

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: December 17, 2012

T. Narten, Ed.  
IBM  
M. Sridharan  
Microsoft  
D. Dutt

D. Black  
EMC  
L. Kreeger  
Cisco

June 15, 2012

Problem Statement: Overlays for Network Virtualization  
draft-narten-nvo3-overlay-problem-statement-02

## Abstract

This document describes issues associated with providing multi-tenancy in large data center networks and an overlay-based network virtualization approach to addressing them. A key multi-tenancy requirement is traffic isolation, so that a tenant's traffic is not visible to any other tenant. This isolation can be achieved by assigning one or more virtual networks to each tenant such that traffic within a virtual network is isolated from traffic in other virtual networks. The primary functionality required is provisioning virtual networks, associating a virtual machine's NIC with the appropriate virtual network, and maintaining that association as the virtual machine is activated, migrated and/or deactivated. Use of an overlay-based approach enables scalable deployment on large network infrastructures.

## Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 17, 2012.

## Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                      |   |                    |
|----------------------|---|--------------------|
| <a href="#">1.</a>   | <a href="#">Introduction</a>  | <a href="#">4</a>  |
| <a href="#">2.</a>   | <a href="#">Problem Details</a>   | <a href="#">5</a>  |
| <a href="#">2.1.</a> | <a href="#">Multi-tenant Environment Scale</a>                                | <a href="#">5</a>  |
| <a href="#">2.2.</a> | <a href="#">Virtual Machine Mobility Requirements</a>                         | <a href="#">5</a>  |
| <a href="#">2.3.</a> | <a href="#">Span of Virtual Networks</a>                                      | <a href="#">6</a>  |
| <a href="#">2.4.</a> | <a href="#">Inadequate Forwarding Table Sizes in Switches</a>                 | <a href="#">6</a>  |
| <a href="#">2.5.</a> | <a href="#">Decoupling Logical and Physical Configuration</a>                 | <a href="#">6</a>  |
| <a href="#">2.6.</a> | <a href="#">Support Communication Between VMs and Non-virtualized Devices</a> | <a href="#">7</a>  |
| <a href="#">2.7.</a> | <a href="#">Overlay Design Characteristics</a>                                | <a href="#">7</a>  |
| <a href="#">3.</a>   | <a href="#">Network Overlays</a>  | <a href="#">8</a>  |
| <a href="#">3.1.</a> | <a href="#">Limitations of Existing Virtual Network Models</a>                | <a href="#">8</a>  |
| <a href="#">3.2.</a> | <a href="#">Benefits of Network Overlays</a>                                  | <a href="#">9</a>  |
| <a href="#">3.3.</a> | <a href="#">Overlay Networking Work Areas</a>                                 | <a href="#">10</a> |
| <a href="#">4.</a>   | <a href="#">Related Work</a>  | <a href="#">12</a> |
| <a href="#">4.1.</a> | <a href="#">IEEE 802.1aq - Shortest Path Bridging</a>                         | <a href="#">12</a> |
| <a href="#">4.2.</a> | <a href="#">ARMD</a>  | <a href="#">12</a> |
| <a href="#">4.3.</a> | <a href="#">TRILL</a>   | <a href="#">12</a> |
| <a href="#">4.4.</a> | <a href="#">L2VPNs</a>  | <a href="#">13</a> |
| <a href="#">4.5.</a> | <a href="#">Proxy Mobile IP</a>   | <a href="#">13</a> |
| <a href="#">4.6.</a> | <a href="#">LISP</a>  | <a href="#">13</a> |
| <a href="#">4.7.</a> | <a href="#">Individual Submissions</a>  | <a href="#">13</a> |
| <a href="#">5.</a>   | <a href="#">Further Work</a>  | <a href="#">14</a> |
| <a href="#">6.</a>   | <a href="#">Summary</a>   | <a href="#">14</a> |
| <a href="#">7.</a>   | <a href="#">Acknowledgments</a>   | <a href="#">14</a> |

|                             |                                   |                    |
|-----------------------------|-----------------------------------|--------------------|
| <a href="#">8.</a>          | IANA Considerations . . . . .     | <a href="#">14</a> |
| <a href="#">9.</a>          | Security Considerations . . . . . | <a href="#">14</a> |
| <a href="#">10.</a>         | Informative References . . . . .  | <a href="#">14</a> |
| <a href="#">Appendix A.</a> | Change Log . . . . .              | <a href="#">16</a> |
| <a href="#">A.1.</a>        | Changes from -01 . . . . .        | <a href="#">16</a> |

|                              |                    |
|------------------------------|--------------------|
| Authors' Addresses . . . . . | <a href="#">16</a> |
|------------------------------|--------------------|

## 1. Introduction

Server virtualization is increasingly becoming the norm in data centers. With server virtualization, each physical server supports multiple virtual machines (VMs), each running its own operating system, middleware and applications. Virtualization is a key enabler of workload agility, i.e., allowing any server to host any application and providing the flexibility of adding, shrinking, or moving services within the physical infrastructure. Server virtualization provides numerous benefits, including higher utilization, increased data security, reduced user downtime, reduced power usage, etc.

Large scale multi-tenant data centers are taking advantage of the benefits of server virtualization to provide a new kind of hosting, a virtual hosted data center. Multi-tenant data centers are ones where individual tenants could belong to a different company (in the case of a public provider) or a different department (in the case of an internal company data center). Each tenant has the expectation of a level of security and privacy separating their resources from those of other tenants. For example, one tenant's traffic must never be exposed to another tenant, except through carefully controlled interfaces, such as a security gateway.

To a tenant, virtual data centers are similar to their physical counterparts, consisting of end stations attached to a network, complete with services such as load balancers and firewalls. But unlike a physical data center, end stations connect to a virtual network. To end stations, a virtual network looks like a normal

network (e.g., providing an ethernet service), except that the only end stations connected to the virtual network are those belonging to the tenant.

A tenant is the administrative entity that is responsible for and manages a specific virtual network instance and its associated services (whether virtual or physical). In a cloud environment, a tenant would correspond to the customer that has defined and is using a particular virtual network. However, a tenant may also find it useful to create multiple different virtual network instances. Hence, there is a one-to-many mapping between tenants and virtual network instances. A single tenant may operate multiple individual virtual network instances, each associated with a different service.

How a virtual network is implemented does not matter to the tenant. It could be a pure routed network, a pure bridged network or a combination of bridged and routed networks. The key requirement is that each individual virtual network instance be isolated from other virtual network instances.

This document outlines the problems encountered in scaling the number of isolated networks in a data center, as well as the problems of managing the creation/deletion, membership and span of these networks and makes the case that an overlay based approach, where individual networks are implemented within individual virtual networks that are dynamically controlled by a standardized control plane provides a number of advantages over current approaches. The purpose of this document is to identify the set of problems that any solution has to address in building multi-tenant data centers. With this approach, the goal is to allow the construction of standardized, interoperable implementations to allow the construction of multi-tenant data centers.

[Section 2](#) describes the problem space details. [Section 3](#) describes network overlays in more detail and the potential work areas. Sections [4](#) and [5](#) review related and further work, while [Section 6](#) closes with a summary.

## [2.](#) Problem Details

The following subsections describe aspects of multi-tenant networking

that pose problems for large scale network infrastructure. Different problem aspects may arise based on the network architecture and scale.

### [2.1.](#) Multi-tenant Environment Scale

Cloud computing involves on-demand elastic provisioning of resources for multi-tenant environments. A common example of cloud computing is the public cloud, where a cloud service provider offers these elastic services to multiple customers over the same infrastructure. This elastic on-demand nature in conjunction with trusted hypervisors to control network access by VMs calls for resilient distributed network control mechanisms.

### [2.2.](#) Virtual Machine Mobility Requirements

A key benefit of server virtualization is virtual machine (VM) mobility. A VM can be migrated from one server to another, live i.e. as it continues to run and without shutting down the VM and restarting it at a new location. A key requirement for live migration is that a VM retain its IP address(es) and MAC address(es) in its new location (to avoid tearing down existing communication). Today, servers are assigned IP addresses based on their physical location, typically based on the ToR (Top of Rack) switch for the server rack or the VLAN configured to the server. This works well for physical servers, which cannot move, but it restricts the

placement and movement of the more mobile VMs within the data center (DC). Any solution for a scalable multi-tenant DC must allow a VM to be placed (or moved to) anywhere within the data center, without being constrained by the subnet boundary concerns of the host servers.

### [2.3.](#) Span of Virtual Networks

Another use case is cross pod expansion. A pod typically consists of one or more racks of servers with its associated network and storage connectivity. Tenants may start off on a pod and, due to expansion, require servers/VMs on other pods, especially the case when tenants on the other pods are not fully utilizing all their resources. This use case requires that virtual networks span multiple pods in order to provide connectivity to all of the tenant's servers/VMs.

#### [2.4.](#) Inadequate Forwarding Table Sizes in Switches

Today's virtualized environments place additional demands on the forwarding tables of switches. Instead of just one link-layer address per server, the switching infrastructure has to learn addresses of the individual VMs (which could range in the 100s per server). This is a requirement since traffic from/to the VMs to the rest of the physical network will traverse the physical network infrastructure. This places a much larger demand on the switches' forwarding table capacity compared to non-virtualized environments, causing more traffic to be flooded or dropped when the addresses in use exceeds the forwarding table capacity.

#### [2.5.](#) Decoupling Logical and Physical Configuration

Data center operators must be able to achieve high utilization of server and network capacity. For efficient and flexible allocation, operators should be able to spread a virtual network instance across servers in any rack in the data center. It should also be possible to migrate compute workloads to any server anywhere in the network while retaining the workload's addresses. This can be achieved today by stretching VLANs (e.g., by using TRILL or SPB).

However, in order to limit the broadcast domain of each VLAN, multi-destination frames within a VLAN should optimally flow only to those devices that have that VLAN configured. When workloads migrate, the physical network (e.g., access lists) may need to be reconfigured which is typically time consuming and error prone.

#### [2.6.](#) Support Communication Between VMs and Non-virtualized Devices

Within data centers, not all communication will be between VMs. Network operators will continue to use non-virtualized servers for various reasons, traditional routers to provide L2VPN and L3VPN services, traditional load balancers, firewalls, intrusion detection engines and so on. Any virtual network solution should be capable of working with these existing systems.

## [2.7.](#) Overlay Design Characteristics

There are existing layer 2 overlay protocols in existence, but they were not necessarily designed to solve the problem in the environment of a highly virtualized data center. Below are some of the characteristics of environments that must be taken into account by the overlay technology:

1. Highly distributed systems. The overlay should work in an environment where there could be many thousands of access switches (e.g. residing within the hypervisors) and many more end systems (e.g. VMs) connected to them. This leads to a distributed mapping system that puts a low overhead on the overlay tunnel endpoints.
2. Many highly distributed virtual networks with sparse membership. Each virtual network could be highly dispersed inside the data center. Also, along with expectation of many virtual networks, the number of end systems connected to any one virtual network is expected to be relatively low; Therefore, the percentage of access switches participating in any given virtual network would also be expected to be low. For this reason, efficient pruning of multi-destination traffic should be taken into consideration.
3. Highly dynamic end systems. End systems connected to virtual networks can be very dynamic, both in terms of creation/deletion/power-on/off and in terms of mobility across the access switches.
4. Work with existing, widely deployed network Ethernet switches and IP routers without requiring wholesale replacement. The first hop switch that adds and removes the overlay header will require new equipment and/or new software.
5. Network infrastructure administered by a single administrative domain. This is consistent with operation within a data center, and not across the Internet.

## [3.](#) Network Overlays



Virtual Networks are used to isolate a tenant's traffic from that of other tenants (or even traffic within the same tenant that requires isolation). There are two main characteristics of virtual networks:

1. Providing network address space that is isolated from other virtual networks. The same network addresses may be used in different virtual networks on the same underlying network infrastructure.
2. Limiting the scope of frames sent on the virtual network. Frames sent by end systems attached to a virtual network are delivered as expected to other end systems on that virtual network and may exit a virtual network only through controlled exit points such as a security gateway. Likewise, frames sourced outside of the virtual network may enter the virtual network only through controlled entry points, such as a security gateway.

### 3.1. Limitations of Existing Virtual Network Models

Virtual networks are not new to networking. For example, VLANs are a well known construct in the networking industry. A VLAN is an L2 bridging construct that provides some of the semantics of virtual networks mentioned above: a MAC address is unique within a VLAN, but not necessarily across VLANs. Traffic sourced within a VLAN (including broadcast and multicast traffic) remains within the VLAN it originates from. Traffic forwarded from one VLAN to another typically involves router (L3) processing. The forwarding table look up operation is keyed on {VLAN, MAC address} tuples.

But there are problems and limitations with L2 VLANs. VLANs are a pure L2 bridging construct and VLAN identifiers are carried along with data frames to allow each forwarding point to know what VLAN the frame belongs to. A VLAN today is defined as a 12 bit number, limiting the total number of VLANs to 4096 (though typically, this number is 4094 since 0 and 4095 are reserved). Due to the large number of tenants that a cloud provider might service, the 4094 VLAN limit is often inadequate. In addition, there is often a need for multiple VLANs per tenant, which exacerbates the issue.

In the case of IP networks, many routers provide a Virtual Routing and Forwarding (VRF) service. The same router operates multiple instances of forwarding tables, one for each tenant. Each forwarding table instance is populated separately via routing protocols, either running (conceptually) as separate instances for each VRF, or as a single instance-aware routing protocol that supports VRFs directly (e.g., [\[RFC4364\]](#)). Each VRF instance provides address and traffic

isolation. The forwarding table look up operation is keyed on {VRF, IP address} tuples.

VRF's are a pure routing construct and do not have end-to-end significance in the sense that the data plane carries a VRF indicator on an end-to-end basis. Instead, the VRF is derived at each hop using a combination of incoming interface and some information in the frame (e.g., local VLAN tag). Furthermore, the VRF model has typically assumed that a separate control plane governs the population of the forwarding table within that VRF. Thus, a traditional VRF model assumes multiple, independent control planes and has no specific tag within a data frame to identify the VRF of the frame.

There are number of VPN approaches that provide some of the desired semantics of virtual networks (e.g., [RFC4364](#)). But VPN approaches have traditionally been deployed across WANs and have not seen widespread deployment within enterprise data centers. They are not necessarily seen as supporting the characteristics outlined in [Section 2.7](#).

### [3.2](#). Benefits of Network Overlays

To address the problems described earlier, a network overlay model can be used.

The idea behind an overlay is quite straightforward. Each virtual network instance is implemented as an overlay. The original frame is encapsulated by the first hop network device. The encapsulation identifies the destination of the device that will perform the decapsulation before delivering the frame to the endpoint. The rest of the network forwards the frame based on the encapsulation header and can be oblivious to the payload that is carried inside. To avoid belaboring the point each time, the first hop network device can be a traditional switch or router or the virtual switch residing inside a hypervisor. Furthermore, the endpoint can be a VM or it can be a physical server. Some examples of network overlays are tunnels such as IP GRE [RFC2784](#), LISP [I-D.ietf-lisp](#) or TRILL [RFC6325](#).

With the overlay, a virtual network identifier (or VNID) can be carried as part of the overlay header so that every data frame explicitly identifies the specific virtual network the frame belongs to. Since both routed and bridged semantics can be supported by a virtual data center, the original frame carried within the overlay header can be an Ethernet frame complete with MAC addresses or just the IP packet.

The use of a large (e.g., 24-bit) VNID would allow 16 million

distinct virtual networks within a single data center, eliminating current VLAN size limitations. This VNID needs to be carried in the data plane along with the packet. Adding an overlay header provides a place to carry this VNID.

A key aspect of overlays is the decoupling of the "virtual" MAC and IP addresses used by VMs from the physical network infrastructure and the infrastructure IP addresses used by the data center. If a VM changes location, the switches at the edge of the overlay simply update their mapping tables to reflect the new location of the VM within the data center's infrastructure space. Because an overlay network is used, a VM can now be located anywhere in the data center that the overlay reaches without regards to traditional constraints implied by L2 properties such as VLAN numbering, or the span of an L2 broadcast domain scoped to a single pod or access switch.

Multi-tenancy is supported by isolating the traffic of one virtual network instance from traffic of another. Traffic from one virtual network instance cannot be delivered to another instance without (conceptually) exiting the instance and entering the other instance via an entity that has connectivity to both virtual network instances. Without the existence of this entity, tenant traffic remains isolated within each individual virtual network instance. External communications (from a VM within a virtual network instance to a machine outside of any virtual network instance, e.g. on the Internet) is handled by having an ingress switch forward traffic to an external router, where an egress switch decapsulates a tunneled packet and delivers it to the router for normal processing. This router is external to the overlay, and behaves much like existing external facing routers in data centers today.

Overlays are designed to allow a set of VMs to be placed within a single virtual network instance, whether that virtual network provides the bridged network or a routed network.

### [3.3.](#) Overlay Networking Work Areas

There are three specific and separate potential work areas needed to realize an overlay solution. The areas correspond to different

possible "on-the-wire" protocols, where distinct entities interact with each other.

One area of work concerns the address dissemination protocol an NVE uses to build and maintain the mapping tables it uses to deliver encapsulated frames to their proper destination. One approach is to build mapping tables entirely via learning (as is done in 802.1 networks). But to provide better scaling properties, a more sophisticated approach is needed, i.e., the use of a specialized

control plane protocol. While there are some advantages to using or leveraging an existing protocol for maintaining mapping tables, the fact that large numbers of NVE's will likely reside in hypervisors places constraints on the resources (cpu and memory) that can be dedicated to such functions. For example, routing protocols (e.g., IS-IS, BGP) may have scaling difficulties if implemented directly in all NVEs, based on both flooding and convergence time concerns. This suggests that use of a standard lookup protocol between NVEs and a smaller number of network nodes that implement the actual routing protocol (or the directory-based "oracle") is a more promising approach at larger scale.

From an architectural perspective, one can view the address mapping dissemination problem as having two distinct and separable components. The first component consists of a back-end "oracle" that is responsible for distributing and maintaining the mapping information for the entire overlay system. The second component consists of the on-the-wire protocols an NVE uses when interacting with the oracle.

The back-end oracle could provide high performance, high resiliency, failover, etc. and could be implemented in different ways. For example, one model uses a traditional, centralized "directory-based" database, using replicated instances for reliability and failover (e.g., LISP-XXX). A second model involves using and possibly extending an existing routing protocol (e.g., BGP, IS-IS, etc.). To support different architectural models, it is useful to have one standard protocol for the NVE-oracle interaction while allowing different protocols and architectural approaches for the oracle itself. Separating the two allows NVEs to interact with different types of oracles, i.e., either of the two architectural models described above. Having separate protocols also allows for a

simplified NVE that only interacts with the oracle for the mapping table entries it needs and allows the oracle (and its associated protocols) to evolve independently over time with minimal impact to the NVEs.

A third work area considers the attachment and detachment of VMs (or Tenant End Systems [[I-D.lasserre-nvo3-framework](#)] more generally) from a specific virtual network instance. When a VM attaches, the Network Virtualization Edge (NVE) [[I-D.lasserre-nvo3-framework](#)] associates the VM with a specific overlay for the purposes of tunneling traffic sourced from or destined to the VM. When a VM disconnects, it is removed from the overlay and the NVE effectively terminates any tunnels associated with the VM. To achieve this functionality, a standardized interaction between the NVE and hypervisor may be needed, for example in the case where the NVE resides on a separate device from the VM.

In summary, there are three areas of potential work. The first area concerns the oracle itself and any on-the-wire protocols it needs. A second area concerns the interaction between the oracle and NVEs. The third work area concerns protocols associated with attaching and detaching a VM from a particular virtual network instance. The latter two items are the priority work areas and can be done largely independent of any oracle-related work.

## [4.](#) Related Work

### [4.1.](#) IEEE 802.1ah - Shortest Path Bridging

Shortest Path Bridging (SPB) is an IS-IS based overlay for L2 Ethernet. SPB supports multi-pathing and addresses a number of shortcomings in the original Ethernet Spanning Tree Protocol. SPB-M uses IEEE 802.1ah MAC-in-MAC encapsulation and supports a 24-bit I-SID, which can be used to identify virtual network instances. SPB is entirely L2 based, extending the L2 Ethernet bridging model.

### [4.2.](#) ARMD

ARMD is chartered to look at data center scaling issues with a focus on address resolution. ARMD is currently chartered to develop a problem statement and is not currently developing solutions. While

an overlay-based approach may address some of the "pain points" that have been raised in ARMD (e.g., better support for multi-tenancy), an overlay approach may also push some of the L2 scaling concerns (e.g., excessive flooding) to the IP level (flooding via IP multicast). Analysis will be needed to understand the scaling trade offs of an overlay based approach compared with existing approaches. On the other hand, existing IP-based approaches such as proxy ARP may help mitigate some concerns.

#### [4.3.](#) TRILL

TRILL is an L2-based approach aimed at improving deficiencies and limitations with current Ethernet networks and STP in particular. Although it differs from Shortest Path Bridging in many architectural and implementation details, it is similar in that it provides an L2-based service to end systems. TRILL as defined today, supports only the standard (and limited) 12-bit VLAN model. Approaches to extend TRILL to support more than 4094 VLANs are currently under investigation [[I-D.ietf-trill-fine-labeling](#)]

#### [4.4.](#) L2VPNs

The IETF has specified a number of approaches for connecting L2 domains together as part of the L2VPN Working Group. That group, however has historically been focused on Provider-provisioned L2 VPNs, where the service provider participates in management and provisioning of the VPN. In addition, much of the target environment for such deployments involves carrying L2 traffic over WANs. Overlay approaches are intended be used within data centers where the overlay network is managed by the data center operator, rather than by an outside party. While overlays can run across the Internet as well, they will extend well into the data center itself (e.g., up to and including hypervisors) and include large numbers of machines within the data center itself.

Other L2VPN approaches, such as L2TP [[RFC2661](#)] require significant tunnel state at the encapsulating and decapsulating end points. Overlays require less tunnel state than other approaches, which is

important to allow overlays to scale to hundreds of thousands of end points. It is assumed that smaller switches (i.e., virtual switches in hypervisors or the physical switches to which VMs connect) will be part of the overlay network and be responsible for encapsulating and decapsulating packets.

#### [4.5.](#) Proxy Mobile IP

Proxy Mobile IP [[RFC5213](#)] [[RFC5844](#)] makes use of the GRE Key Field [[RFC5845](#)] [[RFC6245](#)], but not in a way that supports multi-tenancy.

#### [4.6.](#) LISP

LISP[I-D.ietf-lisp] essentially provides an IP over IP overlay where the internal addresses are end station Identifiers and the outer IP addresses represent the location of the end station within the core IP network topology. The LISP overlay header uses a 24 bit Instance ID used to support overlapping inner IP addresses.

#### [4.7.](#) Individual Submissions

Many individual submissions also look to addressing some or all of the issues addressed in this draft. Examples of such drafts are VXLAN [[I-D.mahalingam-dutt-dcops-vxlan](#)], NVGRE [[I-D.sridharan-virtualization-nvgre](#)] and Virtual Machine Mobility in L3 networks[I-D.wkumari-dcops-l3-vm-mobility].

### [5.](#) Further Work

It is believed that overlay-based approaches may be able to reduce the overall amount of flooding and other multicast and broadcast related traffic (e.g, ARP and ND) currently experienced within current data centers with a large flat L2 network. Further analysis is needed to characterize expected improvements.

### [6.](#) Summary

This document has argued that network virtualization using L3 overlays addresses a number of issues being faced as data centers scale in size. In addition, careful consideration of a number of issues would lead to the development of interoperable implementation of virtualization overlays.

Three potential work were identified. The first involves the interaction that take place when a VM attaches or detaches from an overlay. A second involves the protocol an NVE would use to communicate with a backend "oracle" to learn and disseminate mapping information about the VMs the NVE communicates with. The third potential work area involves the backend oracle itself, i.e., how it provides failover and how it interacts with oracles in other domains.

## 7. Acknowledgments

Helpful comments and improvements to this document have come from Ariel Hendel, Vinit Jain, and Benson Schliesser.

## 8. IANA Considerations

This memo includes no request to IANA.

## 9. Security Considerations

TBD

## 10. Informative References

[I-D.ietf-lisp]

Farinacci, D., Fuller, V., Meyer, D., and D. Lewis,  
"Locator/ID Separation Protocol (LISP)",  
[draft-ietf-lisp-23](#) (work in progress), May 2012.

Narten, et al.

Expires December 17, 2012

[Page 14]

---

Internet-Draft

Overlays for Network Virtualization

June 2012

[I-D.ietf-trill-fine-labeling]

Eastlake, D., Zhang, M., Agarwal, P., Perlman, R., and D.  
Dutt, "TRILL: Fine-Grained Labeling",  
[draft-ietf-trill-fine-labeling-01](#) (work in progress),



June 2012.

[I-D.kreeger-nvo3-overlay-cp]

Black, D., Dutt, D., Kreeger, L., Sridhavan, M., and T. Narten, "Network Virtualization Overlay Control Protocol Requirements", [draft-kreeger-nvo3-overlay-cp-00](#) (work in progress), January 2012.

[I-D.lasserre-nvo3-framework]

Lasserre, M., Balus, F., Morin, T., Bitar, N., Rekhter, Y., and Y. Ikejiri, "Framework for DC Network Virtualization", [draft-lasserre-nvo3-framework-01](#) (work in progress), March 2012.

[I-D.mahalingam-dutt-dcops-vxlan]

Sridhar, T., Bursell, M., Kreeger, L., Dutt, D., Wright, C., Mahalingam, M., Duda, K., and P. Agarwal, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", [draft-mahalingam-dutt-dcops-vxlan-01](#) (work in progress), February 2012.

[I-D.sridharan-virtualization-nvgre]

Sridhavan, M., Duda, K., Ganga, I., Greenberg, A., Lin, G., Pearson, M., Thaler, P., Tumuluri, C., and Y. Wang, "NVGRE: Network Virtualization using Generic Routing Encapsulation", [draft-sridharan-virtualization-nvgre-00](#) (work in progress), September 2011.

[I-D.wkumari-dcops-l3-vm-mobility]

Kumari, W. and J. Halpern, "Virtual Machine mobility in L3 Networks.", [draft-wkumari-dcops-l3-vm-mobility-00](#) (work in progress), August 2011.

[RFC2661] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"", [RFC 2661](#), August 1999.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", [RFC 2784](#), March 2000.

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

- [RFC5213] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", [RFC 5213](#), August 2008.
- [RFC5844] Wakikawa, R. and S. Gundavelli, "IPv4 Support for Proxy Mobile IPv6", [RFC 5844](#), May 2010.
- [RFC5845] Muhanna, A., Khalil, M., Gundavelli, S., and K. Leung, "Generic Routing Encapsulation (GRE) Key Option for Proxy Mobile IPv6", [RFC 5845](#), June 2010.
- [RFC6245] Yegani, P., Leung, K., Lior, A., Chowdhury, K., and J. Navali, "Generic Routing Encapsulation (GRE) Key Extension for Mobile IPv4", [RFC 6245](#), May 2011.
- [RFC6325] Perlman, R., Eastlake, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", [RFC 6325](#), July 2011.

## [Appendix A](#). Change Log

### [A.1](#). Changes from -01

1. Removed [Section 4.2](#) (Standardization Issues) and [Section 5](#) (Control Plane) as those are more appropriately covered in and overlap with material in [[I-D.lasserre-nvo3-framework](#)] and [[I-D.kreeger-nvo3-overlay-cp](#)].
2. Expanded introduction and better explained terms such as tenant and virtual network instance. These had been covered in a section that has since been removed.
3. Added [Section 3.3](#) "Overlay Networking Work Areas" to better articulate the three separable work components (or "on-the-wire protocols") where work is needed.
4. Added section on Shortest Path Bridging in Related Work section.
5. Revised some of the terminology to be consistent with [[I-D.lasserre-nvo3-framework](#)] and [[I-D.kreeger-nvo3-overlay-cp](#)].

Internet-Draft

Overlays for Network Virtualization

June 2012

#### Authors' Addresses

Thomas Narten (editor)  
IBM

Email: [narten@us.ibm.com](mailto:narten@us.ibm.com)

Murari Sridharan  
Microsoft

Email: [muraris@microsoft.com](mailto:muraris@microsoft.com)

Dinesh Dutt

Email: [ddutt.ietf@hobbesdutt.com](mailto:ddutt.ietf@hobbesdutt.com)

David Black  
EMC

Email: [david.black@emc.com](mailto:david.black@emc.com)

Lawrence Kreeger  
Cisco

Email: [kreeger@cisco.com](mailto:kreeger@cisco.com)

Narten, et al.

Expires December 17, 2012

[Page 17]