Internet Engineering Task Force
Internet Draft
Intended status: Informational
Expires: May 2014

Marc Lasserre
Florin Balus
Alcatel-Lucent

Thomas Morin
France Telecom Orange

Nabil Bitar
Verizon

Yakov Rekhter
Juniper

November 12, 2013

Framework for DC Network Virtualization
draft-ietf-nvo3-framework-04.txt

Abstract

This document provides a framework for Network Virtualization over L3 (NVO3) and it defines a reference model along with logical components required to design a solution.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

This document provides a framework for Data Center Network Virtualization over Layer3 (L3) tunnels. This framework is intended to aid in standardizing protocols and mechanisms to support large-scale network virtualization for data centers.

[NVOPS] defines the rationale for using overlay networks in order to build large multi-tenant data center networks. Compute, storage and network virtualization are often used in these large data centers to support a large number of communication domains and end systems.

This document provides reference models and functional components of data center overlay networks as well as a discussion of technical issues that have to be addressed.

1.1. General terminology

This document uses the following terminology:

NVO3 Network: An overlay network that provides an Layer2 (L2) or Layer3 (L3) service to Tenant Systems over an L3 underlay network using the architecture and protocols as defined by the NVO3 Working Group.

Network Virtualization Edge (NVE). An NVE is the network entity that sits at the edge of an underlay network and implements L2 and/or L3 network virtualization functions. The network-facing side of the NVE uses the underlying L3 network to tunnel tenant frames to and from other NVEs. The tenant-facing side of the NVE sends and receives Ethernet frames to and from individual Tenant Systems. An NVE could be implemented as part of a virtual switch within a hypervisor, a

physical switch or router, a Network Service Appliance, or be split across multiple devices.

Virtual Network (VN): A VN is a logical abstraction of a physical network that provides L2 or L3 network services to a set of Tenant Systems. A VN is also known as a Closed User Group (CUG).

Virtual Network Instance (VNI): A specific instance of a VN from the perspective of an NVE.

Virtual Network Context (VN Context) Identifier: Field in overlay encapsulation header that identifies the specific VN the packet belongs to. The egress NVE uses the VN Context identifier to deliver the packet to the correct Tenant System. The VN Context identifier can be a locally significant identifier or a globally unique identifier.

Underlay or Underlying Network: The network that provides the connectivity among NVEs and over which NVO3 packets are tunneled, where an NVO3 packet carries an NVO3 overlay header followed by a tenant packet. The Underlay Network does not need to be aware that it is carrying NVO3 packets. Addresses on the Underlay Network appear as "outer addresses" in encapsulated NVO3 packets. In general, the Underlay Network can use a completely different protocol (and address family) from that of the overlay. In the case of NVO3, the underlay network is IP.

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, compute and/or storage services. One such service is virtualized infrastructure data center services, also known as Infrastructure as a Service.

Virtual Data Center (Virtual DC): A container for virtualized compute, storage and network services. A Virtual DC is associated with a single tenant, and can contain multiple VNs and Tenant Systems connected to one or more of these VNs.

Virtual machine (VM): A software implementation of a physical machine that runs programs as if they were executing on a physical, non-virtualized machine. Applications (generally) do not know they are running on a VM as opposed to running on a "bare metal" host or server, though some systems provide a para-virtualization

environment that allows an operating system or application to be aware of the presences of virtualization for optimization purposes.

Hypervisor: Software running on a server that allows multiple VMs to run on the same physical server. The hypervisor manages and provides shared compute/memory/storage and network connectivity to the VMs that it hosts. Hypervisors often embed a Virtual Switch (see below).

Server: A physical end host machine that runs user applications. A standalone (or "bare metal") server runs a conventional operating system hosting a single-tenant application. A virtualized server runs a hypervisor supporting one or more VMs.

Virtual Switch (vSwitch): A function within a Hypervisor (typically implemented in software) that provides similar forwarding services to a physical Ethernet switch. A vSwitch forwards Ethernet frames between VMs running on the same server, or between a VM and a physical NIC card connecting the server to a physical Ethernet switch or router. A vSwitch also enforces network isolation between VMs that by policy are not permitted to communicate with each other (e.g., by honoring VLANs). A vSwitch may be bypassed when an NVE is enabled on the host server.

Tenant: The customer using a virtual network and any associated resources (e.g., compute, storage and network). A tenant could be an enterprise, or a department/organization within an enterprise.

Tenant System: A physical or virtual system that can play the role of a host, or a forwarding element such as a router, switch, firewall, etc. It belongs to a single tenant and connects to one or more VNs of that tenant.

Tenant Separation: Tenant Separation refers to isolating traffic of different tenants such that traffic from one tenant is not visible to or delivered to another tenant, except when allowed by policy. Tenant Separation also refers to address space separation, whereby different tenants can use the same address space without conflict.

Virtual Access Points (VAPs): A logical connection point on the NVE for connecting a Tenant System to a virtual network. Tenant Systems connect to VNIs at an NVE through VAPs. VAPs can be physical ports or virtual ports identified through logical interface identifiers (e.g., VLAN ID, internal vSwitch Interface ID connected to a VM).

End Device: A physical device that connects directly to the DC Underlay Network. This is in contrast to a Tenant System, which connects to a corresponding tenant VN. An End Device is administered by the DC operator rather than a tenant, and is part of the DC infrastructure. An End Device may implement NVO3 technology in support of NVO3 functions. Examples of an End Device include hosts (e.g., server or server blade), storage systems (e.g., file servers, iSCSI storage systems), and network devices (e.g., firewall, load-balancer, IPSec gateway).

Network Virtualization Authority (NVA): Entity that provides reachability and forwarding information to NVEs.

1.2. DC network architecture

A generic architecture for Data Centers is depicted in Figure 1:

```
                                ,---------.
                              ,'           `.
                             (  IP/MPLS WAN )
                              `.           ,'
                                `-+------+'
                                 \      /
                          +--------+   +--------+
                          |   DC   |+-+|   DC   |
                          |gateway |+-+|gateway |
                          +--------+   +--------+
                                |       /
                                .--. .--.
                              (    '    '.--.
                            .-.' Intra-DC     '
                           (     network      )
                            (             .'-'
                             '--'._.'.    )\ \
                             / /     '--'  \ \
                            / /      | |    \ \
                   +--------+   +--------+   +--------+
                   | access |   | access |   | access |
                   | switch |   | switch |   | switch |
                   +--------+   +--------+   +--------+
                      /     \    /    \     /      \
                   __/_      \  /      \   /_      _\__
             '--------'   '--------'   '--------'   '--------'
             :  End   :   :  End   :   :  End   :   :  End   :
             : Device :   : Device :   : Device :   : Device :
             '--------'   '--------'   '--------'   '--------'
```

Figure 1 : A Generic Architecture for Data Centers

An example of multi-tier DC network architecture is presented in Figure 1. It provides a view of physical components inside a DC.

A DC network is usually composed of intra-DC networks and network services, and inter-DC network and network connectivity services.

DC networking elements can act as strict L2 switches and/or provide IP routing capabilities, including network service virtualization.

In some DC architectures, some tier layers could provide L2 and/or L3 services. In addition, some tier layers may be collapsed, and Internet connectivity, inter-DC connectivity and VPN support may be handled by a smaller number of nodes. Nevertheless, one can assume

that the network functional blocks in a DC fit in the architecture depicted in Figure 1.

The following components can be present in a DC:

- o Access switch: Hardware-based Ethernet switch aggregating all Ethernet links from the End Devices in a rack representing the entry point in the physical DC network for the hosts. It may also provide routing functionality, virtual IP network connectivity, or Layer2 tunneling over IP for instance. Access switches are usually multi-homed to aggregation switches in the Intra-DC network. A typical example of an access switch is a Top of Rack (ToR) switch. Other deployment scenarios may use an intermediate Blade Switch before the ToR, or an EoR (End of Row) switch, to provide similar function as a ToR.

- o Intra-DC Network: Network composed of high capacity core nodes (Ethernet switches/routers). Core nodes may provide virtual Ethernet bridging and/or IP routing services.

- o DC Gateway (DC GW): Gateway to the outside world providing DC Interconnect and connectivity to Internet and VPN customers. In the current DC network model, this may be simply a router connected to the Internet and/or an IP Virtual Private Network (VPN)/L2VPN PE. Some network implementations may dedicate DC GWs for different connectivity types (e.g., a DC GW for Internet, and another for VPN).

Note that End Devices may be single or multi-homed to access switches.

2. Reference Models

2.1. Generic Reference Model

Figure 2 depicts a DC reference model for network virtualization using L3 (IP/MPLS) overlays where NVEs provide a logical interconnect between Tenant Systems that belong to a specific VN.

```
         +--------+                                    +--------+
         | Tenant +--+                            +----| Tenant |
         | System |  |                           (')   | System |
         +--------+  |    .................     (   )  +--------+
                     |  +---+           +---+    (_)
                     +--|NVE|---+   +---|NVE|-----+
                        +---+   |   |   +---+
                        / .    +-----+      .
                       /  . +--| NVA |      .
                      /   . |  +-----+      .
                     |    . |               .
                     |    . |  L3 Overlay +--+--++--------+
         +--------+  |    . |   Network   | NVE || Tenant |
         | Tenant +--+    . |             |     || System |
         | System |       .  \ +---+      +--+--++--------+
         +--------+       .....|NVE|.........
                               +---+
                                 |
                                 |
                       =====================
                         |               |
                     +--------+      +--------+
                     | Tenant |      | Tenant |
                     | System |      | System |
                     +--------+      +--------+
```

Figure 2 : Generic reference model for DC network virtualization
over a Layer3 (IP) infrastructure

In order to get reachability information, NVEs may exchange information directly between themselves via a protocol. In this case, a control plane module resides in every NVE. This is how routing control plane modules are implemented in routers for instance.

It is also possible for NVEs to communicate with an external Network Virtualization Authority (NVA) to obtain reachability and forwarding information. In this case, a protocol is used between NVEs and NVA(s) to exchange information. OpenFlow [OF] is one example of such a protocol.

It should be noted that NVAs may be organized in clusters for redundancy and scalability and can appear as one logically centralized controller. In this case, inter-NVA communication is necessary to synchronize state among nodes within a cluster or share information across clusters. The information exchanged between NVAs of the same cluster could be different from the information exchanged across clusters.

A Tenant System can be attached to an NVE in several ways:

- locally, by being co-located in the same End Device

- remotely, via a point-to-point connection or a switched network

When an NVE is co-located with a Tenant System, the state of the Tenant System can be provided without protocol assistance. For instance, the operational status of a VM can be communicated via a local API. When an NVE is remotely connected to a Tenant System, the state of the Tenant System or NVE needs to be exchanged directly or via a management entity, using a control plane protocol or API, or directly via a dataplane protocol.

The functional components in Figure 2 do not necessarily map directly to the physical components described in Figure 1. For example, an End Device can be a server blade with VMs and a virtual switch. A VM can be a Tenant System and the NVE functions may be performed by the host server. In this case, the Tenant System and NVE function are co-located. Another example is the case where the End Device is the Tenant System, and the NVE function can be implemented by the connected ToR. In this case, the Tenant System and NVE function are not co-located.

Underlay nodes utilize L3 technologies to interconnect NVE nodes. These nodes perform forwarding based on outer L3 header information, and generally do not maintain per tenant-service state albeit some

applications (e.g., multicast) may require control plane or forwarding plane information that pertain to a tenant, group of tenants, tenant service or a set of services that belong to one or more tenants. Mechanisms to control the amount of state maintained in the underlay may be needed.

2.2. NVE Reference Model

Figure 3 depicts the NVE reference model. One or more VNIs can be instantiated on an NVE. A Tenant System interfaces with a corresponding VNI via a VAP. An overlay module provides tunneling overlay functions (e.g., encapsulation and decapsulation of tenant traffic, tenant identification and mapping, etc.).

```
                   +-------- L3 Network -------+
                   |                           |
                   |        Tunnel Overlay     |
       +------------+---------+       +---------+------------+
       | +----------+-------+ |       | +---------+--------+ |
       | |  Overlay Module  | |       | |  Overlay Module  | |
       | +---------+--------+ |       | +---------+--------+ |
       |           |VN context|       | VN context|          |
       |           |          |       |           |          |
       |  +--------+-------+  |       |  +--------+-------+  |
       |  | |VNI|   .  |VNI|  |       |  | |VNI|   .  |VNI|  |
  NVE1 |  +-+------------+-+  |       |  +-+-----------+--+  | NVE2
       |    |   VAPs     |    |       |    |    VAPs   |     |
       +----+------------+----+       +----+-----------+-----+
            |            |                 |           |
            |            |                 |           |
           Tenant Systems                 Tenant Systems
```

Figure 3 : Generic NVE reference model

Note that some NVE functions (e.g., data plane and control plane functions) may reside in one device or may be implemented separately in different devices. In addition, NVE functions can be implemented in a hierarchical fashion. For instance, an End Device can act as an NVE Spoke, while an access switch can act as an NVE hub.

2.3. NVE Service Types

An NVE provides different types of virtualized network services to multiple tenants, i.e. an L2 service or an L3 service. Note that an NVE may be capable of providing both L2 and L3 services for a tenant. This section defines the service types and associated attributes.

2.3.1. L2 NVE providing Ethernet LAN-like service

An L2 NVE implements Ethernet LAN emulation, an Ethernet based multipoint service similar to an IETF VPLS or EVPN service, where the Tenant Systems appear to be interconnected by a LAN environment over an L3 overlay. As such, an L2 NVE provides per-tenant virtual switching instance (L2 VNI), and L3 (IP/MPLS) tunneling encapsulation of tenant MAC frames across the underlay. Note that the control plane for an L2 NVE could be implemented locally on the NVE or in a separate control entity.

2.3.2. L3 NVE providing IP/VRF-like service

An L3 NVE provides Virtualized IP forwarding service, similar to IETF IP VPN (e.g., BGP/MPLS IPVPN [RFC4364]) from a service definition perspective. That is, an L3 NVE provides per-tenant forwarding and routing instance (L3 VNI), and L3 (IP/MPLS) tunneling encapsulation of tenant IP packets across the underlay. Note that routing could be performed locally on the NVE or in a separate control entity.

3. Functional components

This section decomposes the Network Virtualization architecture into functional components described in Figure 3 to make it easier to discuss solution options for these components.

3.1. Service Virtualization Components

3.1.1. Virtual Access Points (VAPs)

Tenant Systems are connected to VNIs through Virtual Access Points (VAPs).

VAPs can be physical ports or virtual ports identified through logical interface identifiers (e.g., VLAN ID, internal vSwitch Interface ID connected to a VM).

3.1.2. Virtual Network Instance (VNI)

A VNI is a specific VN instance on an NVE. Each VNI defines a forwarding context that contains reachability information and policies.

3.1.3. Overlay Modules and VN Context

Mechanisms for identifying each tenant service are required to allow the simultaneous overlay of multiple tenant services over the same underlay L3 network topology. In the data plane, each NVE, upon sending a tenant packet, must be able to encode the VN Context for the destination NVE in addition to the L3 tunneling information (e.g., source IP address identifying the source NVE and the destination IP address identifying the destination NVE, or MPLS label). This allows the destination NVE to identify the tenant service instance and therefore appropriately process and forward the tenant packet.

The Overlay module provides tunneling overlay functions: tunnel initiation/termination as in the case of stateful tunnels (see Section 3.1.4), and/or simply encapsulation/decapsulation of frames from VAPs/L3 underlay.

In a multi-tenant context, tunneling aggregates frames from/to different VNIs. Tenant identification and traffic demultiplexing are based on the VN Context identifier.

The following approaches can be considered:

- One VN Context identifier per Tenant: A globally unique (on a per-DC administrative domain) VN identifier is used to identify the corresponding VNI. Examples of such identifiers in existing technologies are IEEE VLAN IDs and ISID tags that identify virtual L2 domains when using IEEE 802.1aq and IEEE 802.1ah, respectively.

- One VN Context identifier per VNI: A per-VNI local value is automatically generated by the egress NVE, or a control plane associated with that NVE, and usually distributed by a control

plane protocol to all the related NVEs. An example of this approach is the use of per VRF MPLS labels in IP VPN [RFC4364].

- One VN Context identifier per VAP: A per-VAP local value is assigned and usually distributed by a control plane protocol. An example of this approach is the use of per CE-PE MPLS labels in IP VPN [RFC4364].

Note that when using one VN Context per VNI or per VAP, an additional global identifier (e.g., a VN identifier or name) may be used by the control plane to identify the Tenant context.

3.1.4. Tunnel Overlays and Encapsulation options

Once the VN context identifier is added to the frame, an L3 Tunnel encapsulation is used to transport the frame to the destination NVE.

Different IP tunneling options (e.g., GRE, L2TP, IPSec) and MPLS tunneling can be used. Tunneling could be stateless or stateful. Stateless tunneling simply entails the encapsulation of a tenant packet with another header necessary for forwarding the packet across the underlay (e.g., IP tunneling over an IP underlay). Stateful tunneling on the other hand entails maintaining tunneling state at the tunnel endpoints (i.e., NVEs). Tenant packets on an ingress NVE can then be transmitted over such tunnels to a destination (egress) NVE by encapsulating the packets with a corresponding tunneling header. The tunneling state at the endpoints may be configured or dynamically established. Solutions should specify the tunneling technology used, whether it is stateful or stateless. In this document, however, tunneling and tunneling encapsulation are used interchangeably to simply mean the encapsulation of a tenant packet with a tunneling header necessary to carry the packet between an ingress NVE and an egress NVE across the underlay. It should be noted that stateful tunneling, especially when configuration is involved, does impose management overhead and scale constraints. Thus, stateless tunneling is preferred when feasible.

3.1.5. Control Plane Components

3.1.5.1. Distributed vs Centralized Control Plane

A control/management plane entity can be centralized or distributed. Both approaches have been used extensively in the past. The routing model of the Internet is a good example of a distributed approach.

Transport networks have usually used a centralized approach to manage transport paths.

It is also possible to combine the two approaches, i.e., using a hybrid model. A global view of network state can have many benefits but it does not preclude the use of distributed protocols within the network. Centralized models provide a facility to maintain global state, and distribute that state to the network. When used in combination with distributed protocols, greater network efficiencies, improved reliability and robustness can be achieved. Domain and/or deployment specific constraints define the balance between centralized and distributed approaches.

3.1.5.2. Auto-provisioning/Service discovery

NVEs must be able to identify the appropriate VNI for each Tenant System. This is based on state information that is often provided by external entities. For example, in an environment where a VM is a Tenant System, this information is provided by VM orchestration systems, since these are the only entities that have visibility of which VM belongs to which tenant.

A mechanism for communicating this information to the NVE is required. VAPs have to be created and mapped to the appropriate VNI. Depending upon the implementation, this control interface can be implemented using an auto-discovery protocol between Tenant Systems and their local NVE or through management entities. In either case, appropriate security and authentication mechanisms to verify that Tenant System information is not spoofed or altered are required. This is one critical aspect for providing integrity and tenant isolation in the system.

NVEs may learn reachability information to VNIs on other NVEs via a control protocol exchanging such information among NVEs or via a management control entity.

3.1.5.3. Address advertisement and tunnel mapping

As traffic reaches an ingress NVE on a VAP, a lookup is performed to determine which NVE or local VAP the packet needs to be sent to. If the packet is to be sent to another NVE, the packet is encapsulated with a tunnel header containing the destination information (destination IP address or MPLS label) of the egress NVE. Intermediate nodes (between the ingress and egress NVEs) switch or route traffic based upon the tunnel destination information.

A key step in the above process consists of identifying the destination NVE the packet is to be tunneled to. NVEs are responsible for maintaining a set of forwarding or mapping tables that hold the bindings between destination VM and egress NVE addresses. Several ways of populating these tables are possible: control plane driven, management plane driven, or data plane driven.

When a control plane protocol is used to distribute address reachability and tunneling information, the auto-provisioning/Service discovery could be accomplished by the same protocol. In this scenario, the auto-provisioning/Service discovery could be combined with (be inferred from) the address advertisement and associated tunnel mapping. Furthermore, a control plane protocol that carries both MAC and IP addresses eliminates the need for ARP, and hence addresses one of the issues with explosive ARP handling.

3.1.5.4. Overlay Tunneling

For overlay tunneling, and dependent upon the tunneling technology used for encapsulating the Tenant System packets, it may be sufficient to have one or more local NVE addresses assigned and used in the source and destination fields of a tunneling encapsulating header. Other information that is part of the tunneling encapsulation header may also need to be configured. In certain cases, local NVE configuration may be sufficient while in other cases, some tunneling related information may need to be shared among NVEs. The information that needs to be shared will be technology dependent. For instance, potential information could include tunnel identity, encapsulation type, and/or tunnel resources. In certain cases, such as when using IP multicast in the underlay, tunnels may need to be established, interconnecting NVEs. When tunneling information needs to be exchanged or shared among NVEs, a control plane protocol may be required. For instance, it may be necessary to provide active/standby status information between NVEs, up/down status information, pruning/grafting information for multicast tunnels, etc.

In addition, a control plane may be required to setup the tunnel path for some tunneling technologies. This applies to both unicast and multicast tunneling.

3.2. Multi-homing

Multi-homing techniques can be used to increase the reliability of an NVO3 network. It is also important to ensure that physical

diversity in an NVO3 network is taken into account to avoid single points of failure.

Multi-homing can be enabled in various nodes, from Tenant Systems into TORs, TORs into core switches/routers, and core nodes into DC GWs.

The NVO3 underlay nodes (i.e. from NVEs to DC GWs) rely on IP routing as the means to re-route traffic upon failures techniques or on MPLS re-rerouting capabilities.

When a Tenant System is co-located with the NVE, the Tenant System is effectively single homed to the NVE via a virtual port. When the Tenant System and the NVE are separated, the Tenant System is connected to the NVE via a logical Layer2 (L2) construct such as a VLAN and it can be multi-homed to various NVEs. An NVE may provide an L2 service to the end system or an l3 service. An NVE may be multi-homed to a next layer in the DC at Layer2 (L2) or Layer3 (L3). When an NVE provides an L2 service and is not co-located with the end system, techniques such as Ethernet Link Aggregation Group (LAG) or Spanning Tree Protocol (STP) can be used to switch traffic between an end system and connected NVEs without creating loops. Similarly, when the NVE provides L3 service, similar dual-homing techniques can be used. When the NVE provides a L3 service to the end system, it is possible that no dynamic routing protocol is enabled between the end system and the NVE. The end system can be multi-homed to multiple physically-separated L3 NVEs over multiple interfaces. When one of the links connected to an NVE fails, the other interfaces can be used to reach the end system.

External connectivity out of a DC can be handled by two or more DC gateways. Each gateway provides access to external networks such as VPNs or the Internet. A gateway may be connected to two or more edge nodes in the external network for redundancy. When a connection to an upstream node is lost, the alternative connection is used and the failed route withdrawn.

3.3. VM Mobility

In DC environments utilizing VM technologies, an important feature is that VMs can move from one server to another server in the same or different L2 physical domains (within or across DCs) in a seamless manner.

A VM can be moved from one server to another in stopped or suspended state ("cold" VM mobility) or in running/active state ("hot" VM mobility). With "hot" mobility, VM L2 and L3 addresses need to be preserved. With "cold" mobility, it may be desired to preserve at least VM L3 addresses.

Solutions to maintain connectivity while a VM is moved are necessary in the case of "hot" mobility. This implies that connectivity among VMs is preserved. For instance, for L2 VNs, ARP caches are updated accordingly.

Upon VM mobility, NVE policies that define connectivity among VMs must be maintained.

During VM mobility, it is expected that the path to the VM's default gateway assures adequate performance to VM applications.

4. Key aspects of overlay networks

The intent of this section is to highlight specific issues that proposed overlay solutions need to address.

4.1. Pros & Cons

An overlay network is a layer of virtual network topology on top of the physical network.

Overlay networks offer the following key advantages:

- o Unicast tunneling state management and association of Tenant Systems reachability are handled at the edge of the network (at the NVE). Intermediate transport nodes are unaware of such state. Note that when multicast is enabled in the underlay network to build multicast trees for tenant VNs, there would be more state related to tenants in the underlay core network.

- o Tunneling is used to aggregate traffic and hide tenant addresses from the underlay network, and hence offer the advantage of minimizing the amount of forwarding state required within the underlay network

- o Decoupling of the overlay addresses (MAC and IP) used by VMs from the underlay network for tenant separation and separation of the tenant address spaces from the underlay address space.

- Support of a large number of virtual network identifiers

Overlay networks also create several challenges:

- Overlay networks have typically no control of underlay networks and lack underlay network information (e.g. underlay utilization):

  - Overlay networks and/or their associated management entities typically probe the network to measure link or path properties, such as available bandwidth or packet loss rate. It is difficult to accurately evaluate network properties. It might be preferable for the underlay network to expose usage and performance information.

  - Miscommunication or lack of coordination between overlay and underlay networks can lead to an inefficient usage of network resources.

  - When multiple overlays co-exist on top of a common underlay network, the lack of coordination between overlays can lead to performance issues and/or resource usage inefficiencies.

- Traffic carried over an overlay may not traverse firewalls and NAT devices.

- Multicast service scalability: Multicast support may be required in the underlay network to address tenant flood containment or efficient multicast handling. The underlay may also be required to maintain multicast state on a per-tenant basis, or even on a per-individual multicast flow of a given tenant. Ingress replication at the NVE eliminates that additional multicast state in the underlay core, but depending on the multicast traffic volume, it may cause inefficient use of bandwidth.

- Hash-based load balancing may not be optimal as the hash algorithm may not work well due to the limited number of combinations of tunnel source and destination addresses. Other NVO3 mechanisms may use additional entropy information than source and destination addresses.

4.2. Overlay issues to consider

4.2.1. Data plane vs Control plane driven

In the case of an L2 NVE, it is possible to dynamically learn MAC addresses against VAPs. It is also possible that such addresses be known and controlled via management or a control protocol for both L2 NVEs and L3 NVEs. Dynamic data plane learning implies that flooding of unknown destinations be supported and hence implies that broadcast and/or multicast be supported or that ingress replication be used as described in section 4.2.3. Multicasting in the underlay network for dynamic learning may lead to significant scalability limitations. Specific forwarding rules must be enforced to prevent loops from happening. This can be achieved using a spanning tree, a shortest path tree, or a split-horizon mesh.

It should be noted that the amount of state to be distributed is dependent upon network topology and the number of virtual machines. Different forms of caching can also be utilized to minimize state distribution between the various elements. The control plane should not require an NVE to maintain the locations of all the Tenant Systems whose VNs are not present on the NVE. The use of a control plane does not imply that the data plane on NVEs has to maintain all the forwarding state in the control plane.

4.2.2. Coordination between data plane and control plane

For an L2 NVE, the NVE needs to be able to determine MAC addresses of the Tenant Systems connected via a VAP. This can be achieved via dataplane learning or a control plane. For an L3 NVE, the NVE needs to be able to determine IP addresses of the Tenant Systems connected via a VAP.

In both cases, coordination with the NVE control protocol is needed such that when the NVE determines that the set of addresses behind a VAP has changed, it triggers the NVE control plane to distribute this information to its peers.

4.2.3. Handling Broadcast, Unknown Unicast and Multicast (BUM) traffic

There are several options to support packet replication needed for broadcast, unknown unicast and multicast. Typical methods include:

- o Ingress replication

- o Use of underlay multicast trees

There is a bandwidth vs state trade-off between the two approaches. Depending upon the degree of replication required (i.e. the number of hosts per group) and the amount of multicast state to maintain, trading bandwidth for state should be considered.

When the number of hosts per group is large, the use of underlay multicast trees may be more appropriate. When the number of hosts is small (e.g. 2-3) and/or the amount of multicast traffic is small, ingress replication may not be an issue.

Depending upon the size of the data center network and hence the number of (S,G) entries, but also the duration of multicast flows, the use of underlay multicast trees can be a challenge.

When flows are well known, it is possible to pre-provision such multicast trees. However, it is often difficult to predict application flows ahead of time, and hence programming of (S,G) entries for short-lived flows could be impractical.

A possible trade-off is to use in the underlay shared multicast trees as opposed to dedicated multicast trees.

4.2.4. Path MTU

When using overlay tunneling, an outer header is added to the original frame. This can cause the MTU of the path to the egress tunnel endpoint to be exceeded.

It is usually not desirable to rely on IP fragmentation for performance reasons. Ideally, the interface MTU as seen by a Tenant System is adjusted such that no fragmentation is needed. TCP will adjust its maximum segment size accordingly.

It is possible for the MTU to be configured manually or to be discovered dynamically. Various Path MTU discovery techniques exist in order to determine the proper MTU size to use:

- o Classical ICMP-based MTU Path Discovery [RFC1191] [RFC1981]

  - o

    Tenant Systems rely on ICMP messages to discover the MTU of the end-to-end path to its destination. This method is

not always possible, such as when traversing middle boxes (e.g. firewalls) which disable ICMP for security reasons

- o Extended MTU Path Discovery techniques such as defined in [RFC4821]

It is also possible to rely on the NVE to perform segmentation and reassembly operations without relying on the Tenant Systems to know about the end-to-end MTU. The assumption is that some hardware assist is available on the NVE node to perform such SAR operations. However, fragmentation by the NVE can lead to performance and congestion issues due to TCP dynamics and might require new congestion avoidance mechanisms from the underlay network [FLOYD].

Finally, the underlay network may be designed in such a way that the MTU can accommodate the extra tunneling and possibly additional NVO3 header encapsulation overhead.

4.2.5. NVE location trade-offs

In the case of DC traffic, traffic originated from a VM is native Ethernet traffic. This traffic can be switched by a local virtual switch or ToR switch and then by a DC gateway. The NVE function can be embedded within any of these elements.

There are several criteria to consider when deciding where the NVE function should happen:

- o Processing and memory requirements
  - o Datapath (e.g. lookups, filtering, encapsulation/decapsulation)
  - o Control plane processing (e.g. routing, signaling, OAM) and where specific control plane functions should be enabled
- o FIB/RIB size
- o Multicast support
  - o Routing/signaling protocols
  - o Packet replication capability
  - o Multicast FIB

- o Fragmentation support
- o QoS support (e.g. marking, policing, queuing)
- o Resiliency

4.2.6. Interaction between network overlays and underlays

When multiple overlays co-exist on top of a common underlay network, resources (e.g., bandwidth) should be provisioned to ensure that traffic from overlays can be accommodated and QoS objectives can be met. Overlays can have partially overlapping paths (nodes and links).

Each overlay is selfish by nature. It sends traffic so as to optimize its own performance without considering the impact on other overlays, unless the underlay paths are traffic engineered on a per overlay basis to avoid congestion of underlay resources.

Better visibility between overlays and underlays, or generally coordination in placing overlay demand on an underlay network, may be achieved by providing mechanisms to exchange performance and liveliness information between the underlay and overlay(s) or the use of such information by a coordination system. Such information may include:

- o Performance metrics (throughput, delay, loss, jitter)
- o Cost metrics

5. Security Considerations

NVO3 solutions must at least consider and address the following:

- . Secure and authenticated communication between an NVE and an NVE management system and/or control system.
- . Isolation between tenant overlay networks. The use of per-tenant FIB tables (VNIs) on an NVE is essential.
- . Security of any protocol used to carry overlay network information.
- . Preventing packets from reaching the wrong NVI, especially during VM moves.

. It may desirable to restrict the types of information that can be exchanged between overlays and underlays (e.g. topology information)

6. IANA Considerations

IANA does not need to take any action for this draft.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

7.2. Informative References

[NVOPS] Narten, T. et al, "Problem Statement : Overlays for Network Virtualization", draft-narten-nvo3-overlay-problem-statement (work in progress)

[OVCPREQ] Kreeger, L. et al, "Network Virtualization Overlay Control Protocol Requirements", draft-kreeger-nvo3-overlay-cp (work in progress)

[FLOYD] Sally Floyd, Allyn Romanow, "Dynamics of TCP Traffic over ATM Networks", IEEE JSAC, V. 13 N. 4, May 1995

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

[RFC1191] Mogul, J. "Path MTU Discovery", RFC1191, November 1990

[RFC1981] McCann, J. et al, "Path MTU Discovery for IPv6", RFC1981, August 1996

[RFC4821] Mathis, M. et al, "Packetization Layer Path MTU Discovery", RFC4821, March 2007

8. Acknowledgments

In addition to the authors the following people have contributed to this document:

Dimitrios Stiliadis, Rotem Salomonovitch, Lucy Yong, Thomas Narten, Larry Kreeger.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Marc Lasserre
Alcatel-Lucent
Email: marc.lasserre@alcatel-lucent.com

Florin Balus
Alcatel-Lucent
777 E. Middlefield Road
Mountain View, CA, USA 94043
Email: florin.balus@alcatel-lucent.com

Thomas Morin
France Telecom Orange
Email: thomas.morin@orange.com

Nabil Bitar
Verizon
40 Sylvan Road
Waltham, MA 02145
Email: nabil.bitar@verizon.com

Yakov Rekhter
Juniper
Email: yakov@juniper.net