



---

NUMBER: AB17001  
June 2017

*MDS Orbit Series*

---

GE MDS, LLC. 175 Science Parkway, Rochester, NY 14620 USA  
Phone +1 (585) 242-9600, FAX +1 (585) 242-9620 Web: www.gemds.com

---

## Orbit DMVPN Tunneling

### Dynamic Multipoint Virtual Private Network



#### Introduction

---

This document was created to assist customers use IPsec VPN or DMVPN configurations for access points and remotes. You will be able to use this document to provision unit easily and quickly.

Please note: This document assumes basic knowledge of the Orbit Platform. GE suggests reviewing the YouTube training videos from the link below:

<https://www.youtube.com/watch?v=OcWSG4xERcY&list=PLrbxqFUR561iSD9i6MHBtA6Z692sYr-rq>

## Scope

---

This bulletin is intended for system engineers and end users who are familiar with the Orbit command line interface (CLI) and interested in setting up a multiple IPsec VPN tunnels using Dynamic Multipoint VPN on Orbit devices. Please refer to the MDS Orbit MCR/ECR Technical Manual (05-6632A01) for details on how to access Orbit CLI or Web UI.

## Firmware Compatibility

---

This bulletin is applicable to Orbit devices running firmware version 6.1.2 or greater.

## Terms

---

CLI Command Line Interface  
VPN Virtual private Network  
DMVPN Dynamic Multipoint VPN  
GRE Generic Routing Encapsulation  
NHRP Next Hop Resolution Protocol  
IPSEC Internet Protocol Security

## What is DMVPN?

---

DMVPN (Dynamic Multipoint VPN) is a routing technique we can use to build a VPN network with multiple sites without having to statically configure all devices. It's a "hub and spoke" network where the spokes will be able to communicate with each other directly without having to go through the hub. Encryption is supported through IPsec which makes DMVPN a popular choice for connecting different sites using regular Internet connections. It's a great backup or alternative to private networks like MPLS VPN.

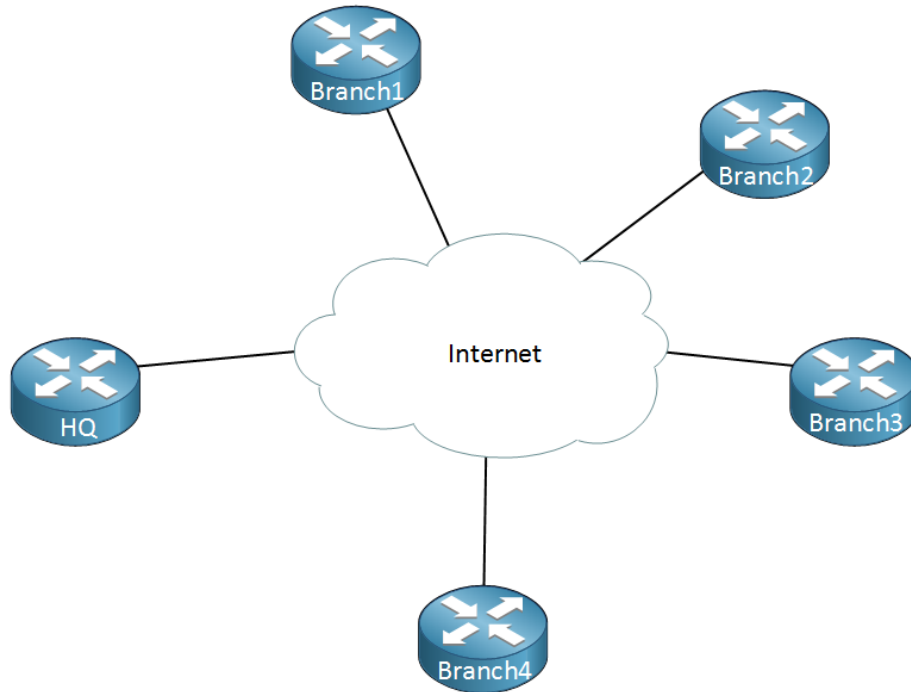
There are four pieces to the DMVPN puzzle:

1. Multipoint GRE (mGRE)
2. NHRP (Next Hop Resolution Protocol)
3. Routing (RIP, OSPF, BGP)
4. IPsec (not required but recommend)

## Multipoint GRE

---

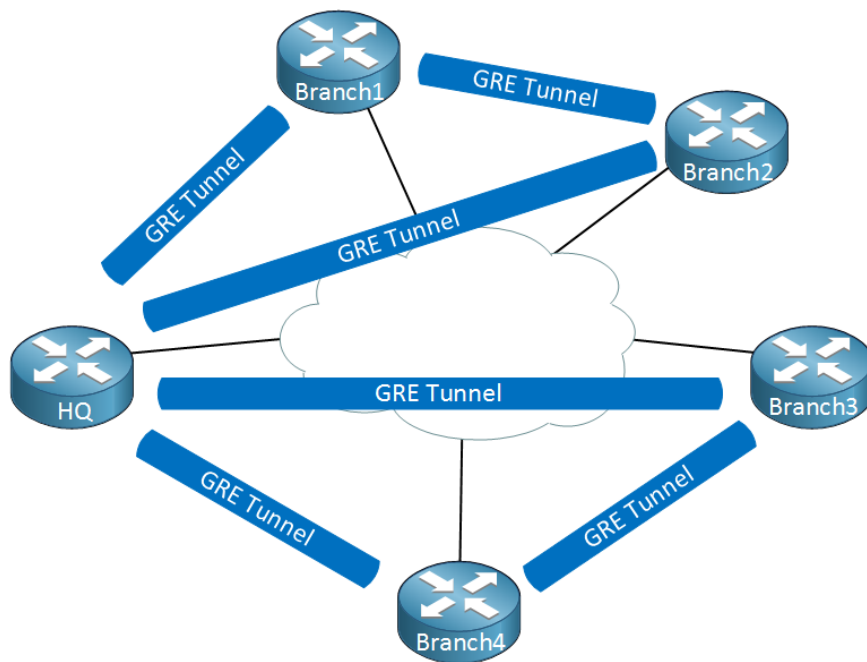
Our “regular” GRE tunnels are point-to-point and don’t scale well. For example, let’s say we have a company network with some sites that we want to connect to each other using regular Internet connections:



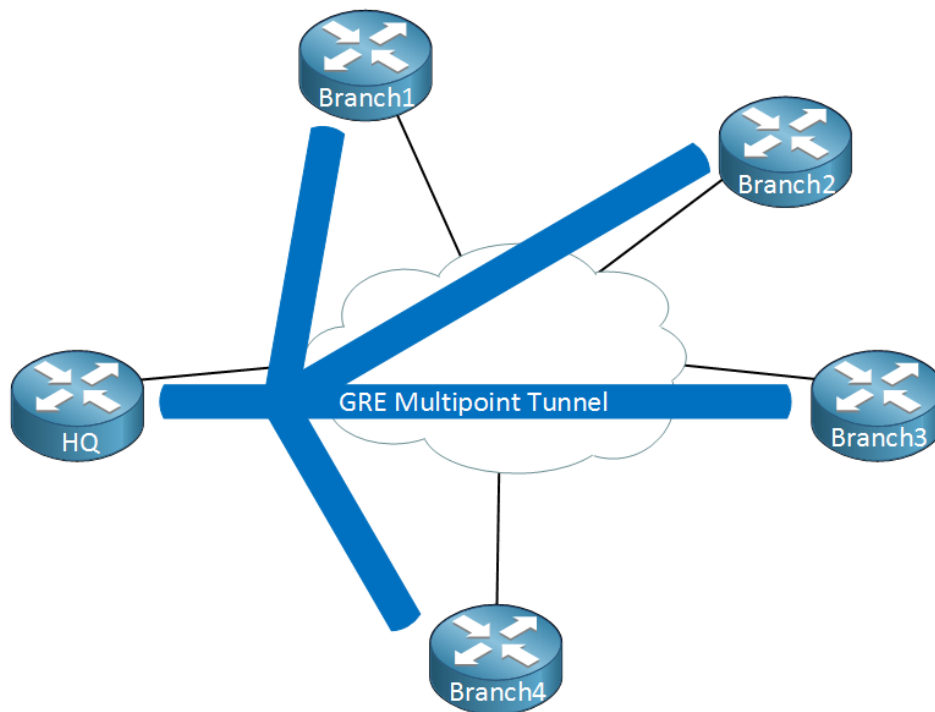
Above we have one router that represents the HQ and there are four branch offices. Let’s say that we have the following requirements:

- Each branch office has to be connected to the HQ.
- Traffic between Branch 1 and Branch 2 has to be tunneled directly.
- Traffic between Branch 3 and Branch 4 has to be tunneled directly.

To accomplish this we will have to configure a bunch of GRE tunnels which will look like this:

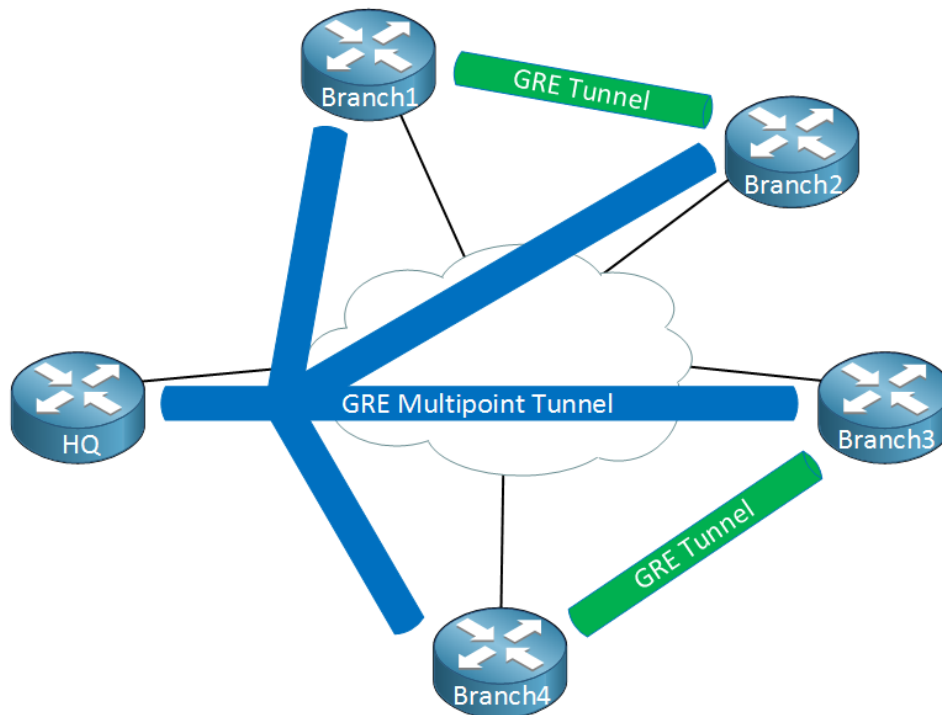


Thing will get messy quickly...we must create multiple tunnel interfaces, set the source/destination IP addresses etc. It will work but it's not a very scalable solution. Multipoint GRE, as the name implies allows us to have **multiple destinations**. When we use them, our picture could look like this:



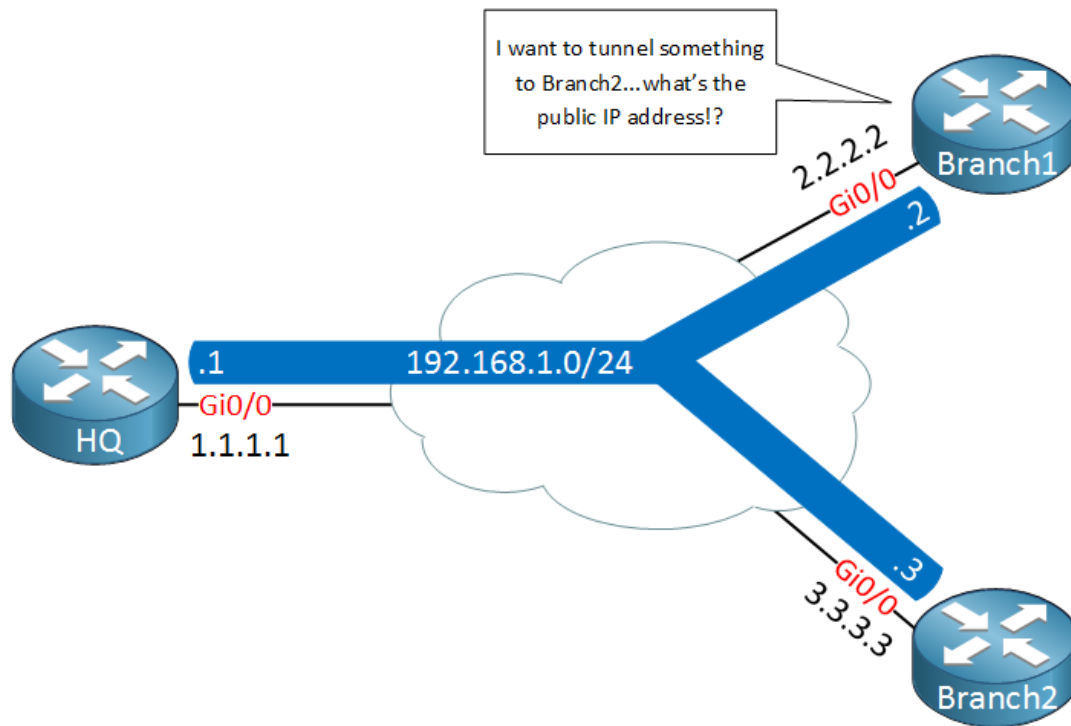
When we use GRE Multipoint, there will be only **one tunnel interface** on each router. The HQ for example has one tunnel with each branch office as its destination. Now you might be wondering, what about the requirement where branch office 1/2 and branch office 3/4 have a direct tunnel?

Right now we have a hub and spoke topology. The cool thing about DMVPN is that we use multipoint GRE so we can have multiple destinations. When we need to tunnel something between branch office 1/2 or 3/4, we automatically “build” new tunnels:



When there is traffic between the branch offices, we can tunnel it directly instead of sending it through the HQ router. This sounds pretty cool but it introduces some problems...

When we configure point-to-point GRE tunnels we have to configure a source and destination IP address that are used to build the GRE tunnel. When two branch routers want to tunnel some traffic, how do they know what IP addresses to use? Let me show you what I’m talking about:



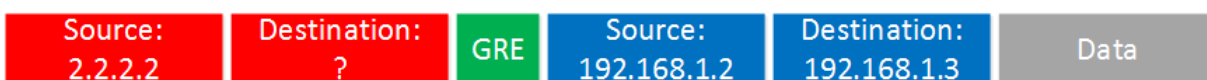
Above we have our HQ and two branch routers, branch1 and branch2. Each router is connected to the Internet and has a public IP address:

- HQ: 1.1.1.1
- Branch1: 2.2.2.2
- Branch2: 3.3.3.3

On the GRE multipoint tunnel interface we use a single subnet with the following private IP addresses:

- HQ: 192.168.1.1
- Branch1: 192.168.1.2
- Branch2: 192.168.1.3

Let's say that we want to send a ping from branch1's tunnel interface to the tunnel interface of branch2. Here's what the GRE encapsulated IP packet will look like:



The “inner” source and destination IP addresses are known to use, these are the IP address of the tunnel interfaces. We encapsulate this IP packet, put a GRE header in front of it and then we have to fill in the “outer” source and destination IP addresses so that this packet can be routed on the Internet. The branch1 router knows it’s own public IP address but it has no clue what the public IP address of branch2 is...

To fix this problem, we need some help from another protocol...

### **NHRP (Next Hop Resolution Protocol)**

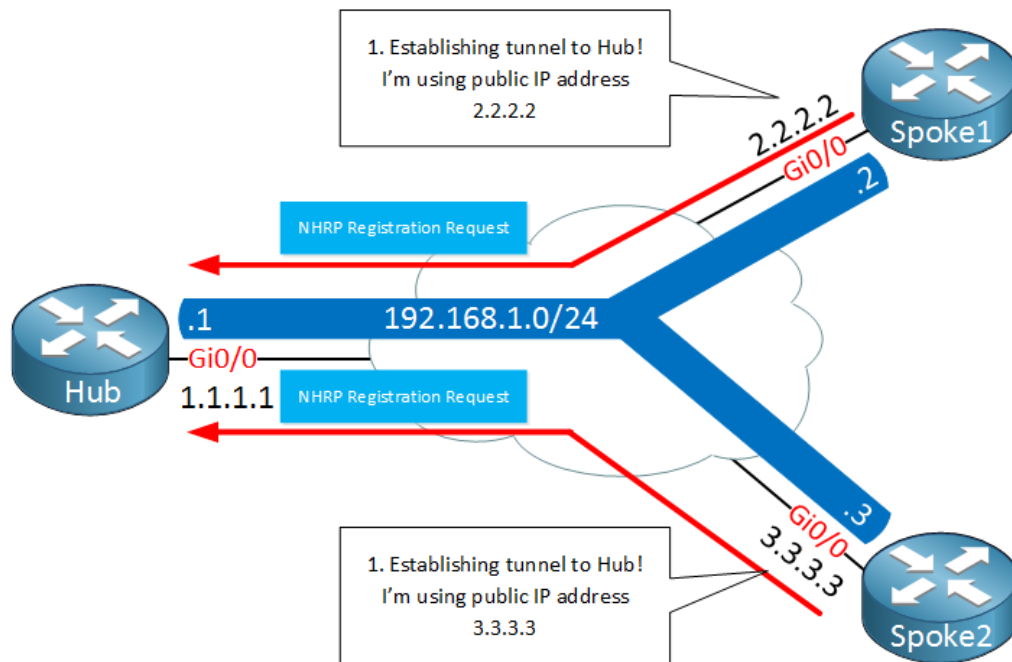
---

We need something that helps our branch1 router figure out what the public IP address is of the branch2 router, we do this with a protocol called **NHRP (Next Hop Resolution Protocol)**. Here’s an explanation of how NHRP works:

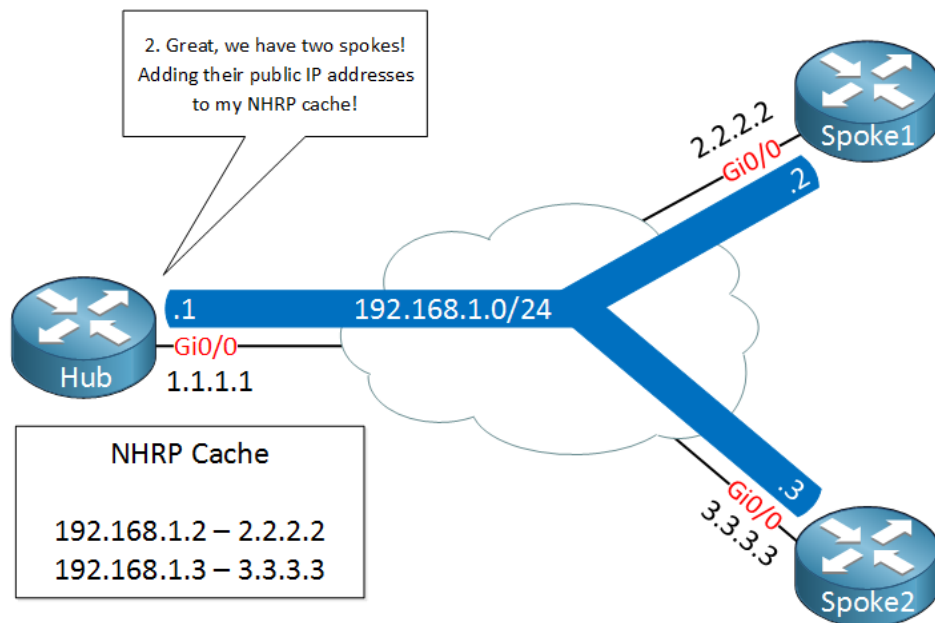
- One router will be the **NHRP server**.
- All other routers will be **NHRP clients**.
- NHRP clients register themselves with the NHRP server and **report their public IP address**.
- The NHRP server **keeps track of all public IP addresses in its cache**.
- When one router wants to tunnel something to another router, it will **request the NHRP server** for the public IP address of the other router.

Since NHRP uses this *server and clients* model, it makes sense to use a hub and spoke topology for multipoint GRE. Our hub router will be the NHRP server and all other routers will be the spokes.

Here’s an an illustration of how NHRP works with multipoint GRE:

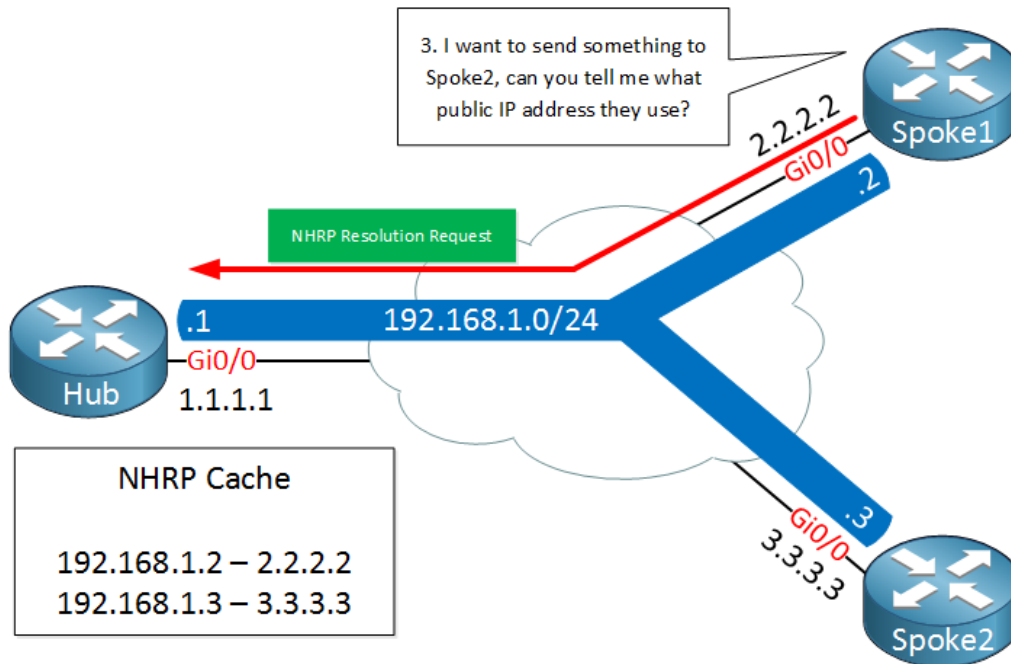


Above we have two spoke routers (NHRP clients) which establish a tunnel to the hub router. Later once we look at the configurations you will see that the destination IP address of the hub router will be **statically configured** on the spoke routers. The hub router will **dynamically** accept spoke routers. The routers will use a **NHRP registration request** message to register their public IP addresses to the hub.

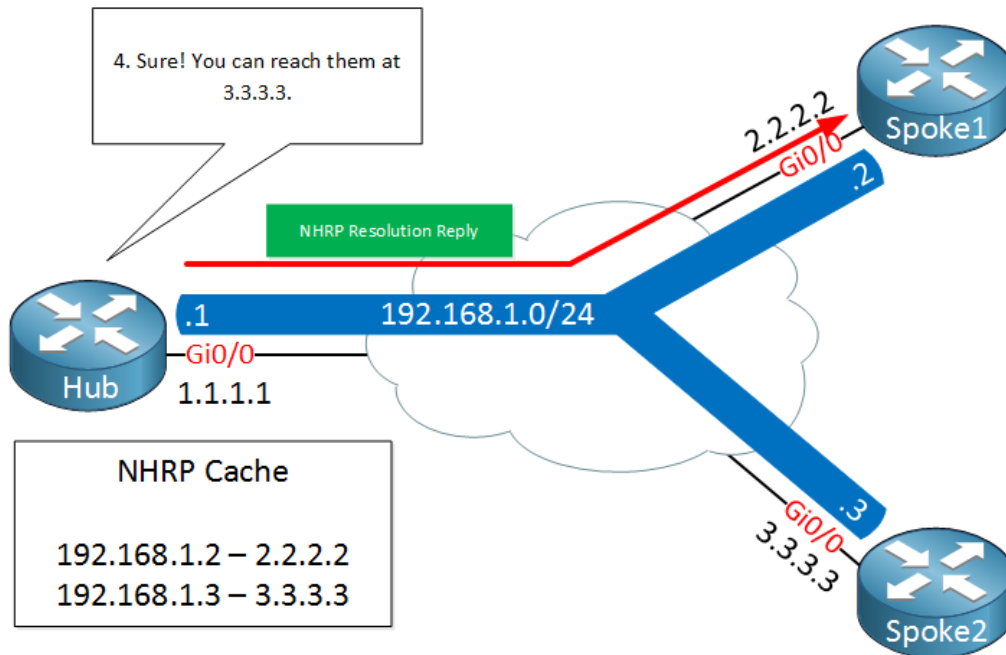




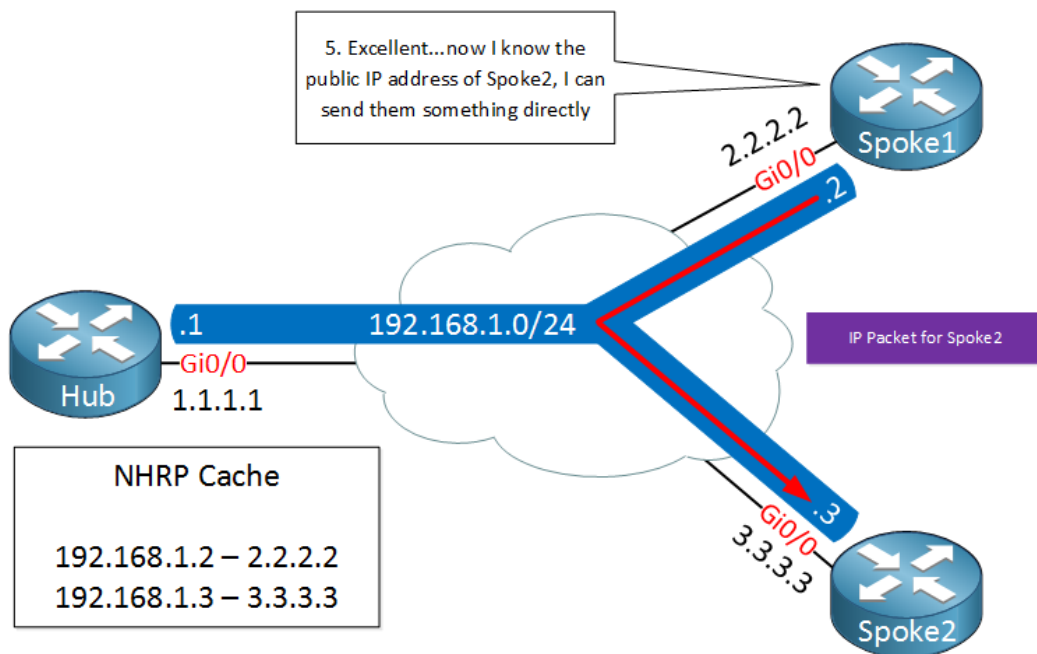
The hub, our NHRP server will create a mapping between the public IP addresses and the IP addresses of the tunnel interfaces.



A few seconds later, spoke1 decides that it wants to send something to spoke2. It needs to figure out the destination public IP address of spoke2 so it will send a **NHRP resolution request**, asking the Hub router what the public IP address of spoke 2 is.



The Hub router checks its cache, finds an entry for spoke 2 and sends the **NHRP resolution reply** to spoke1 with the public IP address of spoke2.



Spoke1 now knows the destination public IP address of spoke2 and is able to tunnel something directly. This is great, we only required the hub to figure out what the public IP address is and all traffic can be sent from spoke to spoke directly.

When we talk about DMVPN, we often refer to an **underlay** and **overlay** network:

- The underlay network is the network we use for connectivity between the different routers, for example the Internet.
- The overlay network is our private network with GRE tunnels.

DMVPN has different versions which we call phases, there's three of them:

- Phase 1
- Phase 2
- Phase 3

Let me give you an overview of the three phases:

#### DMVPN Phase 1

With phase 1 we use NHRP so that spokes can register themselves with the hub. The hub is the only router that is using a multipoint GRE interface, **all spokes will be using regular point-to-point GRE tunnel interfaces**. This means that there will be **no direct spoke-to-spoke** communication, all traffic has to go through the hub!

Since our traffic has to go through the hub, our routing configuration will be quite simple. Spoke routers only need a summary or default route to the hub to reach other spoke routers.

#### DMVPN Phase 2

The disadvantage of phase 1 is that there is no direct spoke to spoke tunnels. In phase 2, **all spoke routers use multipoint GRE** tunnels so we do have direct spoke to spoke tunneling. When a spoke router wants to reach another spoke, it will send an NHRP resolution request to the hub to find the NBMA IP address of the other spoke.

There are two requirements to make spoke to spoke tunnels work:

## Integrating IPsec

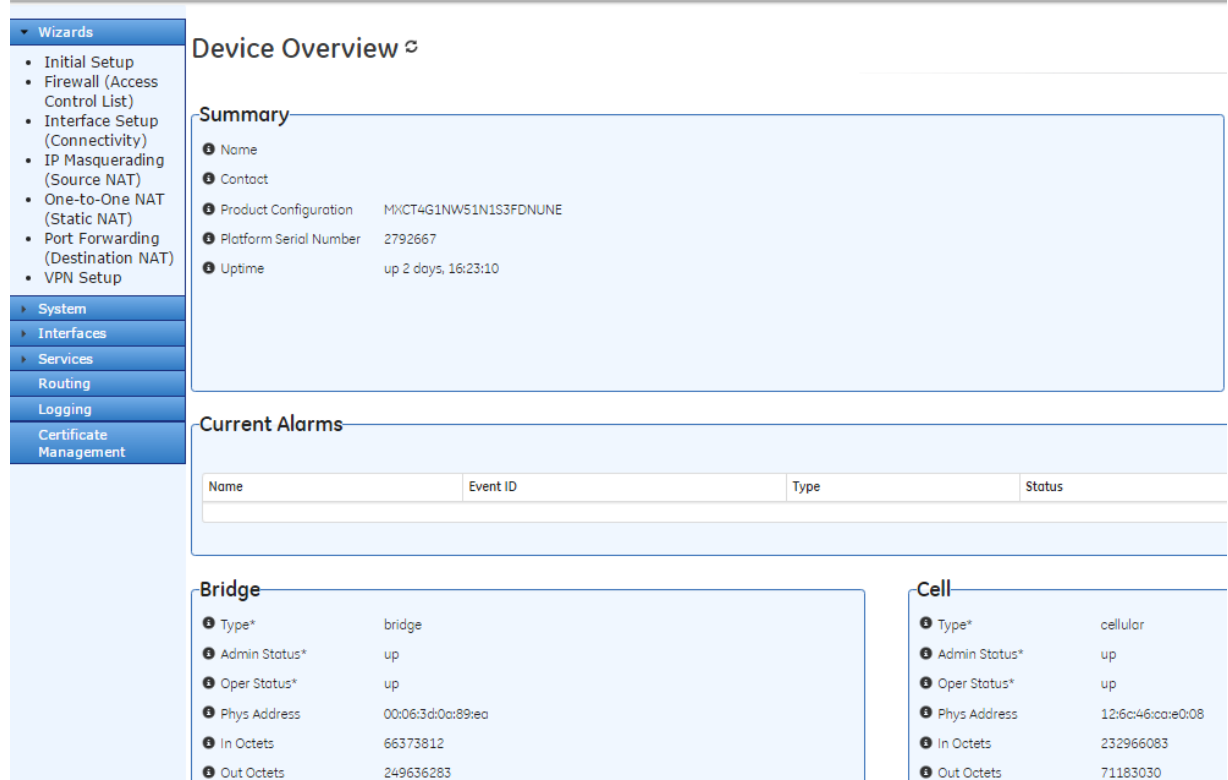
---

Haven't we forgotten something for DMVPN Phase 1/Phase 2? That was IPsec, the components that provides confidentiality and integrity checking to mGRE/NHRP. Now, compared with the complexity of NHRP operations, IPsec integration is straightforward.

First, the hub needs to know how to authentication all the spokes using IKE. The most scalable way is to use X.509 certificates and PKI, but for the simplicity, we will just use the same pre-shared key on all routers. Note that we need to configure the routers with a wild-card pre-shared key, in order to accept IKE negotiation requests from any other dynamic peer.

As for IPsec Phase 2, we need dynamic mapping there, since the hub has no idea of the connecting peer IP addresses. The IPsec phase proxy identities used by the IPsec profile are the source and destination host IP addresses of the tunnel. It makes sense to use IPsec transport mode with mGRE as the latter already provides tunnel encapsulation.

1. Navigate from “Device overview” or the Orbit Home Page to “Wizards” menu as indicated in the screenshot below.



**Wizards**

- Initial Setup
- Firewall (Access Control List)
- Interface Setup (Connectivity)
- IP Masquerading (Source NAT)
- One-to-One NAT (Static NAT)
- Port Forwarding (Destination NAT)
- VPN Setup

**System**

**Interfaces**

**Services**

**Routing**

**Logging**

**Certificate Management**

### Device Overview [↻](#)

#### Summary

- Name
- Contact
- Product Configuration: MXCT4G1NW51N1S3FDNUNE
- Platform Serial Number: 2792667
- Uptime: up 2 days, 16:23:10

#### Current Alarms

Name	Event ID	Type	Status
------	----------	------	--------

#### Bridge

- Type\*: bridge
- Admin Status\*: up
- Oper Status\*: up
- Phys Address: 00:06:3d:0a:89:ea
- In Octets: 66373812
- Out Octets: 249636283

#### Cell

- Type\*: cellular
- Admin Status\*: up
- Oper Status\*: up
- Phys Address: 12:6c:46:ca:e0:08
- In Octets: 232966083
- Out Octets: 71183030

2. Once the “virtual Private Network (VPN) Setup wizard has started, click next to continue.



3. The next screen provides multiple options of creating a VPN tunnel. Choose “Configure Dynamic Multipoint /Mesh VPN (DMVPN)”. Then click next to continue.



4. The next screen will show an image with an example of DMVPN being used. To continue click next. For this page provide a name of your VPN configuration. This name only matters for a local perspective to the orbit you are using. Click next to continue.



- IPsec configuration requires information about the identity of the local orbit and peer orbit. The local portion can be left as a default when configuring an access-point or remote as the orbit can determine its own local address. Additionally, the Peer identity is the Cellular or WAN IP address. This is only defined when configuring a remote with the access-point IP address. When configuring an access-point leave this field default as well. Click next to continue.

## Virtual Private Network (VPN) Setup

IPsec Configuration

Specify the local identity that the peer expects this device to use to identify itself during IKE negotiation.

For pre-shared key based authentication, this is typically the IP address or DNS name (FQDN). If the WAN/Cellular IP address is dynamic, then the DNS name (FQDN) should be specified. For certificate based authentication (pubkey), the Distinguished Name (DN) of the client certificate is implicitly used as the identity. In this case, set this field to default.

Choices▼

Default

Specify the peer identity that this device expects the peer to use to identify itself during IKE negotiation.

For pre-shared key based authentication, this is typically the IP address or DNS name (FQDN) of the peer. If the WAN/Cellular IP address of the peer is dynamic, then the DNS name (FQDN) should be specified. For certificate based authentication (pubkey), the Distinguished Name (DN) of the peer's client certificate is implicitly used as the identity. In this case, set this field to default.

Choices▼

Default

- On this page, we will specify the IKE version, IKE authentication method, and pre-shared key to use for authentication. Remember the IKEv2 should be used unless you are connecting to a third-party device that does not have the option. The pre-shared-key must match the remotes or access-points. Click next to continue.

## Virtual Private Network (VPN) Setup

IPsec Configuration

Specify the IKE version.

Version

Specify the IKE authentication method.

Auth Method \*

Specify the pre-shared-key to use for authentication.

Pre Shared Key

- Moving on with IPsec configuration, specify the encryptions used for Phase 1 and Phase 2. These needs to match any device you are connecting to. We recommend using the encryptions listed in the picture below.

## Virtual Private Network (VPN) Setup

### IPsec Configuration

Specify the IKE (phase-1) encryption, integrity/MAC and key-group parameters. These should match the settings on the peer.

Encryption Algorithm	aes128-cbc
MAC Algorithm	sha256-hmac
DH Group	dh14

Specify the IPsec (phase-2) encryption, integrity/MAC and key-group parameters. These should match the settings on the peer.

Encryption Algorithm	aes128-cbc
MAC Algorithm	sha256-hmac
DH Group	dh14

- GRE configuration section requires only the IP address to be defined. You can add a key or password to the tunnel. You can also modify the MTU or packet size of the tunnel.

## Virtual Private Network (VPN) Setup

### GRE Configuration

Specify tunnel key. This should be the same as the one configured on the peer router.

Key	<input type="text"/>
-----	----------------------

Specify tunnel interface IP address from the DMVPN subnet (e.g. 172.16.0.1/24).

IP address	172.16.0.1/24
------------	---------------

(Optional) Specify tunnel MTU. This should be the same as the one configured on the peer router.

MTU	<input type="text"/>	octets
-----	----------------------	--------



9. NHRP configuration has a few important fields that must be set correctly.
  1. Configure NHRP mapping for the DMVPN router (only check if this Orbit is acting as a remote, if unchecked no other fields are required)
  2. Protocol address refers to the access-point GRE IP address.
  3. Protocol Netmask refers to the GRE subnet mask of the access-point.
  4. NBMA address, is the WAN/Cellular IP address of the hub router.
  5. If the AP or HUB router is a Cisco check the box next to “Cisco”.
  6. Click next to continue

## Virtual Private Network (VPN) Setup

### NHRP Configuration

Uncheck this box if this device acts as a HUB in DMVPN network.

Configure NHRP mapping for the DMVPN HUB router.

Specify the HUB router tunnel network IP address(e.g. 172.16.0.1).

Specify the HUB router tunnel network IP mask (e.g. 255.255.255.0).

Specify the HUB router's transport network address (e.g. 1.1.1.1). Typically, this is the WAN/Cellular IP address of the hub router.

Alternatives ▾

Check this box if HUB router is a Cisco IOS device.

Cisco

(Optional) Select a local interface for which this device will enable creation of shortcut (spoke-to-spoke) tunnels.

Interface

(Optional) Specify a password if peer routers uses NHRP authentication.

10. Finally, this page will apply the applicable firewall rules to the selected interface. Click next to continue.

## Virtual Private Network (VPN) Setup

### Firewall Configuration

Specify the local WAN interface over which this VPN connection will be established. NOTE: The firewall filters applied to this interface will be automatically updated with rules to enable flow of VPN traffic. Select blank entry to prevent any firewall changes.

Interface

11. The next two pages will give you a prompt and a summary review to save your changes. Make sure you click the green submit button to complete the VPN wizard.

## Template Pre-Install Instructions

---

Please follow these instructions. This configuration will be done from the CLI not web gui.

- Open a terminal server client like Putty. If you don't have one you can download it here ( <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html> )
- Log into your Orbit device via ssh with the ip address of the unit
- Once logged in Enter “**configure**” to enter configure mode
- Then enter “ **run set autowizard false**”
- Copy all of the configuration and then right click in the putty window. This will paste the configuration into the orbit. At that point I recommend power cycling your orbit.

## Access-Point Config

---

This section will list configurations for IPSec VPN or DMVPN variations for the Access point Orbit MCR/ECR unit.

### DMVPN Access-Point Config

---

This section will list an generic DMVPN configuration for an Access-point.

- This configuration requires no changes
  - Dynamic VPN AP requires no IP addresses
  - GRE tunnel will work from **172.16.0.1**

```
set services web http enabled false
set services web http port 80
set services web https enabled true
set services web https port 443
set services vpn enabled true
set services vpn ike
set services vpn ike policy DMVPN_AP_t4_ike_policy version ikev2
set services vpn ike policy DMVPN_AP_t4_ike_policy auth-method pre-shared-key
set services vpn ike policy DMVPN_AP_t4_ike_policy pre-shared-key
$4$INrBZmG6dbAPxY9xcJTWQQ==
set services vpn ike policy DMVPN_AP_t4_ike_policy ciphersuite CS1
set services vpn ike peer DMVPN_AP_t4_ike_peer ike-policy
DMVPN_AP_t4_ike_policy
set services vpn ike peer DMVPN_AP_t4_ike_peer local-identity default
set services vpn ike peer DMVPN_AP_t4_ike_peer peer-endpoint any
set services vpn ike peer DMVPN_AP_t4_ike_peer peer-identity default
set services vpn ike peer DMVPN_AP_t4_ike_peer role responder
set services vpn ipsec
```

```

set services vpn ipsec policy DMVPN_AP_t4_ipsec_policy ciphersuite CS1 dh-group
dh14
set services vpn ipsec connection DMVPN_AP_t4 ike-peer DMVPN_AP_t4_ike_peer
set services vpn ipsec connection DMVPN_AP_t4 ipsec-policy
DMVPN_AP_t4_ipsec_policy
set services vpn ipsec connection DMVPN_AP_t4 host-to-host
set services vpn ipsec connection DMVPN_AP_t4 filter input IN_TRUSTED
set services vpn ipsec connection DMVPN_AP_t4 filter output OUT_TRUSTED
set services ssh enabled true
set services ssh port 22
set services nhrp enabled true
set services nhrp interface GRE_DMVPN_AP_t4 authentication
$4$INrBZmG6dbAPxY9xcJTWQQ==
set services netconf enabled true
set services netconf port 830
set services firewall enabled true
set services firewall address-set CELL-IP
set services firewall address-set LOCAL-NETS addresses [ 192.168.1.0/24 ]
set services firewall filter IN_TRUSTED rule 10 match protocol all
set services firewall filter IN_TRUSTED rule 10 actions
set services firewall filter IN_TRUSTED rule 10 actions action accept
set services firewall filter IN_UNTRUSTED rule 1 match protocol icmp
set services firewall filter IN_UNTRUSTED rule 1 actions
set services firewall filter IN_UNTRUSTED rule 1 actions action accept
set services firewall filter IN_UNTRUSTED rule 2 match protocol udp
set services firewall filter IN_UNTRUSTED rule 2 match src-port
set services firewall filter IN_UNTRUSTED rule 2 match src-port services [ dns ike ]
set services firewall filter IN_UNTRUSTED rule 3 match protocol udp
set services firewall filter IN_UNTRUSTED rule 3 match dst-port
set services firewall filter IN_UNTRUSTED rule 3 match dst-port services [ ike ]
set services firewall filter IN_UNTRUSTED rule 3 actions
set services firewall filter IN_UNTRUSTED rule 3 actions action accept
set services firewall filter IN_UNTRUSTED rule 4 match protocol esp
set services firewall filter IN_UNTRUSTED rule 4 actions
set services firewall filter IN_UNTRUSTED rule 4 actions action accept
set services firewall filter IN_UNTRUSTED rule 5 match protocol tcp
set services firewall filter IN_UNTRUSTED rule 5 match dst-port
set services firewall filter IN_UNTRUSTED rule 5 match dst-port services [ http https
ssh ]
set services firewall filter IN_UNTRUSTED rule 5 actions
set services firewall filter IN_UNTRUSTED rule 5 actions action accept
set services firewall filter IN_UNTRUSTED rule 10 match protocol all
set services firewall filter IN_UNTRUSTED rule 10 actions
set services firewall filter IN_UNTRUSTED rule 10 actions action drop
set services firewall filter IN_UNTRUSTED rule 11 match protocol esp

```

```
set services firewall filter IN_UNTRUSTED rule 11 actions
set services firewall filter IN_UNTRUSTED rule 11 actions action accept
set services firewall filter IN_UNTRUSTED rule 12 match protocol all
set services firewall filter IN_UNTRUSTED rule 12 actions
set services firewall filter IN_UNTRUSTED rule 12 actions action drop
set services firewall filter OUT_TRUSTED rule 10 match protocol all
set services firewall filter OUT_TRUSTED rule 10 actions
set services firewall filter OUT_TRUSTED rule 10 actions action accept
set services firewall filter OUT_UNTRUSTED rule 1 match protocol all
set services firewall filter OUT_UNTRUSTED rule 1 match src-address
set services firewall filter OUT_UNTRUSTED rule 1 match src-address address-set
CELL-IP
set services firewall filter OUT_UNTRUSTED rule 1 match src-address add-interface-
address true
set services firewall filter OUT_UNTRUSTED rule 1 actions
set services firewall filter OUT_UNTRUSTED rule 1 actions action accept
set services firewall filter OUT_UNTRUSTED rule 2 match protocol all
set services firewall filter OUT_UNTRUSTED rule 2 actions
set services firewall filter OUT_UNTRUSTED rule 2 actions action drop
set services firewall filter OUT_UNTRUSTED rule 3 match protocol all
set services firewall filter OUT_UNTRUSTED rule 3 actions
set services firewall filter OUT_UNTRUSTED rule 3 actions action drop
set services firewall filter OUT_UNTRUSTED rule 10 match protocol all
set services firewall filter OUT_UNTRUSTED rule 10 actions
set services firewall filter OUT_UNTRUSTED rule 10 actions action drop
set services firewall nat source rule-set MASQ rule 1 source-nat interface
set interfaces interface GRE_DMVPN_AP_t4 type gre
set interfaces interface GRE_DMVPN_AP_t4 gre-config
set interfaces interface GRE_DMVPN_AP_t4 gre-config mode ip-over-gre
set interfaces interface GRE_DMVPN_AP_t4 gre-config src-address 0.0.0.0
set interfaces interface GRE_DMVPN_AP_t4 gre-config dst-address 0.0.0.0
set interfaces interface GRE_DMVPN_AP_t4 gre-config key 255
set interfaces interface GRE_DMVPN_AP_t4 gre-config ipsec-connection
DMVPN_AP_t4
set interfaces interface GRE_DMVPN_AP_t4 ipv4
set interfaces interface GRE_DMVPN_AP_t4 ipv4 mtu 1500
set interfaces interface GRE_DMVPN_AP_t4 ipv4 address 172.16.0.1 prefix-length 24
set interfaces interface GRE_DMVPN_AP_t4 filter input IN_TRUSTED
set interfaces interface GRE_DMVPN_AP_t4 filter output OUT_TRUSTED
set system name AP
```

## REMOTE Configuration

---

These VPN configurations will be for the remote orbit units in the field. These units will require multiple modifications such as LAN, Cell, and GRE IP addresses. If you need additional assistance, please reach out to the GE technical services group.

## DMVPN REMOTE Configuration

---

This configuration will detail what changes need to be done to the Remote DMVPN configuration.

Variables that need to be changed (NOTE: you can

- **ACCESS-POINT CELL IP** – change this to the ACCESS POINT CELL IP ADDRESS
- **172.16.0.2** – REMOTE GRE IP needs to be incremented by 1

```
set services vpn enabled true
set services vpn ike
set services vpn ike policy DMVPN_REMOTE_t4_ike_policy version ikev2
set services vpn ike policy DMVPN_REMOTE_t4_ike_policy auth-method pre-shared-key
set services vpn ike policy DMVPN_REMOTE_t4_ike_policy pre-shared-key
$4$INrBZmG6dbAPxY9xcJTWQQ==
set services vpn ike policy DMVPN_REMOTE_t4_ike_policy ciphersuite CS1
set services vpn ike peer DMVPN_REMOTE_t4_ike_peer ike-policy
DMVPN_REMOTE_t4_ike_policy
set services vpn ike peer DMVPN_REMOTE_t4_ike_peer local-identity default
set services vpn ike peer DMVPN_REMOTE_t4_ike_peer peer-endpoint address ACCESS-
POINT CELL IP
set services vpn ike peer DMVPN_REMOTE_t4_ike_peer peer-identity address ACCESS-
POINT CELL IP
set services vpn ike peer DMVPN_REMOTE_t4_ike_peer role initiator

set services vpn ipsec
set services vpn ipsec policy DMVPN_REMOTE_t4_ipsec_policy ciphersuite CS1 dh-group
dh14
set services vpn ipsec connection DMVPN_REMOTE_t4_ike_peer
DMVPN_REMOTE_t4_ike_peer
set services vpn ipsec connection DMVPN_REMOTE_t4_ipsec-policy
DMVPN_REMOTE_t4_ipsec_policy
set services vpn ipsec connection DMVPN_REMOTE_t4 host-to-host
set services vpn ipsec connection DMVPN_REMOTE_t4 filter input IN_TRUSTED
set services vpn ipsec connection DMVPN_REMOTE_t4 filter output OUT_TRUSTED
```

```
set services nhrp enabled true
set services nhrp interface GRE_DMVPN_REMOTE_t4 map PRIMARY-HUB protocol-address
172.16.0.1
set services nhrp interface GRE_DMVPN_REMOTE_t4 map PRIMARY-HUB protocol-
netmask 255.255.255.0
set services nhrp interface GRE_DMVPN_REMOTE_t4 map PRIMARY-HUB nbma-address
ACCESS-POINT CELL IP
set services nhrp interface GRE_DMVPN_REMOTE_t4 map PRIMARY-HUB register true
set services nhrp interface GRE_DMVPN_REMOTE_t4 map PRIMARY-HUB cisco true
set services nhrp interface GRE_DMVPN_REMOTE_t4 authentication
$4$INrBZmG6dbAPxY9xcJTWQQ==
```

```
set services firewall enabled true
set services firewall address-set CELL-IP
set services firewall address-set LOCAL-NETS addresses [ 192.168.1.0/24 ]
set services firewall filter IN_TRUSTED rule 10 match protocol all
set services firewall filter IN_TRUSTED rule 10 actions
set services firewall filter IN_TRUSTED rule 10 actions action accept
set services firewall filter IN_UNTRUSTED rule 1 match protocol icmp
set services firewall filter IN_UNTRUSTED rule 1 actions
set services firewall filter IN_UNTRUSTED rule 1 actions action accept
set services firewall filter IN_UNTRUSTED rule 2 match protocol udp
set services firewall filter IN_UNTRUSTED rule 2 match src-port
set services firewall filter IN_UNTRUSTED rule 2 match src-port services [ dns ]
set services firewall filter IN_UNTRUSTED rule 10 match protocol udp
set services firewall filter IN_UNTRUSTED rule 10 match dst-port
set services firewall filter IN_UNTRUSTED rule 10 match dst-port services [ dns ike ntp ]
set services firewall filter IN_UNTRUSTED rule 10 actions
set services firewall filter IN_UNTRUSTED rule 10 actions action accept
set services firewall filter IN_UNTRUSTED rule 11 match protocol esp
set services firewall filter IN_UNTRUSTED rule 11 actions
set services firewall filter IN_UNTRUSTED rule 11 actions action accept
set services firewall filter IN_UNTRUSTED rule 12 match protocol all
set services firewall filter IN_UNTRUSTED rule 12 actions
set services firewall filter IN_UNTRUSTED rule 12 actions action drop
set services firewall filter OUT_TRUSTED rule 10 match protocol all
set services firewall filter OUT_TRUSTED rule 10 actions
set services firewall filter OUT_TRUSTED rule 10 actions action accept
set services firewall filter OUT_UNTRUSTED rule 1 match src-address
set services firewall filter OUT_UNTRUSTED rule 1 match src-address address-set CELL-IP
set services firewall filter OUT_UNTRUSTED rule 1 match src-address add-interface-address
true
set services firewall filter OUT_UNTRUSTED rule 1 actions
set services firewall filter OUT_UNTRUSTED rule 1 actions action accept
```

```
set services firewall filter OUT_UNTRUSTED rule 2 match protocol all
set services firewall filter OUT_UNTRUSTED rule 2 actions
set services firewall filter OUT_UNTRUSTED rule 2 actions action drop
```

```
set interfaces interface GRE_DMVPN_REMOTE_t4 type gre
set interfaces interface GRE_DMVPN_REMOTE_t4 gre-config
set interfaces interface GRE_DMVPN_REMOTE_t4 gre-config mode ip-over-gre
set interfaces interface GRE_DMVPN_REMOTE_t4 gre-config src-address 0.0.0.0
set interfaces interface GRE_DMVPN_REMOTE_t4 gre-config dst-address 0.0.0.0
set interfaces interface GRE_DMVPN_REMOTE_t4 gre-config key 255
set interfaces interface GRE_DMVPN_REMOTE_t4 gre-config ipsec-connection
DMVPN_REMOTE_t4
set interfaces interface GRE_DMVPN_REMOTE_t4 ipv4
set interfaces interface GRE_DMVPN_REMOTE_t4 ipv4 mtu 1500
set interfaces interface GRE_DMVPN_REMOTE_t4 ipv4 address 172.16.0.2 prefix-length 24
set interfaces interface GRE_DMVPN_REMOTE_t4 filter input IN_TRUSTED
set interfaces interface GRE_DMVPN_REMOTE_t4 filter output OUT_TRUSTED
commit
quit
```

## GE Technical Services

---

Please reach out to GE Technical Services if you have any issues with your configurations.  
GEMDS.techsupport@ge.com

OR

Call +1 585.241.5510 (Technical Services Help Line)

## Content Sources

---

1. DMVPN content - <http://blog.ine.com/2008/08/02/dmvpn-explained/>
2. DMVPN content – Cisco.com
3. Orbit related information – gemds.com
4. Custom Created content - GEMDS tech services

*End of application bulletin.*