Internet Engineering Task Force   Marc Lasserre
Internet Draft   Florin Balus
Intended status: Informational   Alcatel-Lucent
Expires: March 2013

                                                    Thomas Morin
                                           France Telecom Orange

                                                     Nabil Bitar
                                                         Verizon

                                                   Yakov Rekhter
                                                         Juniper

                                             September 11, 2012

Framework for DC Network Virtualization
draft-ietf-nvo3-framework-00.txt

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

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 data center networks. The use of virtualization leads to
a very large number of communication domains and end systems to cope
with.

[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, a Network Service Appliance or even be embedded within an End Station.

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

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

Virtual Network Context or VN Context: Field that is part of the overlay encapsulation header which allows the encapsulated frame to be delivered to the appropriate virtual network endpoint by the egress NVE. The egress NVE uses this field to determine the appropriate virtual network context in which to process the packet. This field MAY be an explicit, unique (to the administrative domain) virtual network identifier (VNID) or MAY express the necessary context information in other ways (e.g. a locally significant identifier).

VNID: Virtual Network Identifier. In the case where the VN context has global significance, this is the ID value that is carried in each data packet in the overlay encapsulation that identifies the Virtual Network the packet belongs to.

Underlay or Underlying Network: This is the network that provides the connectivity between NVEs. The Underlying Network can be

completely unaware of the overlay packets. Addresses within the
Underlying Network are also referred to as "outer addresses" because
they exist in the outer encapsulation. The Underlying Network can
use a completely different protocol (and address family) from that
of the overlay.

Data Center (DC): A physical complex housing physical servers,
network switches and routers, Network Service Appliances and
networked storage. The purpose of a Data Center is to provide
application and/or compute and/or storage services. One such service
is virtualized data center services, also known as Infrastructure as
a Service.

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

Tenant: A customer who consumes virtualized data center services
offered by a cloud service provider. A single tenant may consume one
or more Virtual Data Centers hosted by the same cloud service
provider.

Tenant End System: It defines an end system of a particular tenant,
which can be for instance a virtual machine (VM), a non-virtualized
server, or a physical appliance.

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:

```
                     ,---------.
                   ,'           `.
                  (   IP/MPLS WAN )
                   `.           ,'
                     `-+------+'
                 +--+--+    +-+---+
                 |DC GW|+-+|DC GW|
                 +-+---+    +-----+
                   |         /
                 .--. .--.
                 (      '     '.--.
                .-.' Intra-DC     '
               (      network      )
              (                  .'-'
               '--'._.'.     )\ \
                / /      '--'  \ \
               / /        | |    \ \
          +---+--+    +-`.+--+  +--+----+
          | ToR  |    | ToR  |  | ToR   |
          +-+--`.+    +-+-`.-+  +-+--+--+
           .'     \   .'    \  .'      `.
          __/_     _i./      i./_      _\__
      '--------'  '--------'  '--------'   '--------'
      : End   :   : End   :   : End   :   : End    :
      : Device :  : Device :  : Device :  : Device :
      '--------'  '--------'  '--------'   '--------'
```
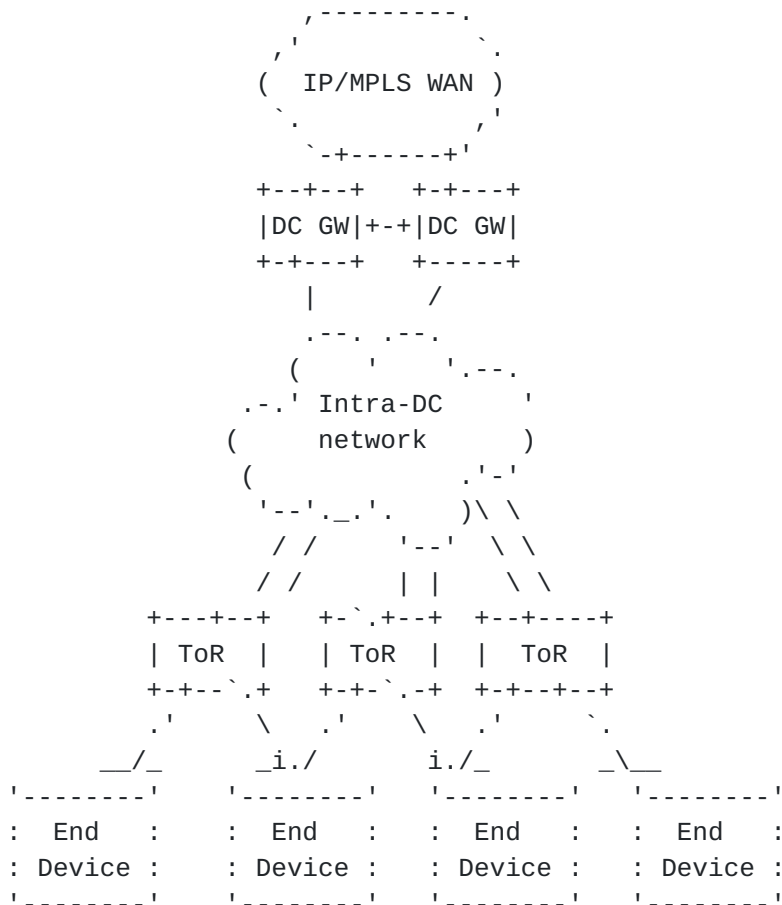
           Figure 1 : A Generic Architecture for Data Centers

An example of multi-tier DC network architecture is presented in
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
also 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 with the architecture above.

The following components can be present in a DC:

   o End Device: a DC resource to which the networking service is
     provided. End Device may be a compute resource (server or
     server blade), storage component or a network appliance
     (firewall, load-balancer, IPsec gateway). Alternatively, the
     End Device may include software based networking functions used
     to interconnect multiple hosts. An example of soft networking
     is the virtual switch in the server blades, used to
     interconnect multiple virtual machines (VMs). End Device may be
     single or multi-homed to the Top of Rack switches (ToRs).

   o 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.

   o 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.

   o 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).

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:

```
                    +----- L3 Infrastructure ----+
                    |                             |
                 ,--+-'.                      ;--+--.
              .....  Rtr1 )......          .  Rtr2 )
              |     '-----'      |             '-----'
              |     Tenant1     |LAN12     Tenant1|
              |LAN11         ....|........        |LAN13
            '':''''''':'        |        |    '':''''''':'
            ,'.      ,'.      ,+.      ,+.    ,'.      ,'.
           (VM )....(VM )    (VM )... (VM )  (VM )....(VM )
            `-'      `-'      `-'      `-'    `-'      `-'
```
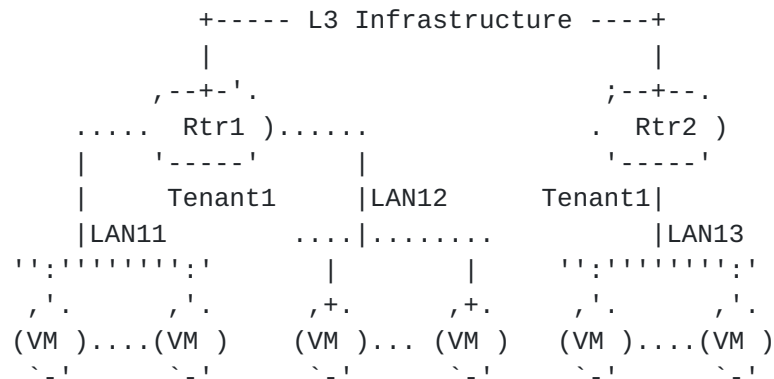
        Figure 2 : Logical Service connectivity for a single tenant

   In this example one or more L3 contexts and one or more LANs (e.g.,
   one per application type) running on DC switches are assigned for DC
   tenant 1.

   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 Layer3 overlays where edge devices provide a
   logical interconnect between Tenant End Systems that belong to
   specific tenant network.

```
         +--------+                                   +--------+
         | Tenant |                                   | Tenant |
         |  End   +--+                          +---|  End   |
         | System |  |                          |   | System |
         +--------+  |    ..................     |   +--------+
               |  +-+--+            +--+-+  |
               |  | NV |            | NV |  |
              +--|Edge|            |Edge|--+
                +-+--+            +--+-+
                 /  .    L3 Overlay   .  \
         +--------+  /  .     Network    .   \    +--------+
         | Tenant +--+   .                .   +----| Tenant |
         |  End   |      .                .        |  End   |
         | System |      .   +----+        .        | System |
         +--------+      .....| NV |........        +--------+
                            |Edge|
                            +----+
                              |
                              |
                         +--------+
                         | Tenant |
                         |  End   |
                         | System |
                         +--------+
```
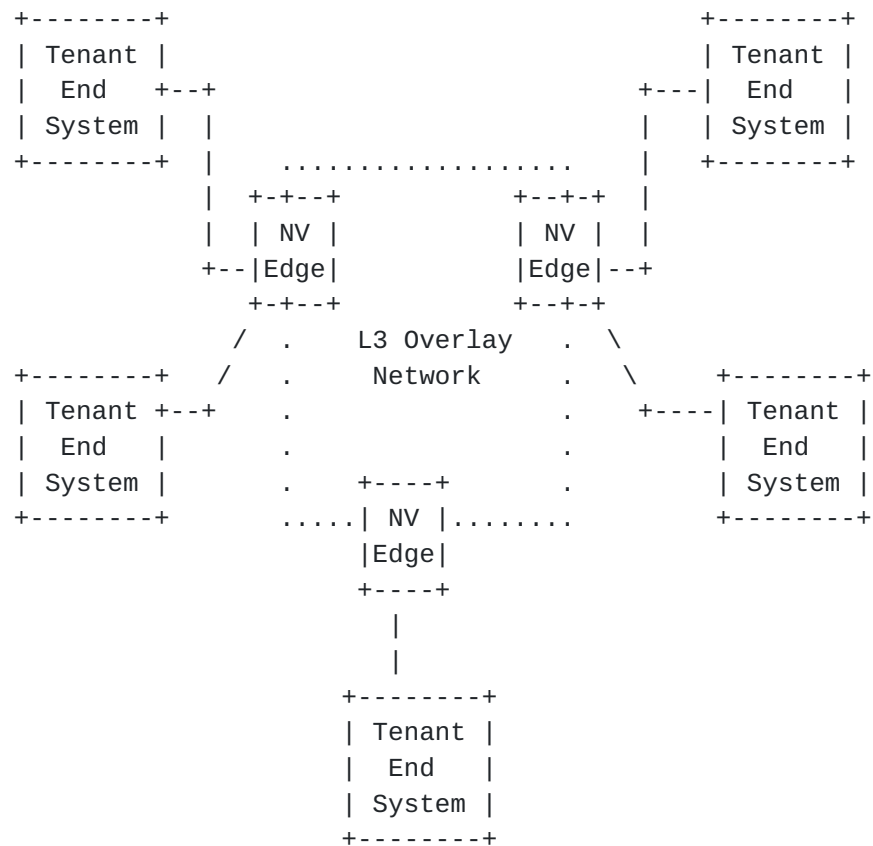
           Figure 3 : Generic reference model for DC network virtualization
                        over a Layer3 infrastructure

      The functional components in this picture 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 End System and the NVE
      functions may be performed by the virtual switch and/or the
      hypervisor.

      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 End System and the NVE functions may be performed by the ToR.
      Other End Devices in this category are Physical Network Appliances
   or Storage Systems.

A Tenant End System attaches to a Network Virtualization Edge (NVE)
node, either directly or via a switched network (typically
Ethernet).

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 Routed Backbone nodes.

Core 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 tunnels. When such tenant or tenant-
service related information is maintained in the core, overlay
virtualization provides knobs to control that information.

2.2. NVE Reference Model

The NVE is composed of a tenant service instance that Tenant End
Systems interface with and an overlay module that provides tunneling
overlay functions (e.g. encapsulation/decapsulation of tenant
traffic from/to the tenant forwarding instance, tenant
identification and mapping, etc), as described in figure 4:

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

```
           |            |     Tenant     |            |
           |            |   Service IF   |            |
        Tenant End Systems           Tenant End Systems
```
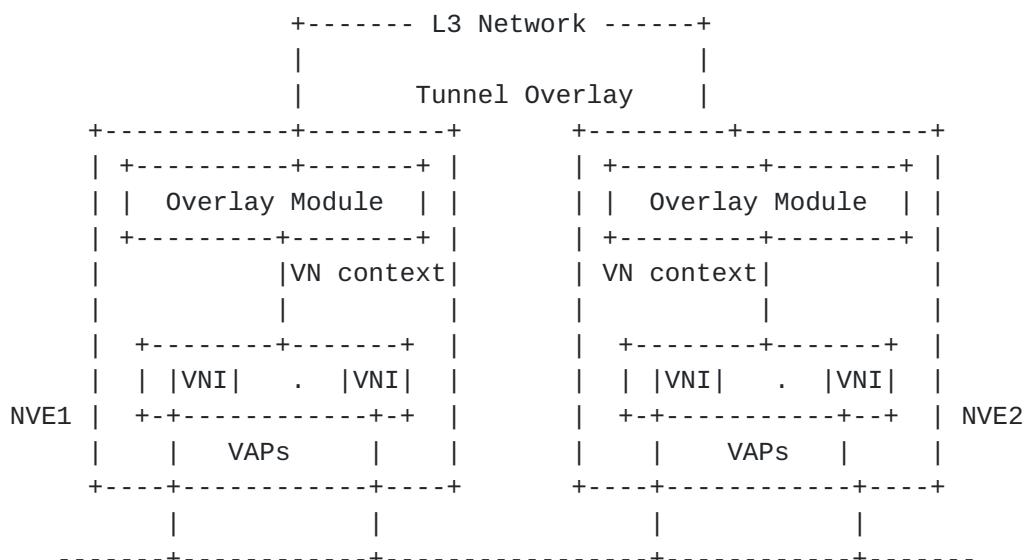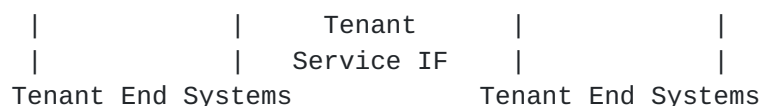
Figure 4 : Generic reference model for NV Edge

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, on the ToRs or on both the End Devices and the ToRs. In the
latter case we say that the the End Device NVE component acts as the
NVE Spoke, and ToRs act as NVE hubs. Tenant End Systems will
interface with the tenant service instances maintained on the NVE
spokes, and tenant service instances maintained on the NVE spokes
will interface with the tenant service instances maintained on the
NVE hubs.

2.3. NVE Service Types

NVE components may be used to provide different types of virtualized
service connectivity. This section defines the service types and
associated attributes

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 End 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 tunnel encapsulation across the core.

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 and
IPsec VPNs). It provides per tenant routing instance with addressing
isolation and L3 tunnel encapsulation across the core.

3. Functional components

This section breaks down the Network Virtualization architecture
into functional components to make it easier to discuss solution
options for different modules.

This version of the document gives an overview of generic functional
components that are shared between L2 and L3 service types. Details
specific for each service type will be added in future revisions.

3.1. Generic service virtualization components

A Network Virtualization solution is built around a number of
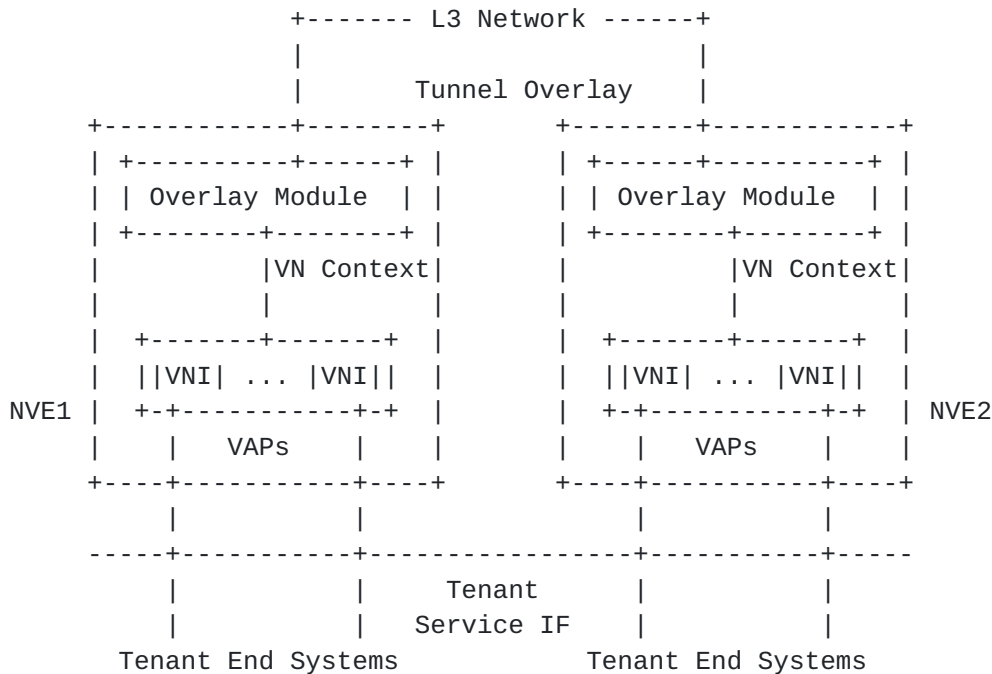functional components as depicted in Figure 5:

```
                      +------- L3 Network ------+
                      |                         |
                      |       Tunnel Overlay    |
          +-----------+-------+      +--------+-----------+
          | +---------+------+ |      | +------+---------+ |
          | | Overlay Module | |      | | Overlay Module | |
          | +--------+-------+ |      | +--------+-------+ |
          |          |VN Context|     |          |VN Context|
          |          |        |       |          |        |
          |   +------+------+  |      |   +------+------+  |
          |   ||VNI| ... |VNI||  |      |   ||VNI| ... |VNI||  |
     NVE1 |   +-+-----------+-+  |      |   +-+-----------+-+  | NVE2
          |     |    VAPs    |   |      |     |    VAPs    |   |
          +----+-----------+----+      +----+-----------+----+
               |           |                |           |
          -----+-----------+----------------+-----------+-----
               |           |     Tenant     |           |
               |           |   Service IF   |           |
            Tenant End Systems          Tenant End Systems
```

Figure 5 : Generic reference model for NV Edge

3.1.1. Virtual Access Points (VAPs)

Tenant End Systems are connected to the VNI Instance through Virtual
Access Points (VAPs). The VAPs can be in reality physical ports on a
ToR or virtual ports identified through logical interface
identifiers (VLANs, internal VSwitch Interface ID leading to a VM).

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 source address identifying the source NVE and the tunnel destination L3 address identifying the destination NVE. 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 (e.g. VNID).

The following approaches can been considered:

  o 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.

  o 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].

  o 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 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 (GRE/L2TP/IPSec) and tunneling
options (BGP VPN, PW, VPLS) are available for both Ethernet and IP
formats.

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 or a
management control entity.

3.1.5.1. Auto-provisioning/Service discovery

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

A mechanism for communicating this information between Tenant End
Systems and the local NVE is required. As a result the VAPs are
created and mapped to the appropriate Tenant Instance.

Depending upon the implementation, this control interface can be
implemented using an auto-discovery protocol between Tenant End
Systems and their local NVE or through management entities.

When a protocol is used, appropriate security and authentication
mechanisms to verify that Tenant End System information is not
spoofed or altered are required. This is one critical aspect for
providing integrity and tenant isolation in the system.

Another control plane protocol can also be used to advertize NVE
tenant service instance (tenant and service type provided to the
tenant) to other NVEs. Alternatively, management control entities
can also be used to perform these functions.

3.1.5.2. 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 address
of the egress overlay node. Intermediate nodes (between the ingress
and egress NVEs) switch or route traffic based upon the outer
destination address.

One key step in this process consists of mapping a final destination
address to the proper tunnel. NVEs are responsible for maintaining
such mappings in their lookup tables. Several ways of populating
these lookup 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 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.

3.1.5.3. Tunnel management

A control plane protocol may be required to exchange tunnel state
information. This may include setting up tunnels and/or providing
tunnel state information.

This applies to both unicast and multicast tunnels.

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.

3.2. 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:

o Unicast tunneling state management is 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.

o Tunnels are used to aggregate traffic 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. 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 End Systems

o Support of a large number of virtual network identifiers

Overlay networks also create several challenges:

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

o Overlays typically probe the network to measure link
  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.

o Miscommunication between overlay and underlay networks can lead
  to an inefficient usage of network resources.

o Fairness of resource sharing and collaboration among end-nodes
  in overlay networks are two critical issues

o When multiple overlays co-exist on top of a common underlay
  network, the lack of coordination between overlays can lead to
  performance issues.

o Overlaid traffic may not traverse firewalls and NAT devices.

o Multicast service scalability. Multicast support may be
  required in the overlay network to address for each tenant
  flood containment or efficient multicast handling.

o 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

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. Multicasting in the core 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.

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 present on a VAP (for instance, dataplane
learning may be relied upon for this purpose). For an L3 NVE, the
NVE needs to be able to determine IP addresses of the end systems
present on 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:

   o Ingress replication

   o Use of core 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 is of consideration.

When the number of hosts per group is large, the use of core
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 core 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 core 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
End 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 End 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 overlay layer to perform
segmentation and reassembly operations without relying on the Tenant
End 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 tunnel 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 VM 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
processing boundary happens:

   o Processing and memory requirements

      o Datapath (e.g. lookups, filtering,
         encapsulation/decapsulation)

      o Control plane processing (e.g. routing, signaling, OAM)

   o FIB/RIB size

   o Multicast support

      o Routing protocols

      o Packet replication capability

   o Fragmentation support

   o QoS transparency

   o Resiliency

[4.2.6](#). **Interaction between network overlays and underlays**

When multiple overlays co-exist on top of a common underlay network,
this can cause some performance issues. These overlays have
partially overlapping paths and nodes.

Each overlay is selfish by nature in that it sends traffic so as to
optimize its own performance without considering the impact on other
overlays, unless the underlay tunnels are traffic engineered on a
per overlay basis so as to avoid sharing underlay resources.

Better visibility between overlays and underlays can be achieved by
providing mechanisms to exchange information about:

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

   o Cost metrics

5. Security Considerations

The tenant to overlay mapping function can introduce significant security risks if appropriate protocols are not used that can support mutual authentication.

No other new security issues are introduced beyond those described already in the related L2VPN and L3VPN RFCs.

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

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