Creating Overlay Networks Using Intel® Ethernet Converged Network Adapters

Technical Brief

Networking Division (ND)

August 2013

Revision 1.0

LEGAL

Revision History

Revision	Date	Comments
1.0	August 2013	Initial release

Contents

1.0 Introduction

In recent years, Cloud computing has emerged as one of the key usage models for data centers. Cloud computing framework enables data center providers to abstract physical resources from users that enable sharing of physical resources. Intel® Virtualization Technology is one of the key building blocks for Cloud computing. Data center providers are sharing processor, memory and I/O resources using Intel® Virtualization technology, along with software from Microsoft*, VMware*, Citrix* and Linux* distributors, which provides a mechanism to provision and manage virtualized hardware. The Intel® Ethernet Converged Network Adapter X520 and Intel® Ethernet Converged Network Adapter X540 families support industry-leading features that are accelerating the performance and implementation of 10 GbE in Cloud computing environments.

As Cloud computing matures, the need to virtualize networking resources has become vital. Cloud service providers expect network security and traffic segmentation in multi-tenant Cloud environments. Software Defined Networks (SDN) and Overlay Networks are industry standard techniques designed to achieve Network Virtualization. Network Overlays such as Generic Routing Encapsulation (GRE) and Virtual eXtensible Local Area Network (VXLAN) achieve Network Virtualization by overlaying layer 2 networks over physical layer 3 networks. These techniques enable network scalability and efficient use of current network infrastructure. Cloud computing frameworks such as EC2 from Amazon*, VMware's Cloud Infrastructure, and Linux based Open Stack use GRE or VXLAN to virtualize, provision and manage network resources.

This document shows how to create Overlay Networks using GRE and VXLAN tunnels on Intel X520 or X540 Ethernet Converged Networking Adapters in a Linux environment.

1.1 Hardware Requirements

- An Intel® Ethernet Converged Network Adapter X520 or X540
- A server platform that supports Intel® Virtualization Technology
- A server platform with an available PCI Express* (PCIe*): X8 5.0 GT/s (Gen2) slot
- 10 GbE Switch

1.2 Software Requirements

- Fedora 19 or Ubuntu 13.04 Linux distribution
- Open vSwitch version 1.10.0 (http://openvswitch.org/release/openvswitch-1.10.0.tar.gz)

2.0 Installation and Configuration

2.1 Enabling the Host Server and OS Installation

1. Install the Intel® Ethernet CNA X520 or X540 server adapter in an available PCIe x8 slot. Ensure that the x8 slot is electrically connected as an x8—some slots are physically x8 but electrically support only x4. Verify this with your server manufacturer or system documentation.
2. Power up the server.
3. Enter the server's BIOS setup and make sure the virtualization technology feature is enabled on the server.

Install either the Fedora 19 or Ubuntu 13.04 Linux distribution.

Software	Fedora 19	Ubuntu 13.04
Development Tools	**`~ # yum groupinstall virtualization`**	`~ # sudo apt-get install build-essentials`
Virtualization	**`~ # yum groupinstall 'Development Tools'`**	`~ # sudo apt-get install qemu-kvm libvirt-bin bridge-utils` `~ # sudo apt-get install ubuntu-vm-builder`

2.2 Overlay Networks using Open vSwitch

Download, configure, compile and install Open vSwitch. Follow the instructions provided within the Open vSwitch package.

2.2.1 GRE Tunnel

Figure 1 shows a typical GRE tunnel setup. VM-A and VM-B are part of a tenant network for ACME Corporation. A Cloud Service provider has assigned the unique Tenant Network Identifier (TNI) 5457 to ACME Corporation's network. A Unique TNI is required for each tenant utilizing the physical network to maintain network security and the tenants' network traffic segregation. Network traffic flow from VM-A on "Host1" to VM-B on "Host2" follows:

1. Traffic from VM-A flows via bridge "br2" using egress internal port "gre0."

2. A GRE header is added to the outgoing Ethernet frames and handed to bridge “br7” via ingress internal port “tep7.”
3. Ethernet frames with GRE header are now placed on the wire using eth7 device destined for VM-B on “Host2.”

The process is reversed on “Host2” and the Ethernet frame is handed to VM-B.

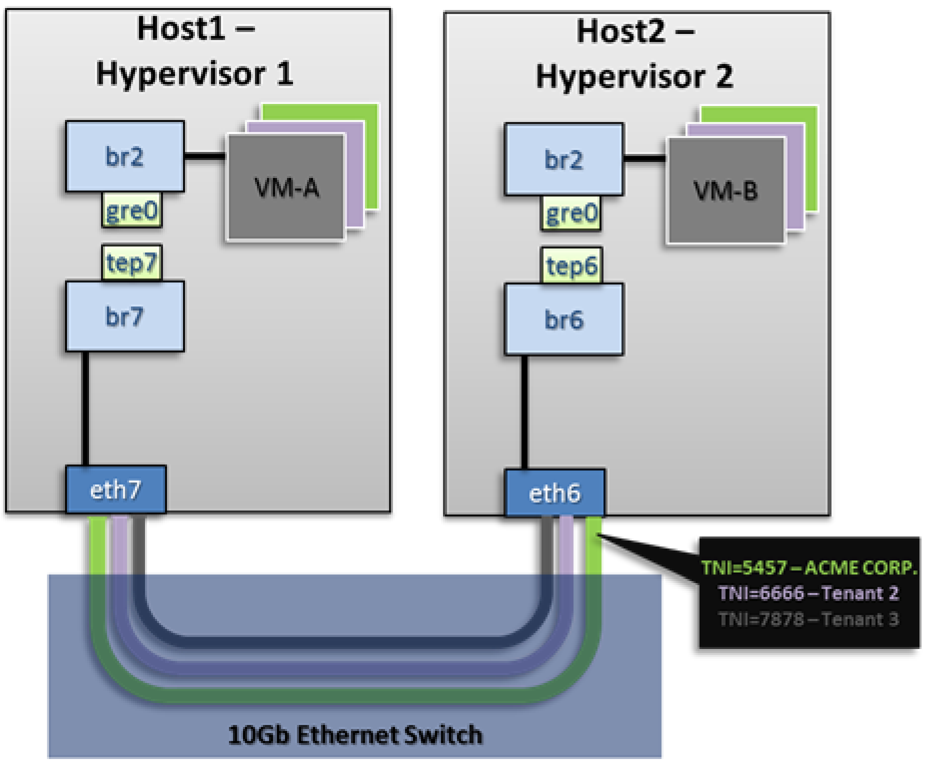

Figure 1. GRE Tunnel

Providing network access to additional tenants is as simple as creating additional OVS Bridges for each tenant using a unique TNI and creating a corresponding Tunnel Endpoint (TEP) port.

Configuration for Host1 is shown below.

```
# ovs-vsctl add-br br7
# ovs-vsctl add-port br7 eth7
# ovs-vsctl add-port br7 tep7 -- set interface tep7 type=internal
# ip addr add 192.168.68.20/24 dev tep7
# ip link set tep7 up
# ovs-vsctl add-br br2
# ovs-vsctl add-port br2 gre0 -- set interface gre0 type=gre
options:remote_ip=192.168.68.10 options:key=5457
#
```

Figure 2. Host 1 Configuration

Configuration for Host2 follows.

```
# ovs-vsctl add-br br6
# ovs-vsctl add-port br6 eth6
# ovs-vsctl add-port br6 tep6 -- set interface tep6 type=internal
# ip addr add 192.168.68.10/24 dev tep6
# ip link set tep6 up
# ovs-vsctl add-br br2
# ovs-vsctl add-port br2 gre0 -- set interface gre0 type=gre
options:remote_ip=192.168.68.20 options:key=5457
#
```

Figure 3. Host 2 Configuration

Figure 4 shows the various elements of an Ethernet frame with GRE header.

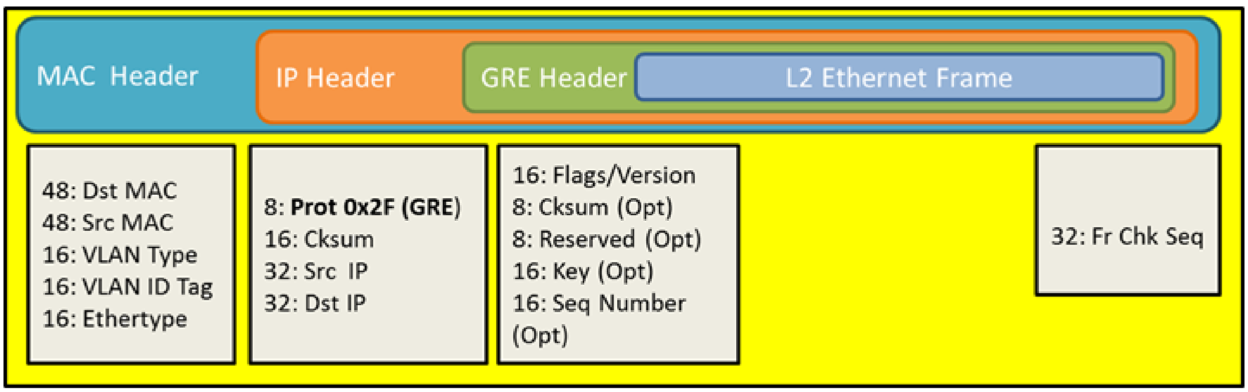

Figure 4. Ethernet Frame Elements

Figure 5 shows the corresponding network packet.

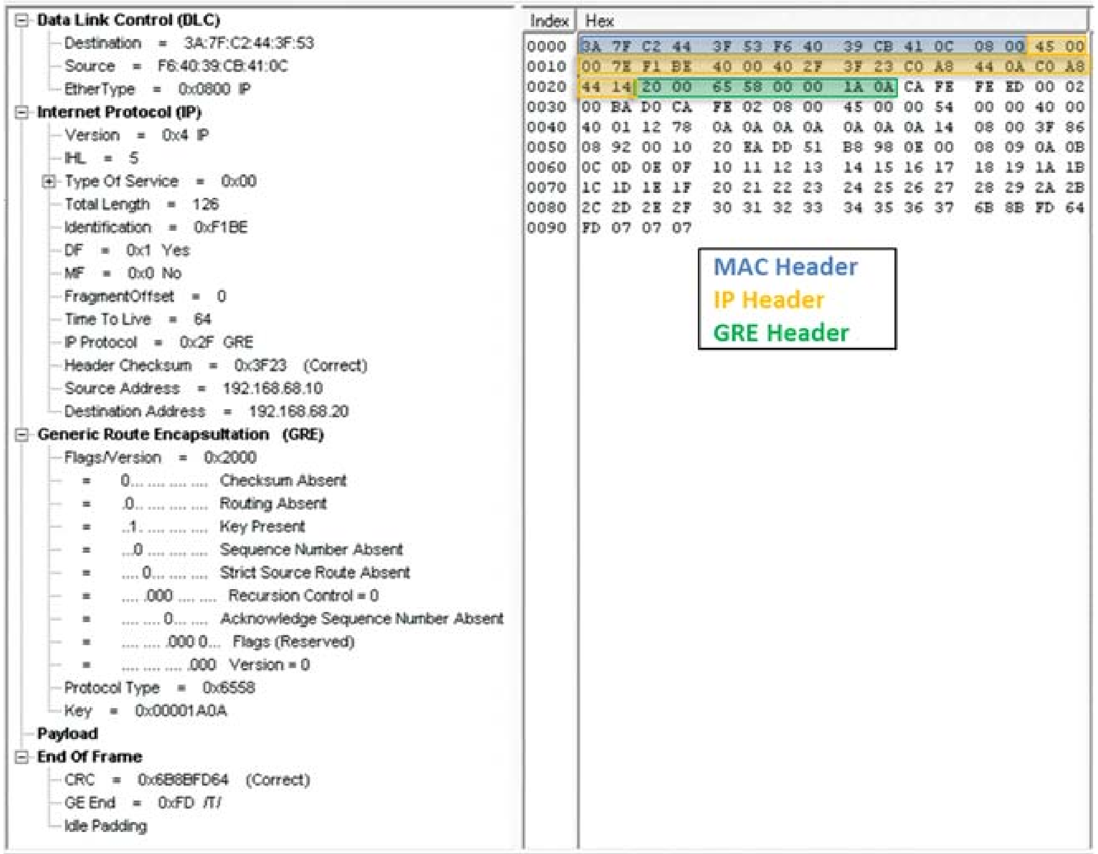

Figure 5. Corresponding Network Packet

2.2.2 VXLAN Tunnel

Figure 6 shows a typical VXLAN tunnel setup. VM-A and VM-B are part of a tenant network for ABC Corporation. Their Cloud Service provider has assigned a unique VNI 5000 to ABC Corporation's network. A unique VNI is required for each tenant utilizing the physical network to maintain network security and tenant network traffic segregation. Network traffic flow from VM-A on “Host1” to VM-B on “Host2” follows:

1. Traffic from VM-A flows via bridge “br2” using egress internal port “vx9.”
2. A VXLAN header is added to the outgoing Ethernet frames and handed to bridge “br2” via ingress internal port “tep2.”
3. Ethernet frames with the VXLAN header are now placed on the wire using “eth2” device destined for VM-B on “Host2.”

4. The process is reversed on "Host2" and the Ethernet frame is handed to VM-B as shown in Figure 6.

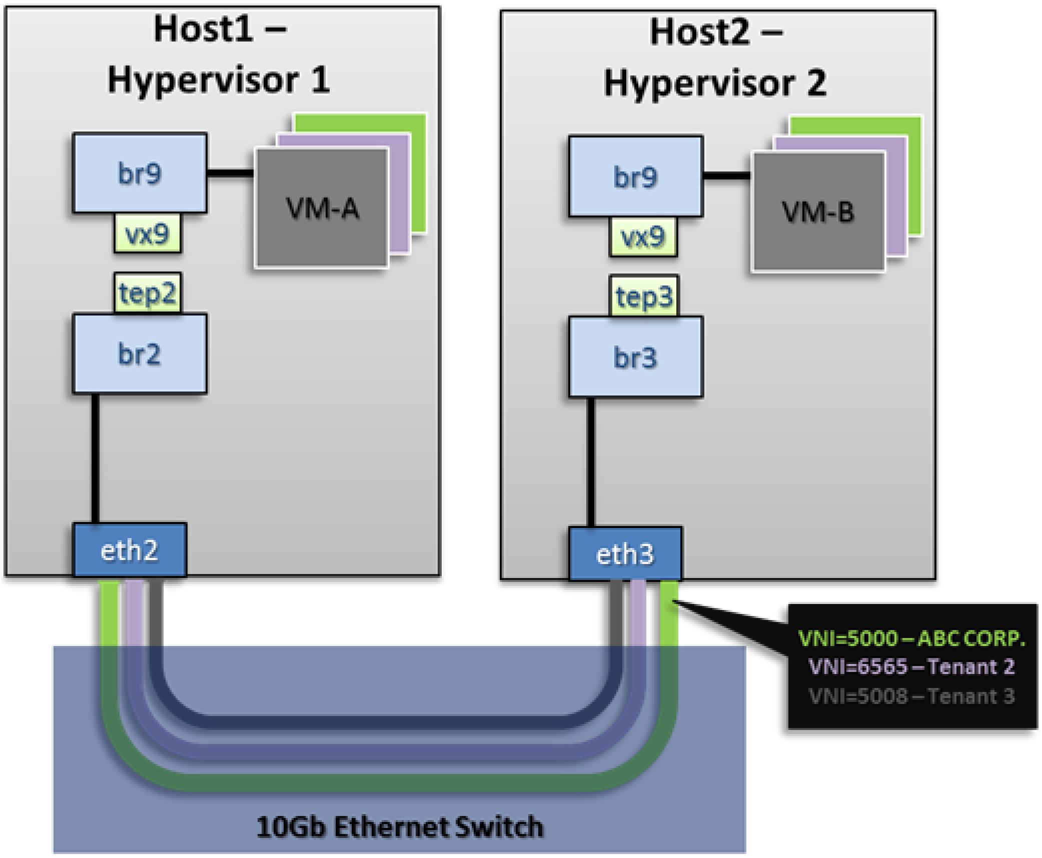

Figure 6. Typical VXLAN Tunnel Setup

Providing network access to additional tenants is as simple as creating an additional OVS Bridge for each by using a unique VNI and creating a corresponding TEP port.

The "key" parameter in the following code corresponds to VXLAN VNI. A VNI value of 5000 is used in the example below.

The "dst_port" parameter is optional and is used to specify a UDP port number for the TEP port's VXLAN traffic.

Note that the UDP port number is implementation specific and should be consistent for the network. Different vendors may choose to use different UDP Ports for VXLAN traffic. For example, VMware* products use UDP port 4789 for VXLAN traffic as their default. To ensure successful VXLAN communication, please be sure to consistently configure the VXLAN network to use the correct UDP Port for your implementation.

The following Figure 7 shows VXLAN Tunnel setup commands for Host1.

```
# ovs-vsctl add-br br2
# ovs-vsctl add-port br2 eth2
# ovs-vsctl add-port br2 tep2 -- set interface tep2 type=internal
# ip link addr add 172.16.68.20/24 dev tep2
# ip link set tep2 up
# ovs-vsctl add-br br9
# ovs-vsctl add-port br9 vx9 -- set interface vx9 type=vxlan
options:remote_ip=172.16.68.10 options:key=5000 options:dst_port=9999
#
```

Figure 7. Host 1 Configuration

Below, Figure 8 shows VXLAN Tunnel setup commands for Host2

```
# ovs-vsctl add-br br3
# ovs-vsctl add-port br3 eth3
# ovs-vsctl add-port br3 tep3 -- set interface tep3 type=internal
# ip link addr add 172.16.68.10/24 dev tep3
# ip link set tep3 up
# ovs-vsctl add-br br9
# ovs-vsctl add-port br9 vx9 -- set interface vx9 type=vxlan
options:remote_ip=172.16.68.20 options:key=5000 options:dst_port=9999
#
```

Figure 8. Host 2 Configuration

Figure 9 shows various elements of an Ethernet frame with VXLAN header.

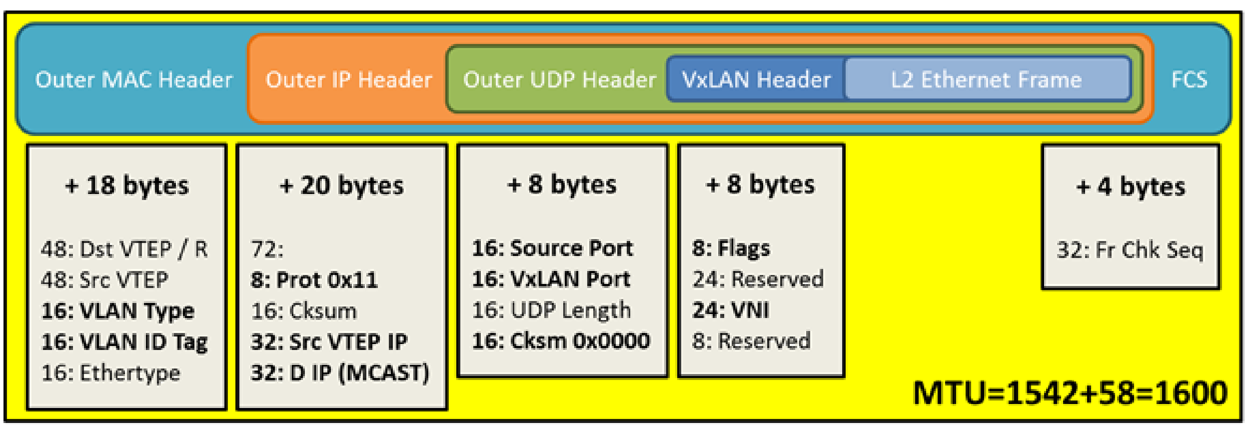

Figure 9. Elements of an Ethernet Frame with VXLAN Header

Note: The VXLAN header adds an additional 58 bytes to the standard L2 Ethernet frame. It is recommended that the MTU size be increased to 1600 on all network switches.

Ethernet frames sent to and from a VXLAN connected device are encapsulated in UDP packets. Linux Drivers for the Intel Ethernet CNA X520 and X540 support Receive Side Scaling (RSS) for UDP. Enabling RSS for Intel Ethernet CNAs may benefit VXLAN network performance.

Figure 10 shows the corresponding network packet.

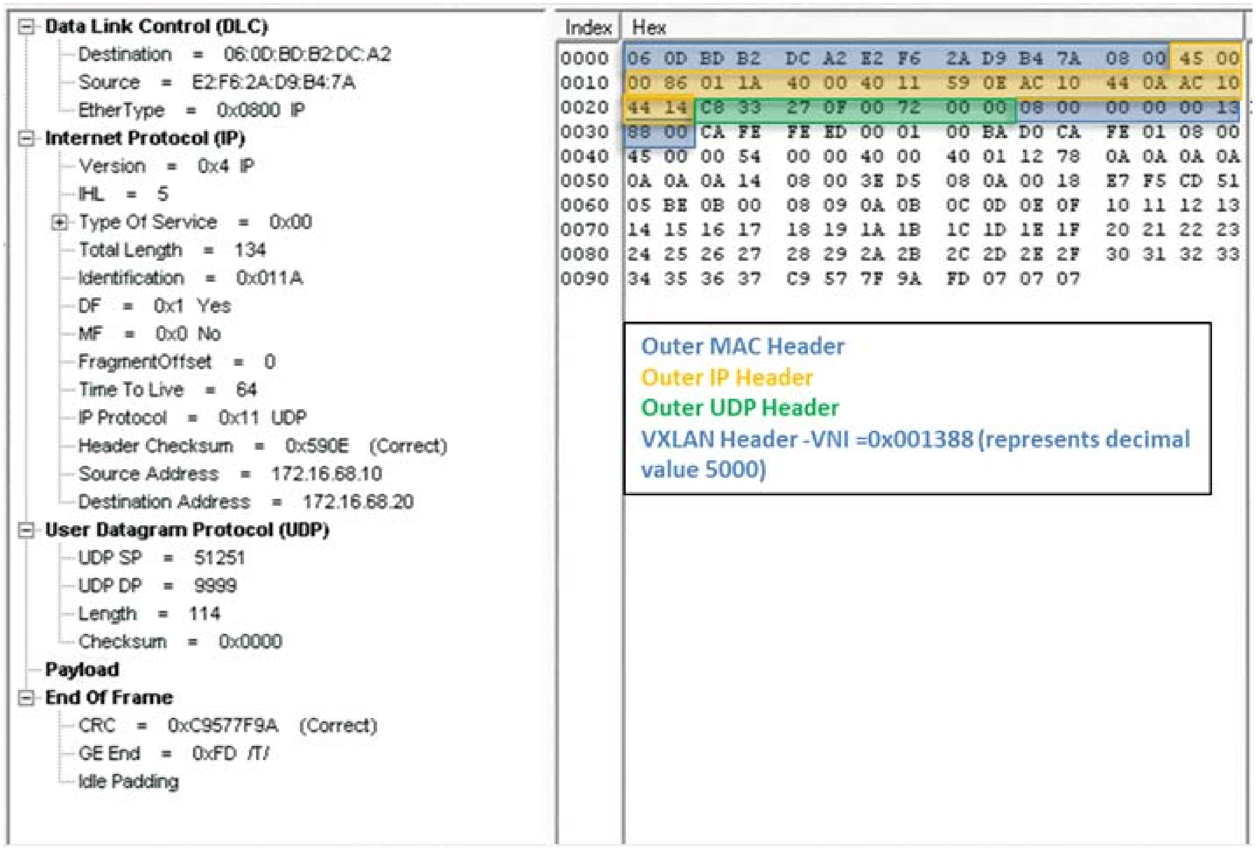

Figure 10. Network Packet Detail

2.3 Network Connectivity to Guest Virtual Machines

Most Linux Distributions ship with a Kernel Based Virtual Machine (KVM) solution, command line KVM management tools such as "virsh," and a graphical user interface (GUI) based "virt-manager." The Cloud Infrastructure provided typically creates Virtual Machines per tenants' specifications. These VMs communicate over a segregated network using Network Overlays such as GRE and/or VXLAN. Cloud Infrastructure Service Providers may choose to implement either GRE, VXLAN, or both, for providing network connectivity to their tenants.

The following sections show how to configure VMs to communicate over GRE and VXLAN. Unfortunately, the GUI based VM management tool virt-manager doesn't allow VM network interface assignment to Open vSwitch at the time this document was authored. Virsh is required to edit a VM's configuration. Use the "`virsh edit virtual-`*`machine-name`*" command on a Linux console to edit the VM settings.

Add the following lines to the device section of the file:

```
# virsh edit vm1
#
#
   <interface type='bridge'>
     <mac address='ca:fe:fe:ed:00:01'/>
     <source bridge='br2'/>
     <virtualport type='openvswitch'>
     </virtualport>
     <model type='virtio'/>
   </interface>
```

In the previous example, the VM is configured to use the GRE network configuration from Section 2.2.1. Each VM must be configured for a unique MAC address.

3.0 Summary

Intel's Best-in-Class 10 GbE solutions are now available with Network Virtualization capabilities. Customers get world-class Ethernet support along with Network Virtualization support in mainstream Linux distributions in a single adapter.

4.0 Customer Support

Intel® Customer Support Services offer a broad selection of programs including phone support and warranty service. For more information, contact us at:

support.intel.com/support/go/network/ adapter/home.htm

Note: Service and availability may vary by country.

5.0 For Product Information

To speak to a customer service representative regarding Intel products, please call 1-800-538-3373 (U.S. and Canada) or visit support.intel.com/support/go/network/contact.htm for the telephone number in your area.