GRE Protocol Interface

This chapter provides information on Generic Routing Encapsulation protocol interface support in the GGSN or P-GW service node. The product Administration Guides provide examples and procedures for configuration of basic services on the system. It is recommended that you select the configuration example that best meets your service model, and configure the required elements for that model, as described in the respective product Administration Guide, before using the procedures in this chapter.

Important GRE protocol interface support is a licensed Cisco feature. A separate feature license may be required. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the *Managing License Keys* section of the *Software Management Operations* chapter in the *System Administration Guide*.

Important Commands used in the configuration samples in this section provide base functionality to the extent that the most common or likely commands and/or keyword options are presented. In many cases, other optional commands and/or keyword options are available. Refer to the *Command Line Interface Reference* for complete information regarding all commands.

Introduction

GRE protocol functionality adds one additional protocol on Cisco's multimedia core platforms (ASR 5500 or higher) to support mobile users to connect to their enterprise networks through Generic Routing Encapsulation (GRE).

GRE tunnels can be used by the enterprise customers of a carrier 1) To transport AAA packets corresponding to an APN over a GRE tunnel to the corporate AAA servers and, 2) To transport the enterprise subscriber packets over the GRE tunnel to the corporation gateway.

The corporate servers may have private IP addresses and hence the addresses belonging to different enterprises may be overlapping. Each enterprise needs to be in a unique virtual routing domain, known as VRF. To differentiate the tunnels between same set of local and remote ends, GRE Key will be used as a differentiator.

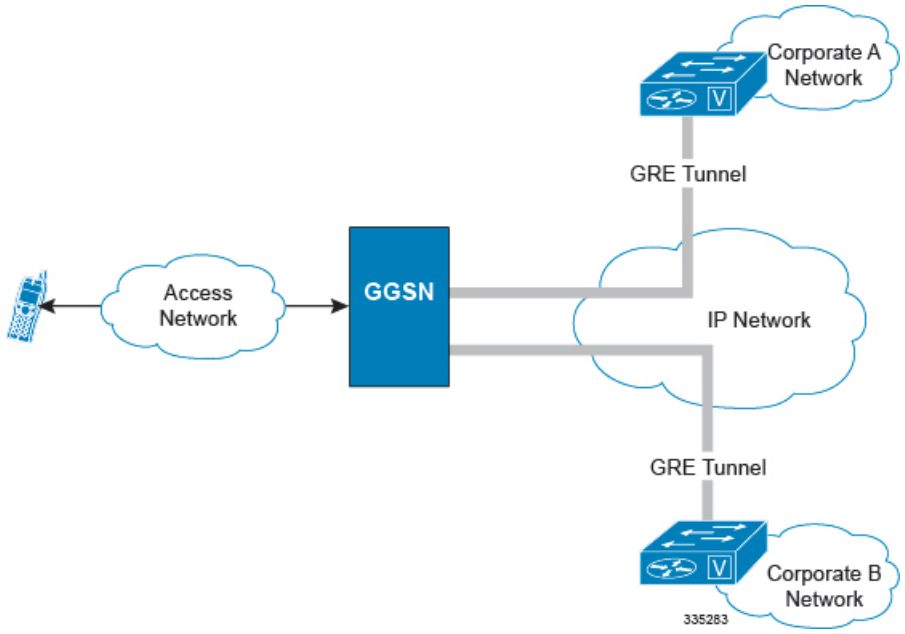
It is a common technique to enable multi-protocol local networks over a single-protocol backbone, to connect non-contiguous networks and allow virtual private networks across WANs. This mechanism encapsulates data packets from one protocol inside a different protocol and transports the data packets unchanged across a foreign network. It is important to note that GRE tunneling does not provide security to the encapsulated protocol, as there is no encryption involved (like IPSEC offers, for example).

GRE Tunneling consists of three main components:

- Passenger protocol-protocol being encapsulated. For example: CLNS, IPv4 and IPv6.
- Carrier protocol-protocol that does the encapsulating. For example: GRE, IP-in-IP, L2TP, MPLS and IPSec.
- Transport protocol-protocol used to carry the encapsulated protocol. The main transport protocol is IP.

The most simplified form of the deployment scenario is shown in the following figure, in which GGSN has two APNs talking to two corporate networks over GRE tunnels.

Figure 1: GRE Interface Deployment Scenario



Supported Standards

Support for the following standards and requests for comments (RFCs) have been added with this interface support:

- RFC 1701, Generic Routing Encapsulation (GRE)
- RFC 1702, Generic Routing Encapsulation over IPv4 networks
- RFC 2784, Generic Routing Encapsulation (GRE)
- RFC 2890, Key and Sequence Number Extensions to GRE

Supported Networks and Platforms

This feature supports all systems with StarOS Release 9.0 or later running GGSN and/or SGSN service for the core network services. The P-GW service supports this feature with StarOS Release 12.0 or later.

Licenses

GRE protocol interface support is a licensed Cisco feature. A separate feature license may be required. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the *Managing License Keys* section of the *Software Management Operations* chapter in the *System Administration Guide*.

Services and Application on GRE Interface

GRE interface implementation provides the following functionality with GRE protocol support.

How GRE Interface Support Works

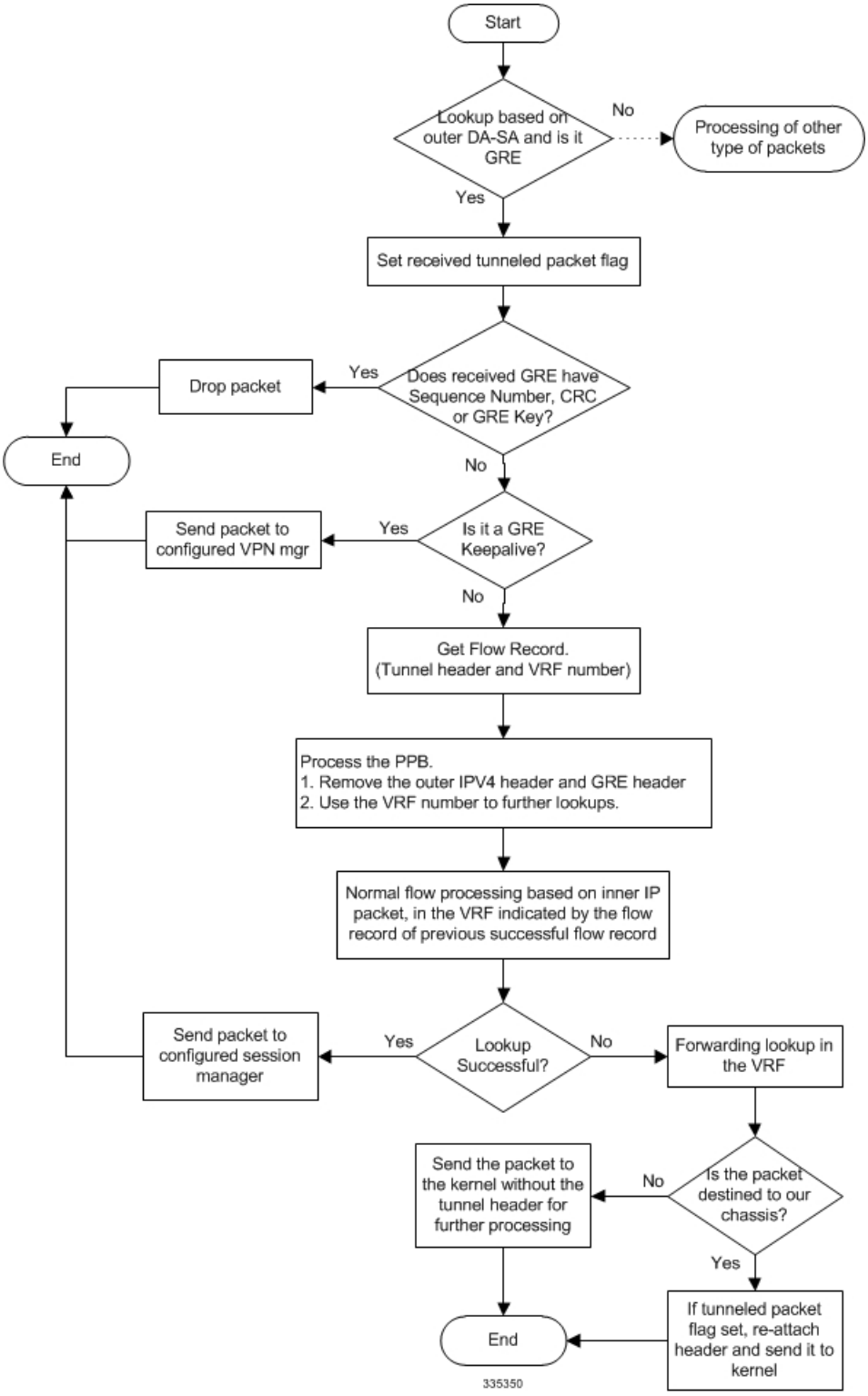
The GRE interface provides two types of data processing; one for ingress packets and another for egress packets.

Ingress Packet Processing on GRE Interface

Figure given below provides a flow of process for incoming packets on GRE interface.

Note that in case the received packet is a GRE keep-alive or a ping packet then the outer IPV4 and GRE header are not stripped off (or get reattached), but instead the packet is forwarded as is to the VPN manager or kernel respectively. In case of all other GRE tunneled packets the IPV4 and GRE header are stripped off before sending the packet for a new flow lookup.

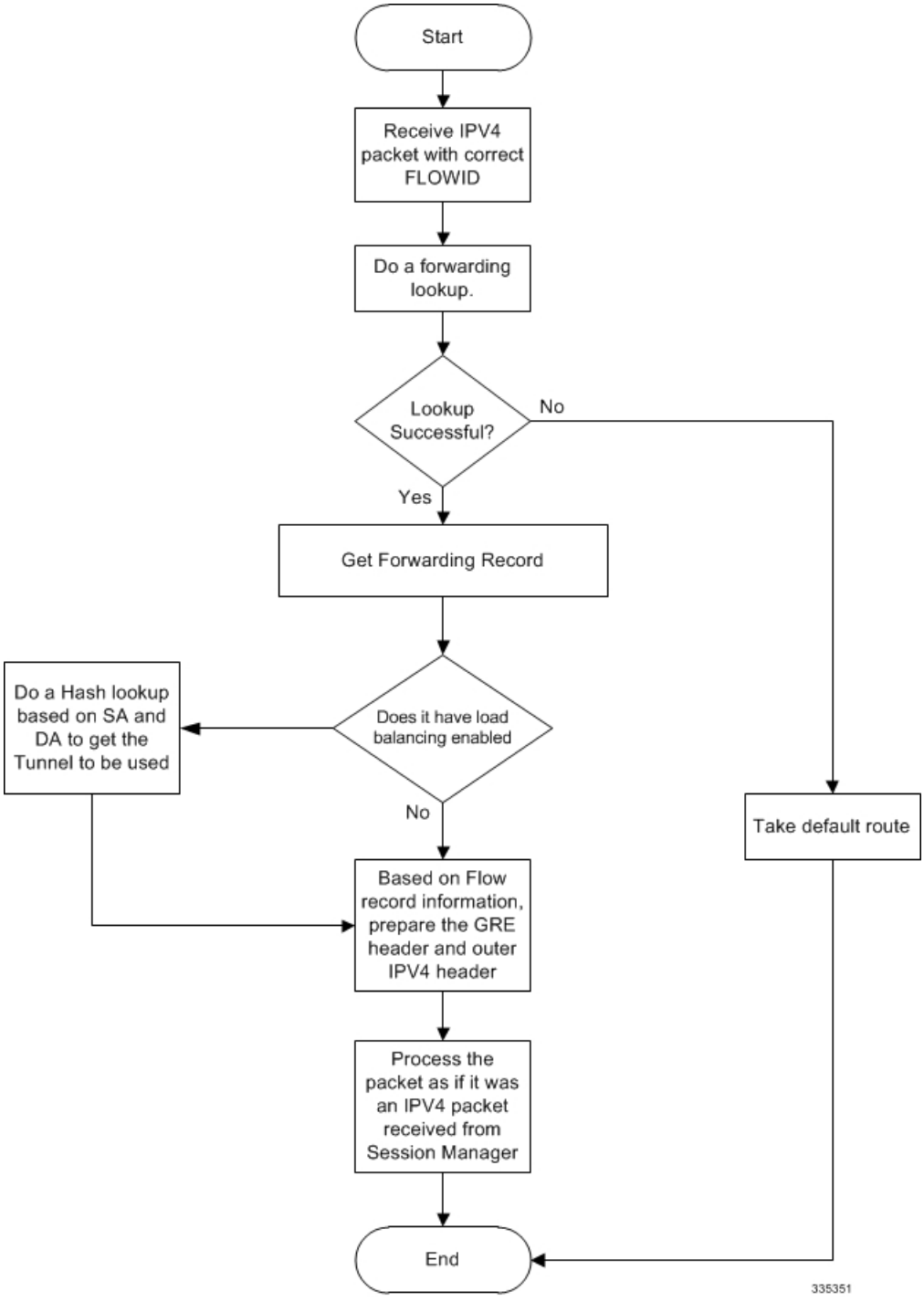
Figure 2: Ingress Packet Processing on GRE Interface



Egress Packet Processing on GRE Interface

Figure given below provides a flow of process for outgoing packets on GRE interface:

Figure 3: Egress Packet Processing on GRE Interface



GRE Interface Configuration

This section provides a high-level series of steps and the associated configuration examples for configuring the system with GRE interface in GGSN or P-GW services.

Important This section provides the minimum instruction set to enable the GRE Protocol Interface support functionality on a GGSN or P-GW. Commands that configure additional functions for this feature are provided in the *Command Line Interface Reference*.

These instructions assume that you have already configured the system level configuration as described in *System Administration Guide* and specific product Administration Guide.

To configure the system to support GRE tunnel interface:

Step 1 Configure the virtual routing and forwarding (VRF) in a context by applying the example configurations presented in Virtual Routing And Forwarding (VRF) Configuration, on page 6.

Step 2 Configure the GRE tunnel interface in a context by applying the example configurations presented in GRE Tunnel Interface Configuration, on page 7.

Step 3 Enable OSPF for the VRF and for the given network by applying the example configurations presented in Enabling OSPF for VRF, on page 7.

Step 4 Associate IP pool and AAA server group with VRF by applying the example configurations presented in Associating IP Pool and AAA Group with VRF, on page 8.

Step 5 Associate APN with VRF through AAA server group and IP pool by applying the example configurations presented in Associating APN with VRF, on page 8.

Step 6 Optional. If the route to the server is not learnt from the corporate over OSPFv2, static route can be configured by applying the example configurations presented in Static Route Configuration, on page 8.

Step 7 Save your configuration to flash memory, an external memory device, and/or a network location using the Exec mode command **save configuration**. For additional information on how to verify and save configuration files, refer to the *System Administration Guide* and the *Command Line Interface Reference*.

Step 8 Verify configuration of GRE and VRF related parameters by applying the commands provided in Verifying Your Configuration, on page 9.

Virtual Routing And Forwarding (VRF) Configuration

This section provides the configuration example to configure the VRF in a context:

```
configure
  context <vpn_context_name> -noconfirm ]
    ip vrf <vrf_name>
      ip maximum-routes <max_routes>
      end
```

Notes:

- <*vpn_context_name*> is the name of the system context you want to use for VRF. For more information, refer *System Administration Guide*.
- A maximum of 300 VRFs per context and up to 2,048 VRFs per chassis can be configured on system.
- <*vrf_name*> is name of the VRF which is to be associated with various interfaces.
- A maximum of 10000 routes can be configured through **ip maximum-routes** <*max_routes*> command.

GRE Tunnel Interface Configuration

This section provides the configuration example to configure the GRE tunnel interface and associate a VRF with GRE interface:

```
configure
   context <vpn_context_name>
      ip interface <intfc_name> tunnel
         ip vrf forwarding <vrf_name>
            ip address <internal_ip_address/mask>
            tunnel-mode gre
            source interface <non_tunn_intfc_to_corp>
            destination address <global_ip_address>
            keepalive interval <value> num-retry <retry>
            end
```

Notes:

- <*vpn_context_name*> is the name of the system context you want to use for GRE interface configuration. For more information, refer *Command Line Interface Reference*.
- A maximum of 511 GRE tunnels + 1 non-tunnel interface can be configured in one context. System needs at least 1 non-tunnel interface as a default.
- <*intfc_name*> is name of the IP interface which is defined as a tunnel type interface and to be used for GRE tunnel interface.
- <*vrf_name*> is the name of the VRF which is preconfigured in context configuration mode.
- <*internal_ip_address*/mask> is the network IP address with sub-net mask to be used for VRF forwarding.
- <*non_tunn_intfc_to_corp*> is the name a non-tunnel interface which is required by system as source interface and preconfigured. For more information on interface configuration refer *System Administration Guide*.
- <*global_ip_address*> is a globally reachable IP address to be used as a destination address.

Enabling OSPF for VRF

This section provides the configuration example to enable the OSPF for VRF to support GRE tunnel interface:

```
configure
   context <vpn_context_name>
      router ospf
         ip vrf <vrf_name>
         network <internal_ip_address/mask>
         end
```

Notes:

- <*vpn_context_name*> is the name of the system context you want to use for OSPF routing. For more information, refer *Routing* in this guide.

- <*vrf_name*> is the name of the VRF which is preconfigured in context configuration mode.
- <*internal_ip_address/mask*> is the network IP address with sub-net mask to be used for OSPF routing.

Associating IP Pool and AAA Group with VRF

This section provides the configuration example for associating IP pool and AAA groups with VRF:

```
configure
   context <vpn_context_name>
      ip pool <ip_pool_name> <internal_ip_address/mask> vrf <vrf_name>
        exit
      aaa group <aaa_server_group>
        ip vrf <vrf_name>
        end
```

Notes:

- <*vpn_context_name*> is the name of the system context you want to use for IP pool and AAA server group.
- <*ip_pool_name*> is name of a preconfigured IP pool. For more information refer *System Administration Guide*.
- <*aaa_server_group*> is name of a preconfigured AAA server group. For more information refer *AAA Interface Administration and Reference*.
- <*vrf_name*> is the name of the VRF which is preconfigured in context configuration mode.
- <*internal_ip_address/mask*> is the network IP address with sub-net mask to be used for IP pool.

Associating APN with VRF

This section provides the configuration example for associating an APN with VRF through AAA group and IP pool:

```
configure
   context <vpn_context_name>
      apn <apn_name>
        aaa group <aaa_server_group>
        ip address pool name <ip_pool_name>
        end
```

Notes:

- <*vpn_context_name*> is the name of the system context you want to use for APN configuration.
- <*ip_pool_name*> is name of a preconfigured IP pool. For more information refer *System Administration Guide*.
- <*aaa_server_group*> is name of a preconfigured AAA server group. For more information refer *AAA Interface Administration and Reference*.
- <*vrf_name*> is the name of the VRF which is preconfigured in context configuration mode.

Static Route Configuration

This section provides the optional configuration example for configuring static routes when the route to the server is not learnt from the corporate over OSPFv2:

```
configure
  context <vpn_context_name>
    ip route <internal_ip_address/mask> tunnel <tunnel_intfc_name> vrf <vrf_name>
    end
```

Notes:

- <*vpn_context_name*> is the name of the system context you want to use for static route configuration.
- <*internal_ip_address/mask*> is the network IP address with sub-net mask to be used as static route.
- <*tunnel_intfc_name*> is name of a predefined tunnel type IP interface which is to be used for GRE tunnel interface.
- <*vrf_name*> is the name of the VRF which is preconfigured in context configuration mode.

Verifying Your Configuration

This section explains how to display and review the configurations after saving them in a .cfg file as described in the *System Administration Guide* and also to retrieve errors and warnings within an active configuration for a service.

Important All commands listed here are under Exec mode. Not all commands are available on all platforms.

These instructions are used to verify the GRE interface configuration.

Step 1 Verify that your interfaces are configured properly by entering the following command in Exec Mode:

show ip interface

The output of this command displays the configuration of the all interfaces configured in a context.

```
Intf Name:     foo1
Intf Type:             Broadcast
Description:
IP State:              UP (Bound to 17/2 untagged, ifIndex 285343745)
IP Address:            209.165.200.225         Subnet Mask:         255.255.255.0
Bcast Address:         209.165.200.254             MTU:                 1500
Resoln Type:           ARP                   ARP timeout:         60 secs
L3 monitor LC-port switchover: Disabled
Number of Secondary Addresses: 0
Intf Name:             foo2
Intf Type:             Tunnel (GRE)
Description:
VRF:                   vrf-tun
IP State:              UP (Bound to local address 209.165.200.225 (foo1), remote address
209.165.200.229)
IP Address:            209.165.200.228              Subnet Mask:         255.255.255.224
Intf Name:             foo3
Intf Type:             Tunnel (GRE)
Description:
IP State:              DOWN (<state explaining the reason of being down>)
IP Address:            209.165.200.232            Subnet Mask:         255.255.255.224
```

Step 2 Verify that GRE keep alive is configured properly by entering the following command in Exec Mode:

show ip interface gre-keepalive

The output of this command displays the configuration of the keepalive for GRE interface configured in a context.