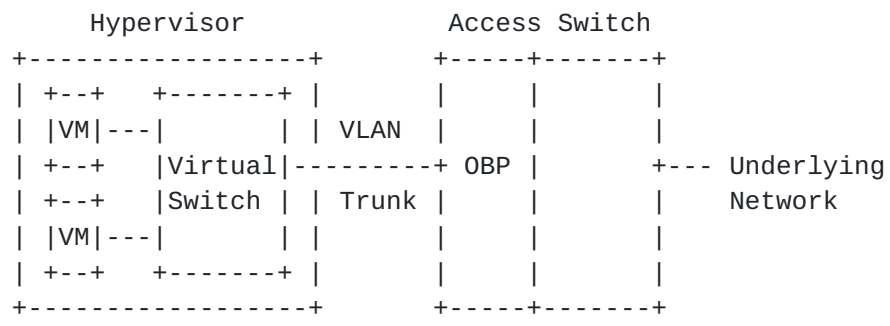
Note that an OBP may provide overlay encapsulation/decapsulation packet forwarding services to End Stations that are co-resident within the same device (e.g. when the OBP is embedded within a hypervisor or a Network Service Appliance), or to End Stations that are externally connected to the OBP (e.g. a physical Network Service Appliance connected to an access switch containing the OBP).

The following figures show examples of scenarios in which the OBP is co-resident within the same device as the End Stations connected to a given VN, and when the OBP is externally located within the access switch.



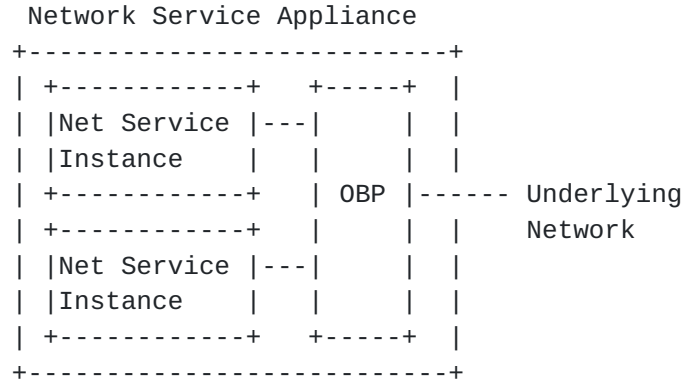
Hypervisor with an Embedded OBP

Figure 1



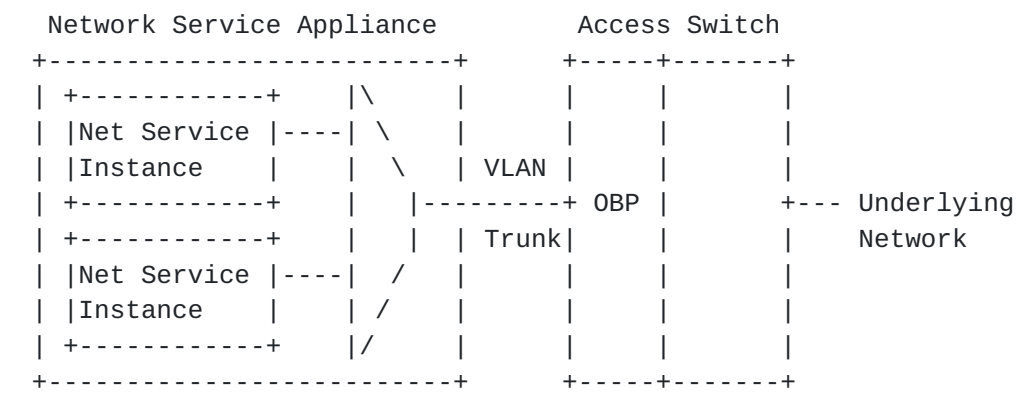
Hypervisor with an External OBP

Figure 2



Network Service Appliance (physical or virtual) with an Embedded OBP

Figure 3



Physical Network Service Appliance with an External OBP

Figure 4

In the examples above where the OBP functionality is located in the physical access switch, the physical VLAN Trunk connecting the Hypervisor or Network Services Appliance to the external OBP 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 wire to the external OBP. When the OBP receives packets, it uses the VLAN tag to identify the VN the End Station belongs to, strips the tag, and adds the appropriate overlay encapsulation for that VN.

Given the above, a control plane protocol is necessary to provide an OBP with the information it needs to maintain its own internal state necessary to carry out its forwarding functions as explained in detail below.

1. An OBP maintains a per-VN table of mappings from End Station (inner) addresses to Underlying Network (outer) addresses of remote OBPs.

2. An OBP maintains per-VN state for delivering multicast and broadcast packets to other End Stations. Such state could include a list of multicast addresses and/or unicast addresses on the Underlying Network for the OBPs associated with a particular VN.

3. Devices (such as a Hypervisor or Network Service Appliance) utilizing an external OBP need to "attach to" and "detach from" an OBP. Specifically, they will need a protocol that runs across the L2 link between the two devices that identifies the End Station and VN Name for which the OBP 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 End Station and VN.

4. An OBP needs a mapping from each unique VN name to the VN-ID value used within encapsulated data packets within the administrative domain that the VN is instantiated.

3.1. Inner to Outer Address Mapping

When presented with a data packet to forward to an End Station within a VN, the OBP needs to know the mapping of the End Station destination (inner) address to the (outer) address on the Underlying Network of the remote OBP which can deliver the packet to the destination End Station.

A protocol is needed to provide this inner to outer mapping to each OBP that requires it and keep the mapping updated in a timely manner. Timely updates are important for maintaining connectivity between End Stations when one End Station is a VM

Note that one technique that could be used to create this mapping without the need for a control protocol is via data plane learning; However, the learning approach requires packets to be flooded to all OBPs participating in the VN when no mapping exists. One goal of using a control protocol is to eliminate this flooding.

3.2. Underlying Network Multi-Destination Delivery Address(es)

Each OBP needs a way to deliver multi-destination packets (i.e. broadcast/multicast) within a given VN to each remote OBP which has a destination End Station for these packets. Three possible ways of accomplishing this:

- Use the multicast capabilities of the Underlying Network.
- Have each OBP replicate the packets and send a copy across the Underlying Network to each remote OBP currently participating in the VN.
- Use one or more distribution servers which replicates the packets on the behalf of the OBPs.

Whichever method is used, a protocol is needed to provide on a per VN basis, one or more multicast address (assuming the Underlying Network supports multicast), and/or one or more unicast addresses of either the remote OBPs which are not multicast reachable, or of one or more distribution servers for the VN.

The protocol must also keep the list of addresses up to date in a timely manner if the set of OBPs for a given VN changes over time. For example, the set of OBPs for a VN could change as VMs power on/ off or migrate to different hypervisors.

3.3. VN Connect/Disconnect Notification

As the previous figures illustrated, OBPs may be embedded within a device (such as a Hypervisor or Network Service Appliance), or within an external networking device (e.g. an access switch). Using an external network device as the OBP 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 device making use of the external OBP.

When an OBP is external, a protocol is needed between a client device making use of the external OBP and the OBP itself in order to make the OBP aware of the changing VN membership requirements of the client device. A key driver for using a protocol rather than using static configuration of the exernal OBP 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 OBP must be notified when a client device requires connection to a particular VN and when it no longer requires connection. This protocol should also provide the inner End Station addresses within the VN that the client device contains (e.g. the virtual MAC address of a VMs virtual NIC) to the external OBP. In addition, the external OBP must provide a local tag value for each connected VN to the client device to use for exchange of packets between the client device to the OBP (e.g. a locally significant 802.1Q tag value).

The Identification of the VN in this protocol should preferably be made using a globally unique VN Name. 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-ID which only the OBP needs to use. A client device that is making use of an offloaded OBP only needs to communicate the VN Name to the OBP, and get back a locally significant tag value. Ideally the VN Name should be compact as well unique to minimize protocol overhead. Using a Universally Unique Identifier (UUID) as discussed in RFC 4122, would work well because it is both compact and a fixed size and can be generated locally with a high likelihood of being unique.

3.4. VN Name to VN-ID Mapping

Once an OBP (embedded or external) receives a VN connect indication with a specified VN name, the OBP must find the VN-ID value to encapsulate packets with. The OBP serving that hypervisor needs a way to get a VN-ID allocated or receive the already allocated VN-ID for a given VN Name. A protocol for an OBP to get this mapping may be a useful function.

4. Control Plane Characteristics

OBPs are expected to be implemented within hypervisors or access switches, or even within a Network Service Appliance. Any resources used by these protocols (e.g. processing or memory) takes away resources that could be better used by these devices to perform their intended functions (e.g. providing resoures for hosted VMs).

A large scale data center may contain hundreds of thousands of these OBPs (which may be several independent implementations); Therefore, any savings in per-OBP resources can be multiplied hundreds of thousands of times.

Given this, the control plane protocol(s) implemented by OBPs to provide the functionality discussed above should have the below characteristics.

1. Minimize the amount of state needed to be stored on each OBP. The OBP should only be required to cache state that it is actively using, and be able to discard any cached state when it is no longer required. For example, an OBP should only need to maintain an inner-to-outer address mapping for destinations to which it is actively sending traffic as opposed to maintaining mappings for all possible destinations.

2. Fast acquisition of needed state. For example, when an End Station emits a packet destined to an inner address that the OBP does not have a mapping for, the OBP should be able to acquire the needed mapping quickly.

3. Fast detection/update of stale cached state information. This only applies if the cached state is actually being used. For example, when a VM moves such that it is connected to a different OBP, the inner to outer mapping for this VM's address that is cached on other OBPs must be updated in a timely manner (if they are actively in use). If the update is not timely, the OBPs will forward data to the wrong OBP until it is updated.

4. Minimize processing overhead. This means that an OBP should only be required to perform protocol processing directly related to maintaining state for the End Stations it is actively communicating with. This requirement is for the OBP functionality only. The network node that contains the OBP may be involved in other functionality for the underlying network that maintains connectivity that the OBP is not actively using (e.g., routing and multicast distribution protocols for the underlying network).

5. Highly scalable. This means scaling to hundreds of thousands of OBPs and several million VNs within a single administrative domain. As the number of OBPs and/or VNs within a data center grows, the protocol overhead at any one OBP should not increase significantly.

6. Minimize the complexity of the implementation. This argues for using the least number of protocols to achieve all the functionality listed above. Ideally a single protocol should be able to be used. The less complex the protocol is on the OBP, the more likely interoperable implementations will be created in a timely manner.

7. Extensible. The protocol should easily accommodate extension to meet related future requirements. For example, access control or QoS policies, or new address families for either inner or outer addresses should be easy to add while maintaining interoperability with OBPs running older versions.

8. Simple protocol configuration. A minimal amount of configuration should be required for a new OBP to be provisioned. Existing OBPs should not require any configuration changes when a new OBP is provisioned. Ideally OBPs should be able to auto configure themselves.

9. Do not rely on IP Multicast in the Underlying Network. Many data centers do not have IP multicast routing enabled. If the Underlying Network is an IP network, the protocol should allow for, but not require the presence of IP multicast services within the data center.

10. Flexible mapping sources. Inner to outer address mappings should be able to be created by either the OBPs themselves or other third party entities (e.g. data center management or orchestration systems). The protocol should allow for mappings created by an OBP to be automatically removed from all other OBPs if it fails or is brought down unexpectedly.

11. Secure. See the Security Considerations section below.

5. 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.

The protocol(s) should protect the integrity of the mapping against both off-path and on-path attacks. It should authenticate the systems that are creating mappings, and rely on light weight security mechanisms to minimize the impact on scalability and allow for simple configuration.

Use of an overlay exposes virtual networks to attacks on the underlying network beyond attacks on the control protocol that is the subject of this draft. In addition to the directly applicable security considerations for the networks involved, the use of an overlay enables attacks on encapsulated virtual networks via the underlying network. Examples of such attacks include traffic injection into a virtual network via injection of encapsulated traffic into the underlying network and modifying underlying network traffic to forward traffic among virtual networks that should have no connectivity. The control protocol should provide functionality to help counter some of these attacks, e.g., distribution of OBP access control lists for each virtual network to enable packets from non-participating OBPs to be discarded, but the primary security measures for the underlying network need to be applied to the underlying network. For example, if the underlying network includes connectivity across the public Internet, use of secure gateways (e.g., based on IPsec [RFC 4301]) may be appropriate.

The inner to outer address mappings used for forwarding data towards a remote OBP could also be used to filter incoming traffic to ensure the inner address sourced packet came from the correct OBP source address, allowing access control to discard traffic that does not originate from the correct OBP. This destination filtering functionality should be optional to use.

6. Acknowledgements

Thanks to the following people for reviewing and providing feedback: Fabio Maino, Victor Moreno, Ajit Sanzgiri, Chris Wright.

Authors' Addresses

Lawrence Kreeger
Cisco

Email: kreeger@cisco.com

Dinesh Dutt
Cisco

Email: ddutt@cisco.com

Thomas Narten
IBM

Email: narten@us.ibm.com

David Black
EMC

Email: david.black@emc.com

Murari Sridharan
Microsoft

Email: muraris@microsoft.com