Internet Engineering Task Force
Internet Draft
Intended status: Informational
Expires: August 2013

Marc Lasserre
Florin Balus
Alcatel-Lucent

Thomas Morin
France Telecom Orange

Nabil Bitar
Verizon

Yakov Rekhter
Juniper

February 4, 2013

Framework for DC Network Virtualization
draft-ietf-nvo3-framework-02.txt

Status of this Memo

Copyright Notice

Abstract

Several IETF drafts relate to the use of overlay networks to support large scale virtual data centers. This draft provides a framework for Network Virtualization over L3 (NVO3) and is intended to help plan a set of work items in order to provide a complete solution set. It defines a logical view of the main components with the intention of streamlining the terminology and focusing the solution set.

Table of Contents

1. Introduction....3
   1.1. Conventions used in this document....4
   1.2. General terminology....4
   1.3. DC network architecture....6
   1.4. Tenant networking view....7
2. Reference Models....8
   2.1. Generic Reference Model....8
   2.2. NVE Reference Model....10
   2.3. NVE Service Types....11
      2.3.1. L2 NVE providing Ethernet LAN-like service....12
      2.3.2. L3 NVE providing IP/VRF-like service....12
3. Functional components....12
   3.1. Service Virtualization Components....12
      3.1.1. Virtual Access Points (VAPs)....12
      3.1.2. Virtual Network Instance (VNI)....12
      3.1.3. Overlay Modules and VN Context....13
      3.1.4. Tunnel Overlays and Encapsulation options....14
      3.1.5. Control Plane Components....14
      3.1.5.1. Distributed vs Centralized Control Plane....14
      3.1.5.2. Auto-provisioning/Service discovery....15

3.1.5.3. Address advertisement and tunnel mapping.....15
3.1.5.4. Overlay Tunneling.....16
3.2. Multi-homing.....16
3.3. VM Mobility.....17
3.4. Service Overlay Topologies.....18
4. Key aspects of overlay networks.....18
4.1. Pros & Cons.....18
4.2. Overlay issues to consider.....20
4.2.1. Data plane vs Control plane driven.....20
4.2.2. Coordination between data plane and control plane.20
4.2.3. Handling Broadcast, Unknown Unicast and Multicast (BUM) traffic.....20
4.2.4. Path MTU.....21
4.2.5. NVE location trade-offs.....22
4.2.6. Interaction between network overlays and underlays.23
5. Security Considerations.....23
6. IANA Considerations.....24
7. References.....24
7.1. Normative References.....24
7.2. Informative References.....24
8. Acknowledgments.....24

1. Introduction

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

Several IETF drafts relate to the use of overlay networks 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. [OVCPREQ] describes the requirements for a control plane protocol required by overlay border nodes to exchange overlay mappings.

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 in the design of standards and mechanisms for large-scale data centers.

1.1. Conventions used in this document

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 RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

1.2. General terminology

This document uses the following terminology:

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, or a network service appliance.

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

VNI: Virtual Network Instance. This is one instance of a virtual overlay network. It refers to the state maintained for a given VN on a given NVE. 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 identifier 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.

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 sppliances 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 or Virtual DC: A container for virtualized compute, storage and network services. Managed by a single tenant, a Virtual DC can contain multiple VNs and multiple Tenant Systems that are connected to one or more of these VNs.

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 or router. It also enforces network isolation between VMs that should not communicate with each other.

Tenant: In a DC, a tenant refers to a customer that could be an organization within an enterprise, or an enterprise with a set of DC compute, storage and network resources associated with it.

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.

End device: A physical system to which networking service is provided. Examples 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). An end device may include internal networking functionality that interconnects the device's components (e.g. virtual switches that interconnect VMs running on the same server). NVE functionality may be implemented as part of that internal networking.

ELAN: MEF ELAN, multipoint to multipoint Ethernet service

EVPN: Ethernet VPN as defined in [EVPN]

1.3. DC network architecture

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

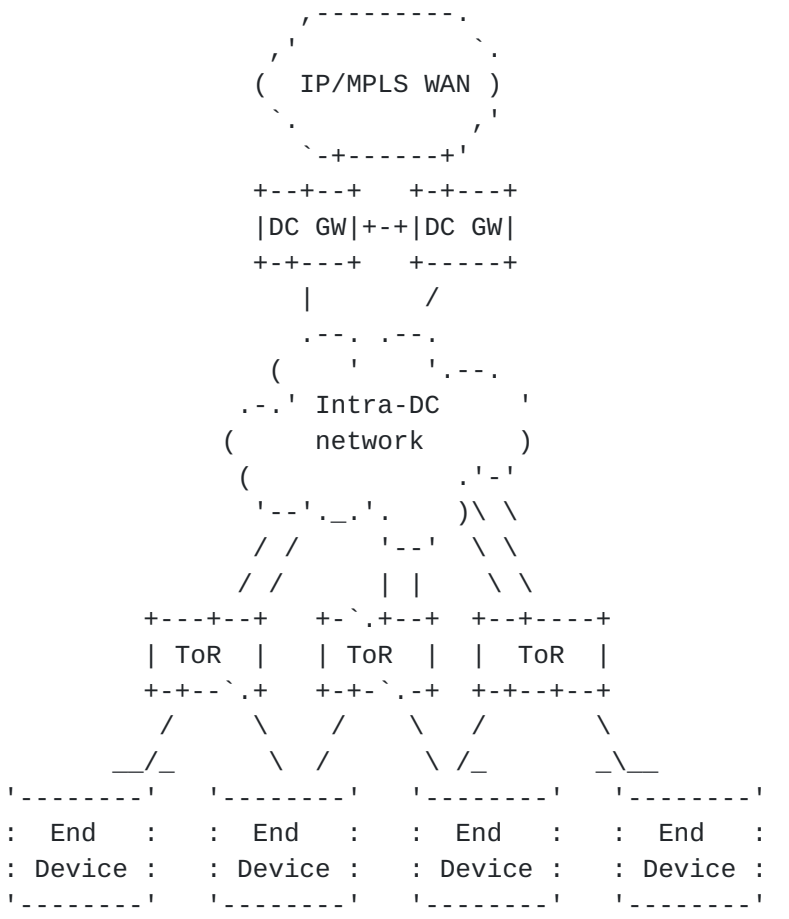

Figure 1 : A Generic Architecture for Data Centers

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

A cloud network is composed of intra-Data Center (DC) networks and network services, and inter-DC network and network connectivity services. Depending upon the scale, DC distribution, operations model, Capex and Opex aspects, DC networking elements can act as strict L2 switches and/or provide IP routing capabilities, including service virtualization.

In some DC architectures, it is possible that some tier layers provide L2 and/or L3 services, are collapsed, and that Internet connectivity, inter-DC connectivity and VPN support are handled by a smaller number of nodes. Nevertheless, one can assume that the functional blocks fit in the architecture above.

The following components can be present in a DC:

- Top of Rack (ToR): 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. ToRs may also provide routing functionality, virtual IP network connectivity, or Layer2 tunneling over IP for instance. ToRs are usually multi-homed to switches in the Intra-DC network. 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.

- Intra-DC Network: High capacity network composed of core switches aggregating multiple ToRs. Core switches are usually Ethernet switches but can also support routing capabilities.

- 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 IPVPN/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 ToRs.

1.4. Tenant networking view

The DC network architecture is used to provide L2 and/or L3 service connectivity to each tenant. An example is depicted in Figure 2:

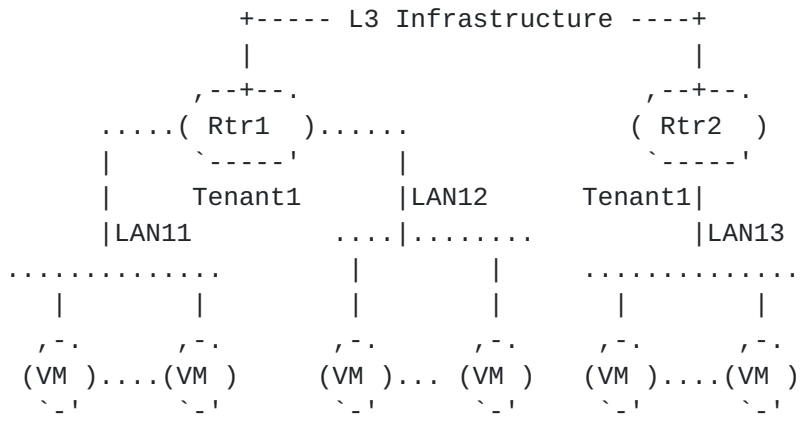

Figure 2 : Logical Service connectivity for a single tenant

In this example, one or more L3 contexts and one or more LANs (e.g., one per application type) are assigned for DC tenant1.

For a multi-tenant DC, a virtualized version of this type of service connectivity needs to be provided for each tenant by the Network Virtualization solution.

2. Reference Models

2.1. Generic Reference Model

The following diagram shows 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 tenant network.

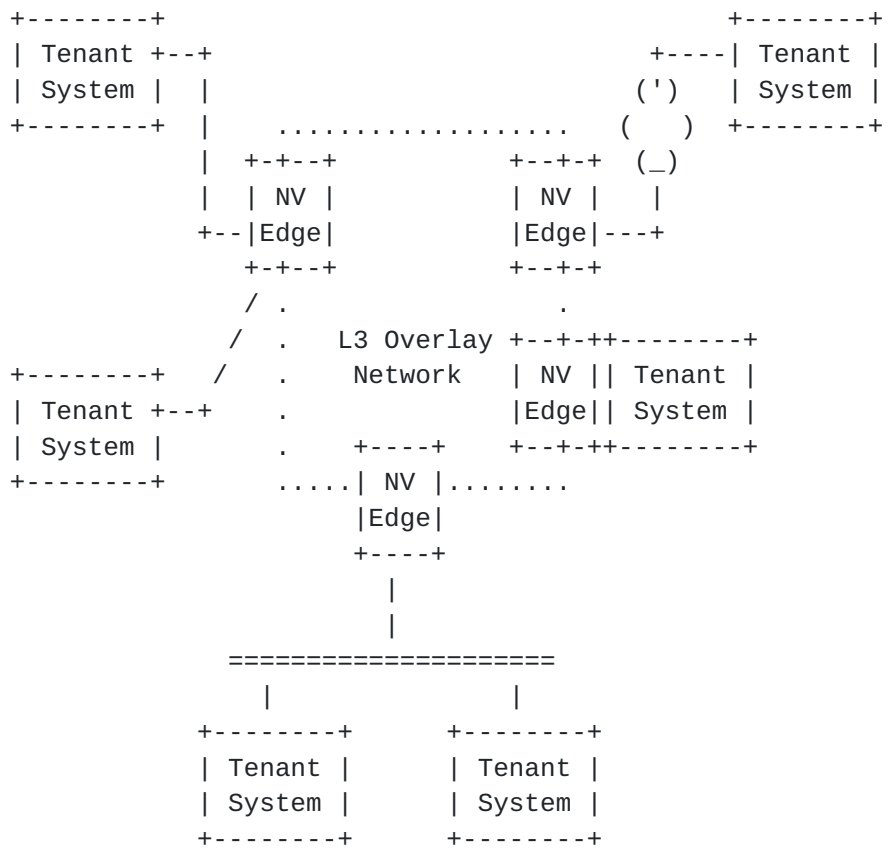

Figure 3 : Generic reference model for DC network virtualization over a Layer3 infrastructure

A Tenant System can be attached to a Network Virtualization Edge (NVE) node in several ways:

- locally, by being co-located in the same device

- remotely, via a point-to-point connection or a switched network (e.g., Ethernet)

When an NVE is local, the state of Tenant Systems 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 remote, the state of Tenant Systems needs to be exchanged via a data or control plane protocol, or via a management entity.

The functional components in Figure 3 do not necessarily map directly with the physical components described in Figure 1.

For example, an End Device can be a server blade with VMs and virtual switch, i.e. the VM is the Tenant System and the NVE functions may be performed by the virtual switch and/or the hypervisor. In this case, the Tenant System and NVE function are co-located.

Another example is the case where an End Device can be a traditional physical server (no VMs, no virtual switch), i.e. the server is the Tenant System and the NVE function may be performed by the ToR. Other End Devices in this category are physical network appliances or storage systems.

The NVE implements network virtualization functions that allow for L2 and/or L3 tenant separation and for hiding tenant addressing information (MAC and IP addresses), tenant-related control plane activity and service contexts from the underlay nodes.

Underlay nodes utilize L3 techniques to interconnect NVE nodes in support of the overlay network. These devices perform forwarding based on outer L3 tunnel header, 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. When such tenant or tenant-service related information is maintained in the underlay, overlay virtualization provides knobs to control the magnitude of that information.

2.2. NVE Reference Model

One or more VNIs can be instantiated on an NVE. Tenant Systems interface with a corresponding VNI via a Virtual Access Point (VAP). An overlay module that provides tunneling overlay functions (e.g., encapsulation and decapsulation of tenant traffic from/to the tenant forwarding instance, tenant identification and mapping, etc), as described in figure 4:

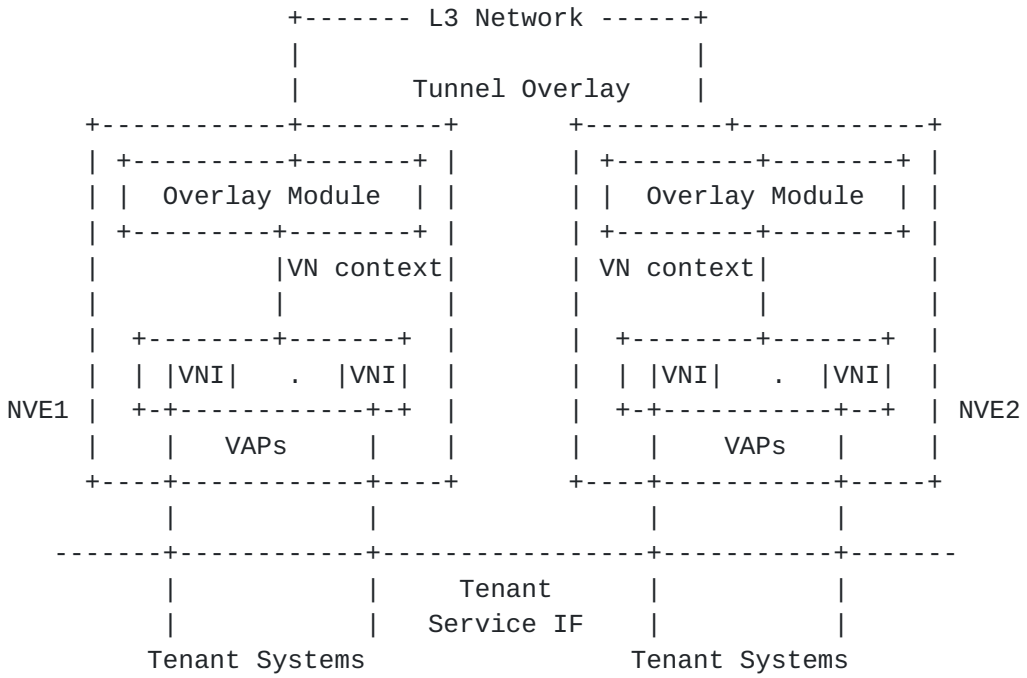

Figure 4 : Generic reference model for NV Edge

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. For example, the NVE functionality could reside solely on the End Devices, or be distributed between the End Devices and the ToRs. In the latter case we say that the End Device NVE component acts as the NVE Spoke, and ToRs act as NVE hubs. Tenant Systems will interface with VNIs maintained on the NVE spokes, and VNIs maintained on the NVE spokes will interface with VNIs maintained on the NVE hubs.

2.3. NVE Service Types

NVE components may be used to provide different types of virtualized network services. This section defines the service types and associated attributes. Note that an NVE may be capable of providing both L2 and L3 services.

2.3.1. L2 NVE providing Ethernet LAN-like service

L2 NVE implements Ethernet LAN emulation (ELAN), an Ethernet based multipoint service where the Tenant Systems appear to be interconnected by a LAN environment over a set of L3 tunnels. It provides per tenant virtual switching instance with MAC addressing isolation and L3 (IP/MPLS) tunnel encapsulation across the underlay.

2.3.2. L3 NVE providing IP/VRF-like service

Virtualized IP routing and forwarding is similar from a service definition perspective with IETF IP VPN (e.g., BGP/MPLS IPVPN [RFC4364] and IPsec VPNs). It provides per tenant routing instance with addressing isolation and L3 (IP/MPLS) tunnel encapsulation across the underlay.

3. Functional components

This section decomposes the Network Virtualization architecture into functional components described in Figure 4 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 the VNI Instance through Virtual Access Points (VAPs).

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

3.1.2. Virtual Network Instance (VNI)

The VNI represents a set of configuration attributes defining access and tunnel policies and (L2 and/or L3) forwarding functions.

Per tenant FIB tables and control plane protocol instances are used to maintain separate private contexts between tenants. Hence tenants are free to use their own addressing schemes without concerns about address overlapping with other tenants.

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 tunnel 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, encapsulation/decapsulation of frames from VAPs/L3 Backbone and may provide for transit forwarding of IP traffic (e.g., transparent tunnel forwarding).

In a multi-tenant context, the tunnel aggregates frames from/to different VNIs. Tenant identification and traffic demultiplexing are based on the VN Context identifier (e.g., VNID).

The following approaches can been considered:

- One VN Context per Tenant: A globally unique (on a per-DC administrative domain) VNID is used to identify the related Tenant instances. An example of this approach is the use of IEEE VLAN or ISID tags to provide virtual L2 domains.

- One VN Context per VNI: A per-tenant local value is automatically generated by the egress 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 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 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, a L3 Tunnel encapsulation is used to transport the frame to the destination NVE. The backbone devices do not usually keep any per service state, simply forwarding the frames based on the outer tunnel header.

Different IP tunneling options (e.g., GRE, L2TP, IPSec) and MPLS tunneling options (e.g., BGP VPN, VPLS) can be used.

3.1.5. Control Plane Components

Control plane components may be used to provide the following capabilities:

. Auto-provisioning/Service discovery

. Address advertisement and tunnel mapping

. Tunnel management

A control plane component can be an on-net control protocol implemented on the NVE or a management control entity.

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 controllers provide a facility to maintain global state, and distribute that state to the network which in combination with distributed protocols can aid in achieving greater network efficiencies, and improve reliability and robustness. Domain and/or deployment specific constraints define the balance between centralized and distributed approaches.

On one hand, a control plane module can reside in every NVE. This is how routing control plane modules are implemented in routers. At the same time, an external controller can manage a group of NVEs via an

agent in each NVE. This is how an SDN controller could communicate with the nodes it controls, via OpenFlow [OF] for instance.

In the case where a logically centralized control plane is preferred, the controller will need to be distributed to more than one node for redundancy and scalability in order to manage a large number of NVEs. Hence, inter-controller communication is necessary to synchronize state among controllers. It should be noted that controllers may be organized in clusters. The information exchanged between controllers of the same cluster could be different from the information exchanged across clusters.

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 compute management systems, since these are the only entities that have visibility of which VM belongs to which tenant.

A mechanism for communicating this information between Tenant Systems and the corresponding NVE is required. As a result the VAPs are 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.

A control plane protocol can also be used to advertize supported VNs to other NVEs. Alternatively, management control entities can also be used to perform these functions.

3.1.5.3. Address advertisement and tunnel mapping

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

One key step in this process consists of mapping a final destination information to the proper tunnel. NVEs are responsible for maintaining such mappings in their forwarding tables. 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 advertisement 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. This includes the discovery and announcement of the tunneling technology used. 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 and/or ECMP techniques or on MPLS re-rerouting capabilities.

When a tenant system is co-located with the NVE on the same end-system, the tenant system is single homed to the NVE via a vport that is virtual NIC (vNIC). When the end system and the NVEs are separated, the end system is connected to the NVE via a logical Layer2 (L2) construct such as a VLAN. In this latter case, an end device or vSwitch on that device could be multi-homed to various NVEs. An NVE may provide an L2 service to the end system or a 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 an nvo3 domain can be handled by two or more nvo3 gateways. Each gateway is connected to a different domain (e.g. ISP), providing access to external networks such as VPNs or the Internet. A gateway may be connected to two nodes. 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 VM L3 addresses.

Solutions to maintain connectivity while a VM is moved are necessary in the case of "hot" mobility. This implies that transport connections among VMs are preserved and that ARP caches are updated accordingly.

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

Optimal routing during VM mobility is also an important aspect to address. It is expected that the VM's default gateway be as close as possible to the server hosting the VM and triangular routing be avoided.

3.4. Service Overlay Topologies

A number of service topologies may be used to optimize the service connectivity and to address NVE performance limitations.

The topology described in Figure 3 suggests the use of a tunnel mesh between the NVEs where each tenant instance is one hop away from a service processing perspective. Partial mesh topologies and an NVE hierarchy may be used where certain NVEs may act as service transit points.

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:

- Unicast tunneling state management and association with tenant systems reachability are handled at the edge of the network. Intermediate transport nodes are unaware of such state. Note that this is not the case when multicast is enabled in the core network.

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

- Decoupling of the overlay addresses (MAC and IP) used by VMs from the underlay network. This offers a clear separation between addresses used within the overlay and the underlay networks and it enables the use of overlapping addresses spaces by Tenant Systems

- Support of a large number of virtual network identifiers

Overlay networks also create several challenges:

- Overlay networks have no controls of underlay networks and lack critical network information

    - Overlays 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.

- Overlaid traffic may not traverse firewalls and NAT devices.

- Multicast service scalability. Multicast support may be required in the underlay network to address for each tenant flood containment or efficient multicast handling. The underlay may be also be required to maintain multicast state on a per-tenant basis, or even on a per-individual multicast flow of a given tenant.

- 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 L2NVE, 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 L2NVEs and L3NVEs.

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 end 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 end 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 local NVE control plane to distribute this information to its peers.

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

There are two techniques to support packet replication needed for broadcast, unknown unicast and multicast:

- Ingress replication

- 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), 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.

In this section, we will only consider the case of an IP overlay.

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:

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

    - 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

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

It is also possible to rely on the overlay layer 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 overlay layer can lead to performance and congestion issues due to TCP dynamics and might require new congestion avoidance mechanisms from then 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:

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

- . 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.

- . Avoiding packets from reaching the wrong NVI, especially during VM moves.

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, Alcatel-Lucent

Lucy Yong, Huawei

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