SAE Gateway Configuration

This chapter provides configuration information for the SAE Gateway (SAEGW).

Important Information about all commands in this chapter can be found in the *Command Line Interface Reference*.

Because each wireless network is unique, the system is designed with a variety of parameters allowing it to perform in various wireless network environments. In this chapter, only the minimum set of parameters are provided to make the system operational. Optional configuration commands specific to the SAEGW product are located in the *Command Line Interface Reference*.

Configuring an SAEGW Service

This section provides a high-level series of steps and the associated configuration file examples for configuring the system to perform as an SAEGW in a test environment. Information provided in this section includes the following:

Information Required

The following sections describe the minimum amount of information required to configure and make the SAEGW operational on the network. To make the process more efficient, it is recommended that this information be available prior to configuring the system.

There are additional configuration parameters that are not described in this section. These parameters deal mostly with fine-tuning the operation of the SAEGW in the network. Information on these parameters can be found in the appropriate sections of the *Command Line Interface Reference*.

Required Local Context Configuration Information

The following table lists the information that is required to configure the local context on an SAEGW.

Table 1: Required Information for Local Context Configuration

Required Information	Description
Management Interface Configuration	
Interface name	An identification string between 1 and 79 characters (alpha and/or numeric) by which the interface will be recognized by the system. Multiple names are needed if multiple interfaces will be configured.
IP address and subnet	IPv4 addresses assigned to the interface. Multiple addresses and subnets are needed if multiple interfaces will be configured.
Physical port number	The physical port to which the interface will be bound. Ports are identified by the chassis slot number where the line card resides followed by the number of the physical connector on the card. For example, port 17/1 identifies connector number 1 on the card in slot 17. A single physical port can facilitate multiple interfaces.
Gateway IP address	Used when configuring static IP routes from the management interface(s) to a specific network.
Security administrator name	The name or names of the security administrator with full rights to the system.
Security administrator password	Open or encrypted passwords can be used.
Remote access type(s)	The type of remote access that will be used to access the system such as telnetd, sshd, and/or ftpd.

Required SAEGW Context Configuration Information

The following table lists the information that is required to configure the SAEGW context on an SAEGW.

Table 2: Required Information for SAEGW Context Configuration

Required Information	Description
SAEGW context name	An identification string from 1 to 79 characters (alpha and/or numeric) by which the SAEGW context will be recognized by the system.
SAEGW Service Configuration	
SAEGW service name	An identification string from 1 to 63 characters (alpha and/or numeric) by which the SAEGW service will be recognized by the system.
S-GW service name	An identification string from 1 to 63 characters (alpha and/or numeric) by which the S-GW service will be recognized by the system.

Required Information	Description
P-GW service name	An identification string from 1 to 63 characters (alpha and/or numeric) by which the P-GW service will be recognized by the system.

SAEGW Configuration

Step 1 Set system configuration parameters such as activating PSCs by applying the example configurations found in the *System Administration Guide*.

Step 2 Set initial configuration parameters such as creating contexts and services by applying the example configurations found in Initial Configuration, on page 3.

Step 3 Configure the system to perform as an SAEGW and associate an eGTP S-GW service and eGTP P-GW service by applying the example configurations presented in SAEGW Service Configuration, on page 4.

Step 4 Verify and save the configuration by following the steps found in Verifying and Saving the Configuration, on page 4.

Initial Configuration

Step 1 Set local system management parameters by applying the example configuration in Modifying the Local Context, on page 3.

Step 2 Create the context where the SAEGW, S-GW, and P-GW services will reside by applying the example configuration in Creating and Configuring an SAEGW Context, on page 4.

Step 3 Configure an eGTP S-GW service by applying the example configurations found in Configuring an eGTP S-GW Service, on page 4.

Step 4 Configure an eGTP P-GW service by applying the example configurations found inConfiguring an eGTP P-GW Service, on page 31.

Modifying the Local Context

Use the following example to set the default subscriber and configure remote access capability in the local context:

```
configure
  context local
    interface <lcl_cntxt_intrfc_name>
      ip address <ip_address> <ip_mask>
      exit
    server ftpd
    exit
    server telnetd
    exit
    subscriber default
    exit
    administrator <name> encrypted password <password> ftp
    ip route <ip_addr/ip_mask> <next_hop_addr> <lcl_cntxt_intrfc_name>
```

```
exit
port ethernet <slot/port>
   no shutdown
   bind interface <lcl_cntxt_intrfc_name> local
   end
```

Notes:

- Service names must be unique across all contexts within a chassis.

Creating and Configuring an SAEGW Context

Use the following example to create the context where the SAEGW, S-GW, and P-GW services will reside:

```
configure
   context <saegw_context_name>
   end
```

SAEGW Service Configuration

Configure the SAEGW service by applying the example configuration in Configuring the SAEGW Service, on page 4.

Configuring the SAEGW Service

Use the following example to configure the SAEGW service:

```
configure
   context <saegw_context_name>
      saegw-service <saegw_service_name> -noconfirm
         associate sgw-service <sgw_service_name>
         associate pgw-service <pgw_service_name>
         end
```

Notes:

- The SAEGW, S-GW, and P-GW services must all reside within the same SAEGW context.
- Service names must be unique across all contexts within a chassis.

Verifying and Saving the Configuration

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*.

Configuring an eGTP S-GW Service

This section provides a high-level series of steps and the associated configuration file examples for configuring the system to perform as an eGTP S-GW in a test environment. Information provided in this section includes the following:

- Information Required, on page 5

Information Required

The following sections describe the minimum amount of information required to configure and make the S-GW operational on the network. To make the process more efficient, you should have this information available prior to configuring the system.

There are additional configuration parameters that are not described in this section. These parameters deal mostly with fine-tuning the operation of the S-GW in the network. Information on these parameters can be found in the appropriate sections of the *Command Line Interface Reference*.

Required S-GW Ingress Context Configuration Information

The following table lists the information that is required to configure the S-GW ingress context on an eGTP S-GW.

Table 3: Required Information for S-GW Ingress Context Configuration

Required Information	Description
S-GW ingress context name	An identification string from 1 to 79 characters (alpha and/or numeric) by which the S-GW ingress context is recognized by the system. **Note** The S-GW service must reside within the SAEGW context, thus this would be the SAEGW context name.
Accounting policy name	An identification string from 1 to 63 characters (alpha and/or numeric) by which the accounting policy is recognized by the system. The accounting policy is used to set parameters for the Rf (off-line charging) interface. **Important** In StarOS releases 19 and later, the Rf interface is not supported on the S-GW.
S1-U/S11 Interface Configuration (To/from eNodeB/MME) **Note** The configuration provided in this guide assumes a shared S1-U/S11 interface. These interfaces can be separated to support a different network architecture. The information below applies to both.	
Interface name	An identification string between 1 and 79 characters (alpha and/or numeric) by which the interface is recognized by the system. Multiple names are needed if multiple interfaces will be configured.
IP address and subnet	IPv4 or IPv6 addresses assigned to the interface. Multiple addresses and subnets are needed if multiple interfaces will be configured.

Required Information	Description
Physical port number	The physical port to which the interface will be bound. Ports are identified by the chassis slot number where the line card resides followed by the number of the physical connector on the card. For example, port 17/1 identifies connector number 1 on the card in slot 17. A single physical port can facilitate multiple interfaces.
Gateway IP address	Used when configuring static IP routes from the interface(s) to a specific network.
Gateway IP address	Used when configuring static IP routes from the interface(s) to a specific network.
GTP-U Service Configuration	
GTP-U service name (for S1-U/S11 interface)	An identification string from 1 to 63 characters (alpha and/or numeric) by which the GTP-U service bound to the S1-U/S11 interface will be recognized by the system.
IP address	S1-U/S11 interface IPv4 or IPv6 address.
S-GW Service Configuration	
S-GW service name	An identification string from 1 to 63 characters (alpha and/or numeric) by which the S-GW service is recognized by the system. Multiple names are needed if multiple S-GW services will be used.
eGTP Ingress Service Configuration	
eGTP S1-U/S11 ingress service name	An identification string from 1 to 63 characters (alpha and/or numeric) by which the eGTP S1-U/S11 ingress service is recognized by the system.

Required S-GW Egress Context Configuration Information

The following table lists the information that is required to configure the S-GW egress context on an eGTP S-GW.

Table 4: Required Information for S-GW Egress Context Configuration

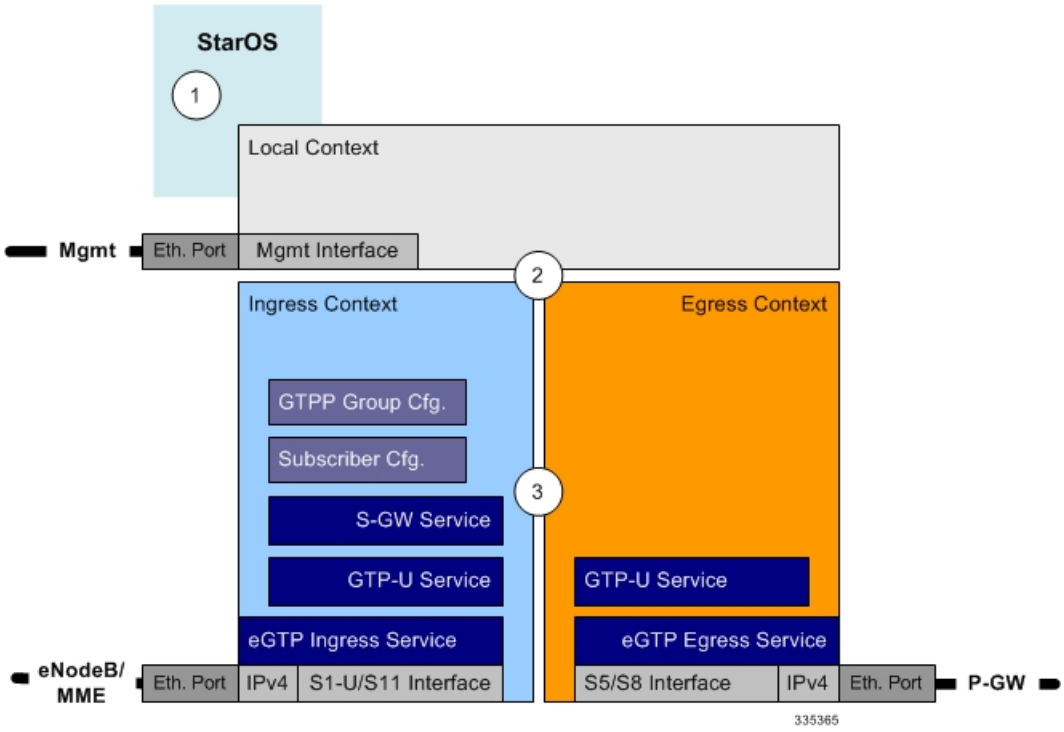
Required Information	Description
S-GW egress context name	An identification string from 1 to 79 characters (alpha and/or numeric) by which the S-GW egress context is recognized by the system. **Note** The S-GW service must reside within the SAEGW context, thus this would be the SAEGW context name.
S5/S8 Interface Configuration (To/from P-GW)	
Interface name	An identification string between 1 and 79 characters (alpha and/or numeric) by which the interface is recognized by the system. Multiple names are needed if multiple interfaces will be configured.

Required Information	Description
IP address and subnet	IPv4 or IPv6 addresses assigned to the interface. Multiple addresses and subnets are needed if multiple interfaces will be configured.
Physical port number	The physical port to which the interface will be bound. Ports are identified by the chassis slot number where the line card resides followed by the number of the physical connector on the card. For example, port 17/1 identifies connector number 1 on the card in slot 17. A single physical port can facilitate multiple interfaces.
Gateway IP address	Used when configuring static IP routes from the interface(s) to a specific network.
GTP-U Service Configuration	
GTP-U service name (for S5/S8 interface)	An identification string from 1 to 63 characters (alpha and/or numeric) by which the GTP-U service bound to the S5/S8 interface will be recognized by the system.
IP address	S5/S8 interface IPv4 or IPv6 address.
eGTP Egress Service Configuration	
eGTP Egress Service Name	An identification string from 1 to 63 characters (alpha and/or numeric) by which the eGTP egress service is recognized by the system.

How This Configuration Works

The following figure and supporting text describe how this configuration with a single ingress and egress context is used by the system to process a subscriber call.

Figure 1: eGTP S-GW Call Processing Using a Single Ingress and Egress Context

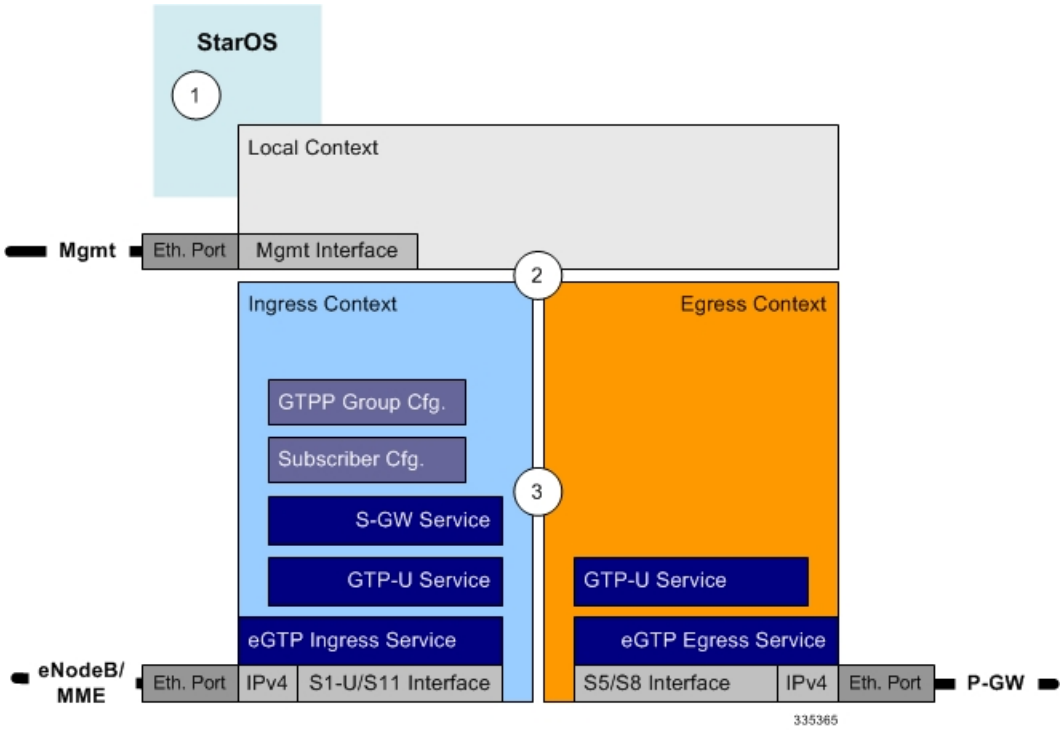


1. A subscriber session from the MME is received by the S-GW service over the S11 interface.
2. The S-GW service determines which context to use to access PDN services for the session. This process is described in the How the System Selects Contexts section located in the Understanding the System Operation and Configuration chapter of the System Administration Guide.
3. S-GW uses the configured egress context to determine the eGTP service to use for the outgoing S5/S8 connection.
4. The S-GW establishes the S5/S8 connection by sending a create session request message to the P-GW.
5. The P-GW responds with a Create Session Response message that includes the PGW S5/S8 Address for control plane and bearer information.
6. The S-GW conveys the control plane and bearer information to the MME in a Create Session Response message.
7. The MME responds with a Create Bearer Response and Modify Bearer Request message.
8. The S-GW sends a Modify Bearer Response message to the MME.

eGTP S-GW Configuration

To configure the system to perform as an eGTP S-GW, review the following graphic and subsequent steps.

Figure 2: eGTP S-GW Configurable Components



Step 1 Set system configuration parameters such as activating PSCs by applying the example configurations found in the System Administration Guide.

Step 2 Set initial configuration parameters such as creating contexts and services by applying the example configurations found in Initial Configuration, on page 9.

Step 3 Configure the system to perform as an eGTP S-GW and set basic S-GW parameters such as eGTP interfaces and an IP route by applying the example configurations presented in eGTP Configuration, on page 11.

Step 4 Verify and save the configuration by following the instruction in Verifying and Saving the Configuration, on page 13.

Initial Configuration

Step 1 Create an ingress context where the S-GW and eGTP ingress service will reside by applying the example configuration in Creating an S-GW Ingress Context, on page 10.

Step 2 Create an eGTP ingress service within the newly created ingress context by applying the example configuration in Creating an eGTP Ingress Service, on page 10.

Step 3 Create an S-GW egress context where the eGTP egress services will reside by applying the example configuration in Creating an S-GW Egress Context, on page 10.

Step 4 Create an eGTP egress service within the newly created egress context by applying the example configuration in Creating an eGTP Egress Service, on page 11.

Step 5 Create a S-GW service within the newly created ingress context by applying the example configuration in Creating an S-GW Service, on page 11.

Creating an S-GW Ingress Context

Use the following example to create an S-GW ingress context and Ethernet interfaces to an MME and eNodeB, and bind the interfaces to configured Ethernet ports.

```
configure
   context <saegw_context_name> -noconfirm
      subscriber default
      exit
      interface <s1u-s11_interface_name>
         ip address <ipv4_address_primary>
         ip address <ipv4_address_secondary>
         exit
      ip route 0.0.0.0 0.0.0.0 <next_hop_address> <sgw_interface_name>
      exit
   port ethernet <slot_number/port_number>
      no shutdown
      bind interface <s1u-s11_interface_name> <saegw_context_name>
      end
```

Notes:

- This example presents the S1-U/S11 connections as a shared interface. These interfaces can be separated to support a different network architecture.
- The S1-U/S11 interface IP address(es) can also be specified as IPv6 addresses using the **ipv6 address** command.
- Service names must be unique across all contexts within a chassis.

Creating an eGTP Ingress Service

Use the following configuration example to create an eGTP ingress service:

```
configure
   context <saegw_context_name>
      egtp-service <egtp_ingress_service_name> -noconfirm
      end
```

Notes:

- Service names must be unique across all contexts within a chassis.

Creating an S-GW Egress Context

Use the following example to create an S-GW egress context and Ethernet interface to a P-GW and bind the interface to configured Ethernet ports.

```
configure
   context <egress_context_name> -noconfirm
      interface <s5s8_interface_name> tunnel
         ipv6 address <address>
            tunnel-mode ipv6ip
```

```
            source interface <name>
            destination address <ipv4 or ipv6 address>
            end
configure
  port ethernet <slot_number/port_number>
    no shutdown
    bind interface <s5s8_interface_name> <egress_context_name>
    end
```

Notes:

- The S5/S8 interface IP address can also be specified as an IPv4 address using the **ip address** command.
- Service names must be unique across all contexts within a chassis.

Creating an eGTP Egress Service

Use the following configuration example to create an eGTP egress service in the S-GW egress context:

```
configure
  context <egress_context_name>
    egtp-service <egtp_egress_service_name> -noconfirm
    end
```

Notes:

- Service names must be unique across all contexts within a chassis.

Creating an S-GW Service

Use the following configuration example to create the S-GW service in the ingress context:

```
configure
    context <saegw_context_name>
      sgw-service <sgw_service_name> -noconfirm
      end
```

Notes:

- Service names must be unique across all contexts within a chassis.

eGTP Configuration

Step 1 Set the system's role as an eGTP S-GW and configure eGTP service settings by applying the example configuration in Setting the System's Role as an eGTP S-GW and Configuring GTP-U and eGTP Service Settings, on page 11.

Step 2 Configure the S-GW service by applying the example configuration in Configuring the S-GW Service, on page 12.

Step 3 Specify an IP route to the eGTP Serving Gateway by applying the example configuration in Configuring an IP Route, on page 13.

Setting the System's Role as an eGTP S-GW and Configuring GTP-U and eGTP Service Settings

Use the following configuration example to set the system to perform as an eGTP S-GW and configure the GTP-U and eGTP services.

Important If you modify the **interface-type** command, the parent service (service within which the eGTP/GTP-U service is configured) will automatically restart. Service restart results in dropping of active calls associated with the parent service.

```
configure
  context <saegw_context_name>
    gtpp group default
    exit
  gtpu-service <gtpu_ingress_service_name>
    bind ipv4-address <s1-u_s11_interface_ip_address>
    exit
  egtp-service <egtp_ingress_service_name>
    interface-type interface-sgw-ingress
      validation-mode default
      associate gtpu-service <gtpu_ingress_service_name>
      gtpc bind address <s1u-s11_interface_ip_address>
      exit
    exit
  context <sgw_egress_context_name>
    gtpu-service <gtpu_egress_service_name>
      bind ipv4-address <s5s8_interface_ip_address>
      exit
    egtp-service <egtp_egress_service_name>
      interface-type interface-sgw-egress
      validation-mode default
      associate gtpu-service <gtpu_egress_service_name>
      gtpc bind address <s5s8_interface_ip_address>
      end
```

Notes:

- The **bind** command in the GTP-U ingress and egress service configuration can also be specified as an IPv6 address using the **ipv6-address** command.
- Service names must be unique across all contexts within a chassis.

Configuring the S-GW Service

Use the following example to configure the S-GW service:

```
configure
  context <saegw_context_name>
    sgw-service <sgw_service_name> -noconfirm
      associate ingress egtp-service <egtp_ingress_service_name>
      associate egress-proto gtp egress-context <egress_context_name>
      qci-qos-mapping <map_name>
      end
```

Notes:

- Service names must be unique across all contexts within a chassis.

Configuring an IP Route

Use the following example to configure an IP Route for control and user plane data communication with an eGTP PDN Gateway:

```
configure
  context <egress_context_name>
    ip route <pgw_ip_addr/mask> <sgw_next_hop_addr> <sgw_intrfc_name>
    end
```

Notes:

- Service names must be unique across all contexts within a chassis.

Verifying and Saving the Configuration

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*.

Configuring Optional Features on the eGTP S-GW

The configuration examples in this section are optional and provided to cover the most common uses of the eGTP S-GW in a live network. The intent of these examples is to provide a base configuration for testing.

The following optional configurations are provided in this section:

Configuring the GTP Echo Timer

The GTP echo timer on the ASR5500 S-GW can be configured to support two different types of path management: default and dynamic. This timer can be configured on the GTP-C and/or the GTP-U channels.

Default GTP Echo Timer Configuration

The following examples describe the configuration of the default eGTP-C and GTP-U interface echo timers:

eGTP-C

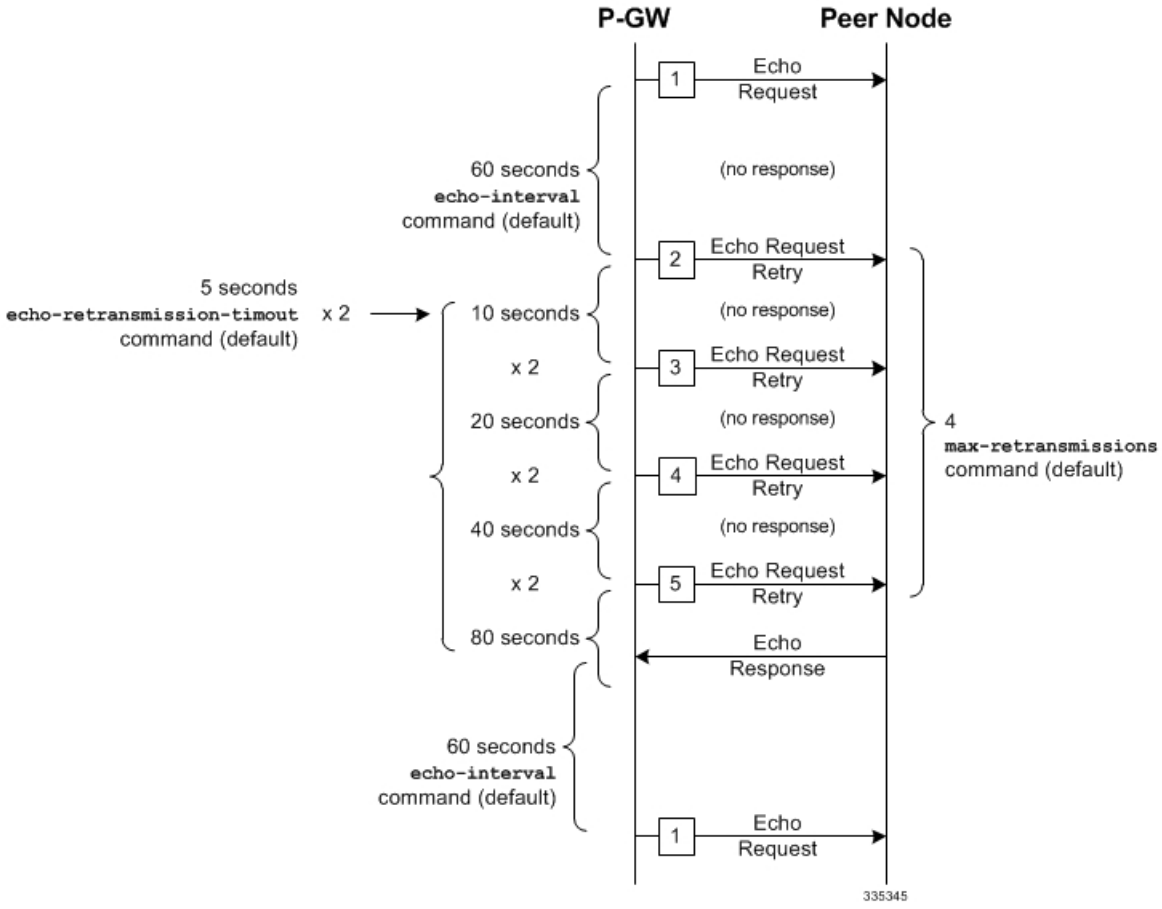
```
configure
  configure
    context <context_name>
```

```
egtp-service <egtp_service_name>
  gtpc echo-interval <seconds>
  gtpc echo-retransmission-timeout <seconds>
  gtpc max-retransmissions <num>
  end
```

Notes:

- This configuration can be used in either the ingress context supporting the S1-U and/or S11 interfaces with the eNodeB and MME respectively; and the egress context supporting the S5/S8 interface with the P-GW.
- Service names must be unique across all contexts within a chassis.
- The following diagram describes a failure and recovery scenario using default settings of the three **gtpc** commands in the example above:

Figure 3: Failure and Recovery Scenario: Example 1



- The multiplier (x2) is system-coded and cannot be configured.

GTP-U

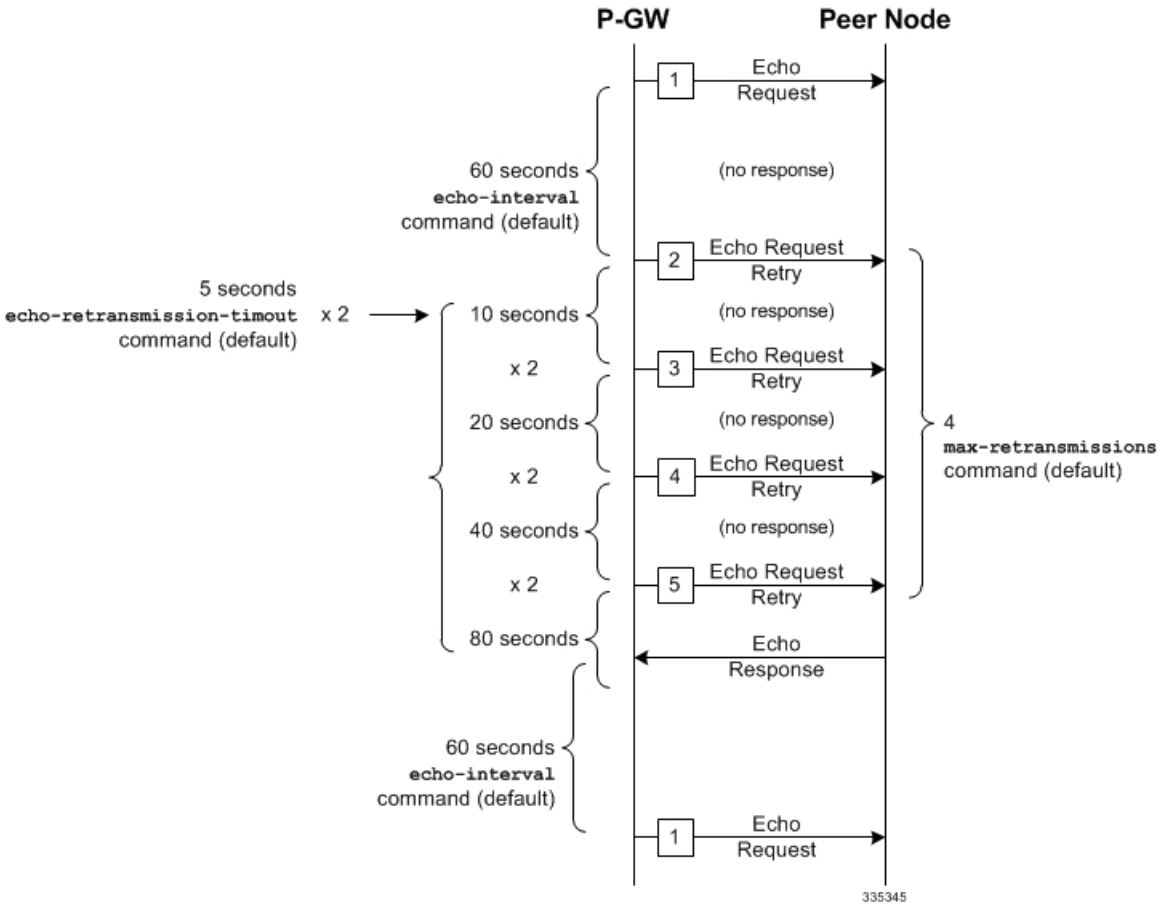
```
configure
  configure
    context <context_name>
```

```
gtpu-service <gtpu_service_name>
   echo-interval <seconds>
   echo-retransmission-timeout <seconds>
   max-retransmissions <num>
   end
```

Notes:

- This configuration can be used in either the ingress context supporting the S1-U interfaces with the eNodeB and the egress context supporting the S5/S8 interface with the P-GW.
- Service names must be unique across all contexts within a chassis.
- The following diagram describes a failure and recovery scenario using default settings of the three GTP-U commands in the example above:

Figure 4: Failure and Recovery Scenario: Example 2



- The multiplier (x2) is system-coded and cannot be configured.

Dynamic GTP Echo Timer Configuration

The following examples describe the configuration of the dynamic eGTP-C and GTP-U interface echo timers:

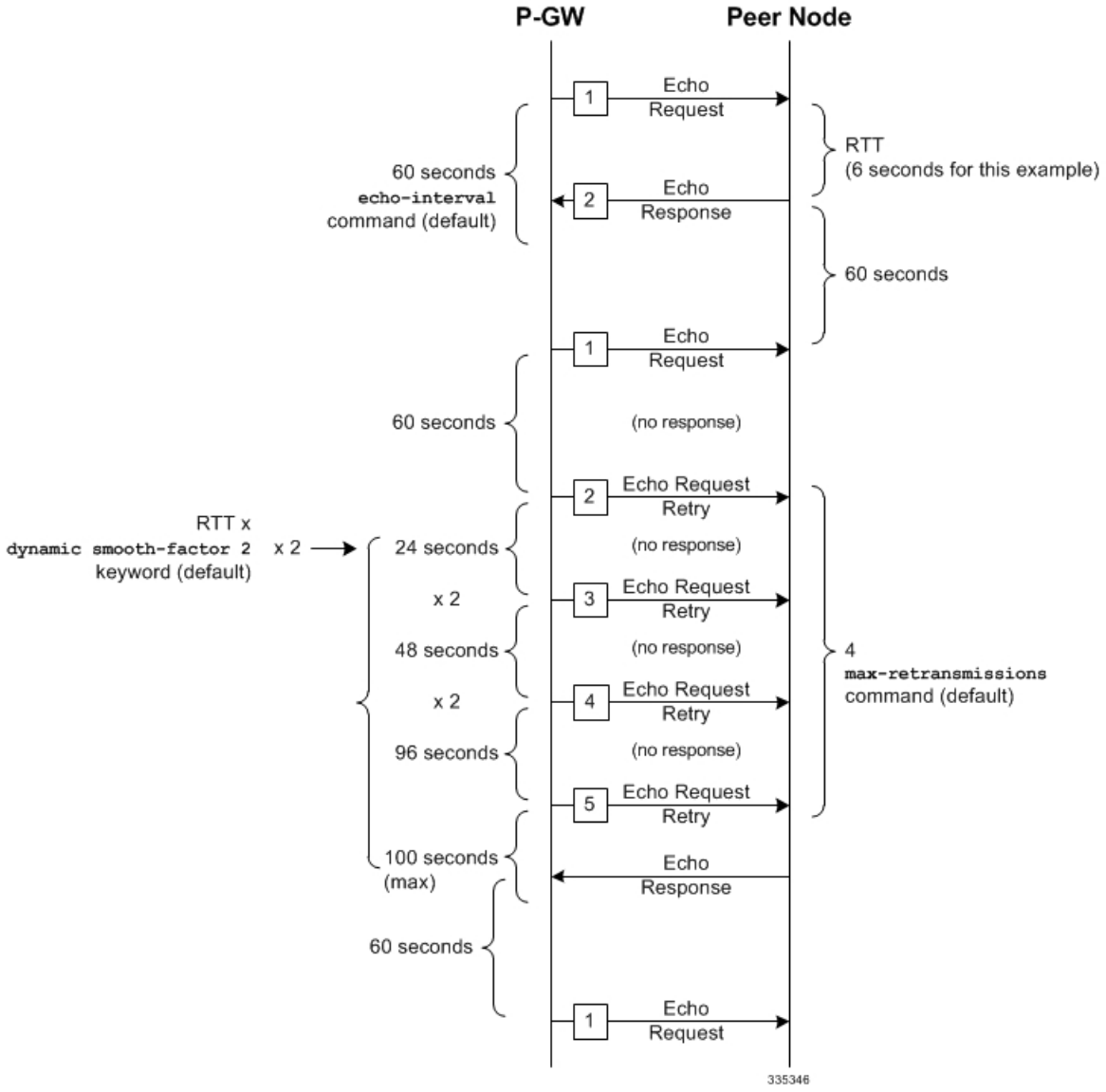
eGTP-C

```
configure
  configure
    context <context_name>
      egtp-service <egtp_service_name>
        gtpc echo-interval <seconds> dynamic smooth-factor <multiplier>
        gtpc echo-retransmission-timeout <seconds>
        gtpc max-retransmissions <num>
        end
```

Notes:

- This configuration can be used in either the ingress context supporting the S1-U and/or S11 interfaces with the eNodeB and MME respectively; and the egress context supporting the S5/S8 interface with the P-GW.
- Service names must be unique across all contexts within a chassis.
- The following diagram describes a failure and recovery scenario using default settings of the three **gtpc** commands in the example above and an example round trip timer (RTT) of six seconds:

Figure 5: Failure and Recovery Scenario: Example 3



- The multiplier (x2) and the 100 second maximum are system-coded and cannot be configured.

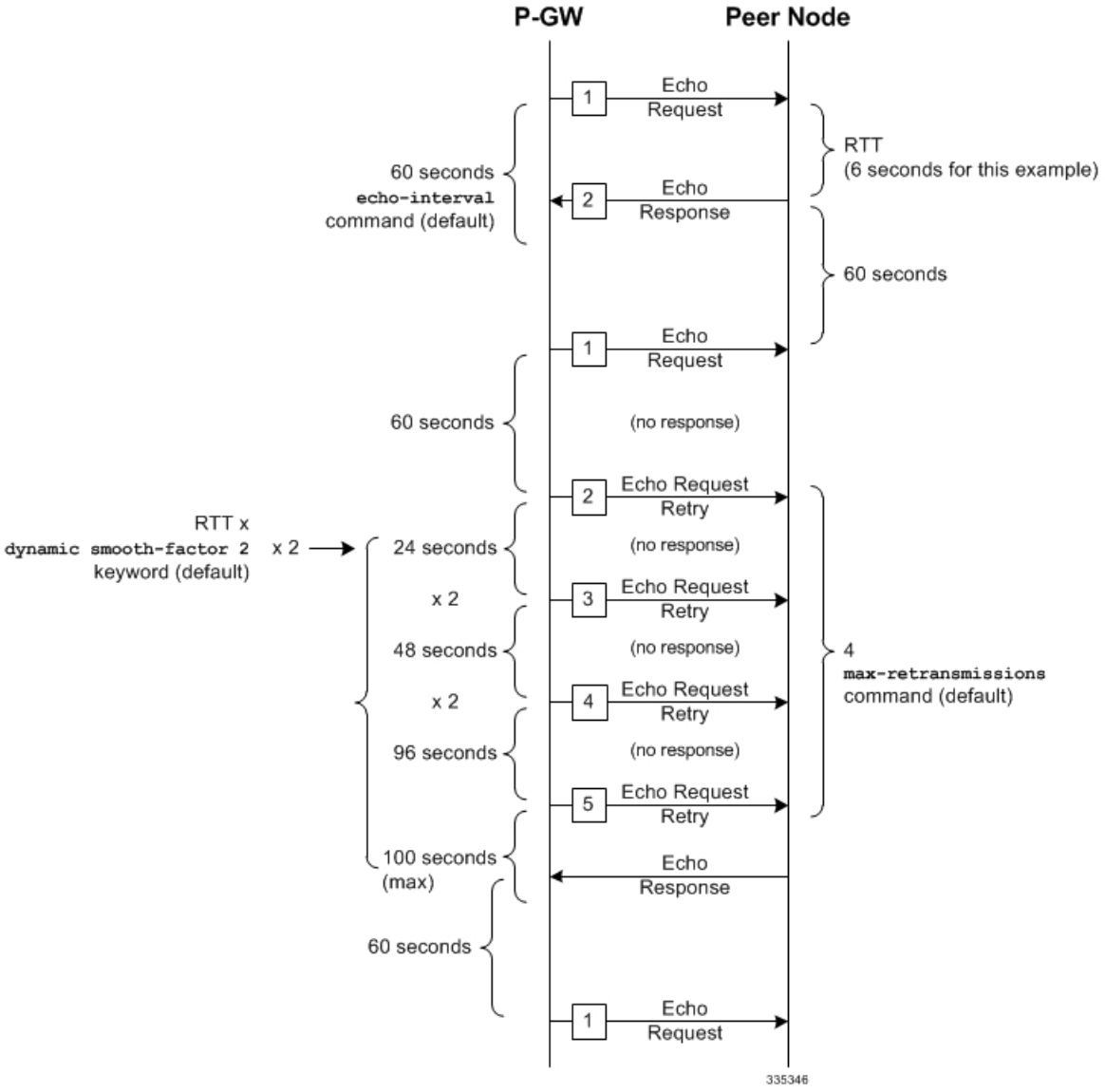
GTP-U

```
configure
  configure
    context <context_name>
      gtpu-service <gtpu_service_name>
        echo-interval <seconds> dynamic smooth-factor <multiplier>
        echo-retransmission-timeout <seconds>
        max-retransmissions <num>
        end
```

Notes:

- This configuration can be used in either the ingress context supporting the S1-U interfaces with the eNodeB and the egress context supporting the S5/S8 interface with the P-GW.
- Service names must be unique across all contexts within a chassis.
- The following diagram describes a failure and recovery scenario using default settings of the three **gtpu** commands in the example above and an example round trip timer (RTT) of six seconds:

Figure 6: Failure and Recovery Scenario: Example 4



- The multiplier (x2) and the 100 second maximum are system-coded and cannot be configured.

Configuring GTPP Offline Accounting on the S-GW

By default the S-GW service supports GTPP accounting. To provide GTPP offline charging during, for example, scenarios where the foreign P-GW does not, configure the S-GW with the example parameters below:

```
configure
  gtpp single-source
    context <saegw_context_name>
      subscriber default
        accounting mode gtpp
        exit
        gtpp group default
          gtpp charging-agent address <gz_ipv4_address>
          gtpp echo-interval <seconds>
          gtpp attribute diagnostics
          gtpp attribute local-record-sequence-number
          gtpp attribute node-id-suffix <string>
          gtpp dictionary <name>
          gtpp server <ipv4_address> priority <num>
          gtpp server <ipv4_address> priority <num> node-alive enable
          exit
        policy accounting <gz_policy_name>
          accounting-level {type}
          operator-string <string>
          cc profile <index> buckets <num>
          cc profile <index> interval <seconds>
          cc profile <index> volume total <octets>
          exit
        sgw-service <sgw_service_name>
          accounting context <saegw_context_name> gtpp group default
          associate accounting-policy <gz_policy_name>
          exit
        exit
    context <saegw_context_name>
      interface <gz_interface_name>
        ip address <address>
        exit
      exit
    port ethernet <slot_number/port_number>
      no shutdown
      bind interface <gz_interface_name> <saegw_context_name>
      end
```

Notes:

- **gtpp single-source** is enabled to allow the system to generate requests to the accounting server using a single UDP port (by way of a AAA proxy function) rather than each AAA manager generating requests on unique UDP ports.
- **gtpp** is the default option for the **accounting mode** command.
- An accounting mode configured for the call-control profile will override this setting.

- **accounting-level** types are: flow, PDN, PDN-QCI, QCI, and subscriber. Refer to the Accounting Profile Configuration Mode Commands chapter in the *Command Line Interface Reference* for more information on this command.
- Service names must be unique across all contexts within a chassis.

Configuring Diameter Offline Accounting on the S-GW

By default the S-GW service supports GTPP accounting. You can enable accounting via RADIUS/Diameter (Rf) for the S-GW service. To provide Rf offline charging during, for example, scenarios where the foreign P-GW does not, configure the S-GW with the example parameters below:

Important In StarOS 19 and later versions, this feature is not supported on the S-GW.

```
configure
   operator-policy name <policy_name>
      associate call-control-profile <call_cntrl_profile_name>
      exit
   call-control-profile <call_cntrl_profile_name>
      accounting mode radius-diameter
         exit
      lte-policy
         subscriber-map <map_name>
            precedence <number> match-criteria all operator-policy-name
<policy_name>
            exit
         exit
      context <saegw_context_name>
         policy accounting <rf_policy_name>
            accounting-level {type}
            operator-string <string>
            exit
         sgw-service <sgw_service_name>
            associate accounting-policy <rf_policy_name>
            associate subscriber-map <map_name>
            exit
         aaa group <rf-radius_group_name>
            radius attribute nas-identifier <id>
            radius accounting interim interval <seconds>
            radius dictionary <name>
            radius mediation-device accounting server <address> key <key>
            diameter authentication dictionary <name>
            diameter accounting dictionary <name>
            diameter accounting endpoint <rf_cfg_name>
            diameter accounting server <rf_cfg_name> priority <num>
            exit
         diameter endpoint <rf_cfg_name>
            use-proxy
            origin realm <realm_name>
            origin host <name> address <rf_ipv4_address>
```

```
      peer <rf_cfg_name> realm <name> address <ofcs_ipv4_or_ipv6_addr>
      route-entry peer <rf_cfg_name>
      exit
   exit
context <saegw_context_name>
   interface <rf_interface_name>
      ip address <rf_ipv4_address>
      exit
   exit
port ethernet <slot_number/port_number>
   no shutdown
   bind interface <rf_interface_name> <saegw_context_name>
   end
```

Notes:

- **accounting-level** types are: flow, PDN, PDN-QCI, QCI, and subscriber. Refer to the Accounting Profile Configuration Mode Commands chapter in the *Command Line Interface Reference* for more information on this command.
- The Rf interface IP address can also be specified as an IPv6 address using the **ipv6 address** command.
- Service names must be unique across all contexts within a chassis.

Configuring APN-level Traffic Policing on the S-GW

To enable traffic policing for scenarios where the foreign subscriber's P-GW doesn't enforce it, use the following configuration example:

```
configure
   apn-profile <apn_profile_name>
     qos rate-limit downlink non-gbr-qci committed-auto-readjust duration
 <seconds> exceed-action {action} violate-action {action}
     qos rate-limit uplink non-gbr-qci committed-auto-readjust duration
<seconds> exceed-action {action} violate-action {action}
     exit
   operator-policy name <policy_name>
     apn default-apn-profile <apn_profile_name>
     exit
   lte-policy
     subscriber-map <map_name>
       precendence <number> match-criteria all operator-policy-name
<policy_name>
       exit
     sgw-service <sgw_service_name>
       associate subscriber-map <map_name>
       end
```

Notes:

- For the **qos rate-limit** command, the actions supported for **violate-action** and **exceed-action** are: **drop**, **lower-ip-precedence**, and **transmit**.
- Service names must be unique across all contexts within a chassis.

Configuring X.509 Certificate-based Peer Authentication

The configuration example in this section enables X.509 certificate-based peer authentication, which can be used as the authentication method for IP Security on the S-GW.

Important Use of the IP Security feature requires that a valid license key be installed. Contact your local Sales or Support representative for information on how to obtain a license.

The following configuration example enables X.509 certificate-based peer authentication on the S-GW.

In Global Configuration Mode, specify the name of the X.509 certificate and CA certificate, as follows:

```
configure
   certificate name <cert_name> pem url <cert_pem_url> private-key pem url
<private_key_url>
   ca-certificate name <ca_cert_name> pem url <ca_cert_url>
   end
```

Notes:

- The **certificate name** and **ca-certificate list ca-cert-name** commands specify the X.509 certificate and CA certificate to be used.
- The PEM-formatted data for the certificate and CA certificate can be specified, or the information can be read from a file via a specified URL as shown in this example.

When creating the crypto template for IPSec in Context Configuration Mode, bind the X.509 certificate and CA certificate to the crypto template and enable X.509 certificate-based peer authentication for the local and remote nodes, as follows:

```
configure
   context <saegw_context_name>
      crypto template <crypto_template_name> ikev2-dynamic
         certificate name <cert_name>
         ca-certificate list ca-cert-name <ca_cert_name>
         authentication local certificate
         authentication remote certificate
         end
```

Notes:

- A maximum of sixteen certificates and sixteen CA certificates are supported per system. One certificate is supported per service, and a maximum of four CA certificates can be bound to one crypto template.
- The **certificate name** and **ca-certificate list ca-cert-name** commands bind the certificate and CA certificate to the crypto template.
- The **authentication local certificate** and **authentication remote certificate** commands enable X.509 certificate-based peer authentication for the local and remote nodes.
- Service names must be unique across all contexts within a chassis.

Configuring Dynamic Node-to-Node IP Security on the S1-U and S5 Interfaces

The configuration example in this section creates IPSec/IKEv2 dynamic node-to-node tunnel endpoints on the S1-U and S5 interfaces.

Important Use of the IP Security feature requires that a valid license key be installed. Contact your local Sales or Support representative for information on how to obtain a license.

The following configuration examples are included in this section:

Creating and Configuring an IPSec Transform Set

The following example configures an IPSec transform set, which is used to define the security association that determines the protocols used to protect the data on the interface:

```
configure
   context <saegw_context_name>
      ipsec transform-set <ipsec_transform-set_name>
         encryption aes-cbc-128
         group none
         hmac sha1-96
         mode tunnel
         end
```

Notes:

- The encryption algorithm, **aes-cbc-128**, or Advanced Encryption Standard Cipher Block Chaining, is the default algorithm for IPSec transform sets configured on the system.
- The **group none** command specifies that no crypto strength is included and that Perfect Forward Secrecy is disabled. This is the default setting for IPSec transform sets configured on the system.
- The **hmac** command configures the Encapsulating Security Payload (ESP) integrity algorithm. The **sha1-96** keyword uses a 160-bit secret key to produce a 160-bit authenticator value. This is the default setting for IPSec transform sets configured on the system.
- The **mode tunnel** command specifies that the entire packet is to be encapsulated by the IPSec header, including the IP header. This is the default setting for IPSec transform sets configured on the system.
- Service names must be unique across all contexts within a chassis.

Creating and Configuring an IKEv2 Transform Set

The following example configures an IKEv2 transform set:

```
configure
   context <saegw_context_name>
      ikev2-ikesa transform-set <ikev2_transform-set_name>
```

```
encryption aes-cbc-128
group 2
hmac sha1-96
lifetime <sec>
prf sha1
end
```

Notes:

- The encryption algorithm, **aes-cbc-128**, or Advanced Encryption Standard Cipher Block Chaining, is the default algorithm for IKEv2 transform sets configured on the system.
- The **group 2** command specifies the Diffie-Hellman algorithm as Group 2, indicating medium security. The Diffie-Hellman algorithm controls the strength of the crypto exponentials. This is the default setting for IKEv2 transform sets configured on the system.
- The **hmac** command configures the Encapsulating Security Payload (ESP) integrity algorithm. The **sha1-96** keyword uses a 160-bit secret key to produce a 160-bit authenticator value. This is the default setting for IKEv2 transform sets configured on the system.
- The **lifetime** command configures the time the security key is allowed to exist, in seconds.
- The **prf** command configures the IKE Pseudo-random Function, which produces a string of bits that cannot be distinguished from a random bit string without knowledge of the secret key. The **sha1** keyword uses a 160-bit secret key to produce a 160-bit authenticator value. This is the default setting for IKEv2 transform sets configured on the system.
- Service names must be unique across all contexts within a chassis.

Creating and Configuring a Crypto Template

The following example configures an IKEv2 crypto template:

```
configure
   context <saegw_context_name>
      crypto template <crypto_template_name> ikev2-dynamic
         ikev2-ikesa transform-set list <name1> . . . <name6>
         ikev2-ikesa rekey
         payload <name> match childsa match ipv4
            ipsec transform-set list <name1> . . . <name4>
            rekey
            end
```

Notes:

- The **ikev2-ikesa transform-set list** command specifies up to six IKEv2 transform sets.
- The **ipsec transform-set list** command specifies up to four IPSec transform sets.
- Service names must be unique across all contexts within a chassis.

Binding the S1-U and S5 IP Addresses to the Crypto Template

The following example configures the binding of the S1-U and S5 interfaces to the crypto template.

Important If you modify the **interface-type** command, the parent service (service within which the eGTP/GTP-U service is configured) will automatically restart. Service restart results in dropping of active calls associated with the parent service.

```
configure
  context <saegw_context_name>
    gtpu-service <gtpu_ingress_service_name>
      bind ipv4-address <s1-u_interface_ip_address> crypto-template
<enodeb_crypto_template>
      exit
    egtp-service <egtp_ingress_service_name>
      interface-type interface-sgw-ingress
      associate gtpu-service <gtpu_ingress_service_name>
      gtpc bind address <s1u_interface_ip_address>
      exit
    exit
  context <sgw_egress_context_name>
    gtpu-service <gtpu_egress_service_name>
      bind ipv4-address <s5_interface_ip_address> crypto-template
<enodeb_crypto_template>
      exit
    egtp-service <egtp_egress_service_name>
      interface-type interface-sgw-egress
      associate gtpu-service <gtpu_egress_service_name>
      gtpc bind address <s5_interface_ip_address>
      exit
    exit
  context <saegw_context_name>
    sgw-service <sgw_service_name> -noconfirm
      egtp-service ingress service <egtp_ingress_service_name>
      egtp-service egress context <sgw_egress_context_name>
      end
```

Notes:

- The **bind** command in the GTP-U ingress and egress service configuration can also be specified as an IPv6 address using the **ipv6-address** command.
- Service names must be unique across all contexts within a chassis.

Configuring ACL-based Node-to-Node IP Security on the S1-U and S5 Interfaces

The configuration example in this section creates IKEv2/IPSec ACL-based node-to-node tunnel endpoints on the S1-U and S5 interfaces.

Important Use of the IP Security feature requires that a valid license key be installed. Contact your local Sales or Support representative for information on how to obtain a license.

The following configuration examples are included in this section:

Creating and Configuring a Crypto Access Control List

The following example configures a crypto ACL (Access Control List), which defines the matching criteria used for routing subscriber data packets over an IPSec tunnel:

```
configure
   context <saegw_context_name>
      ip access-list <acl_name>
         permit tcp host <source_host_address> host <dest_host_address>
         end
```

Notes:

- The **permit** command in this example routes IPv4 traffic from the server with the specified source host IPv4 address to the server with the specified destination host IPv4 address.

Creating and Configuring an IPSec Transform Set

The following example configures an IPSec transform set which is used to define the security association that determines the protocols used to protect the data on the interface:

```
configure
   context <saegw_context_name>
      ipsec transform-set <ipsec_transform-set_name>
         encryption aes-cbc-128
         group none
         hmac sha1-96
         mode tunnel
         end
```

Notes:

- The encryption algorithm, **aes-cbc-128**, or Advanced Encryption Standard Cipher Block Chaining, is the default algorithm for IPSec transform sets configured on the system.
- The **group none** command specifies that no crypto strength is included and that Perfect Forward Secrecy is disabled. This is the default setting for IPSec transform sets configured on the system.
- The **hmac** command configures the Encapsulating Security Payload (ESP) integrity algorithm. The **sha1-96** keyword uses a 160-bit secret key to produce a 160-bit authenticator value. This is the default setting for IPSec transform sets configured on the system.
- The **mode tunnel** command specifies that the entire packet is to be encapsulated by the IPSec header including the IP header. This is the default setting for IPSec transform sets configured on the system.
- Service names must be unique across all contexts within a chassis.

Creating and Configuring an IKEv2 Transform Set

The following example configures an IKEv2 transform set:

```
configure
   context <saegw_context_name>
      ikev2-ikesa transform-set <ikev2_transform-set_name>
         encryption aes-cbc-128
         group 2
         hmac sha1-96
         lifetime <sec>
         prf sha1
         end
```

Notes:

- The encryption algorithm, **aes-cbc-128**, or Advanced Encryption Standard Cipher Block Chaining, is the default algorithm for IKEv2 transform sets configured on the system.
- The **group 2** command specifies the Diffie-Hellman algorithm as Group 2, indicating medium security. The Diffie-Hellman algorithm controls the strength of the crypto exponentials. This is the default setting for IKEv2 transform sets configured on the system.
- The **hmac** command configures the Encapsulating Security Payload (ESP) integrity algorithm. The **sha1-96** keyword uses a 160-bit secret key to produce a 160-bit authenticator value. This is the default setting for IKEv2 transform sets configured on the system.
- The **lifetime** command configures the time the security key is allowed to exist, in seconds.
- The **prf** command configures the IKE Pseudo-random Function which produces a string of bits that cannot be distinguished from a random bit string without knowledge of the secret key. The **sha1** keyword uses a 160-bit secret key to produce a 160-bit authenticator value. This is the default setting for IKEv2 transform sets configured on the system.
- Service names must be unique across all contexts within a chassis.

Creating and Configuring a Crypto Map

The following example configures an IKEv2 crypto map and applies it to the S1-U interface:

```
configure
   context <saegw_context_name>
      crypto map <crypto_map_name> ikev2-ipv4
         match address <acl_name>
         peer <ipv4_address>
         authentication local pre-shared-key key <text>
         authentication remote pre-shared-key key <text>
         ikev2-ikesa transform-set list <name1> . . . <name6>
         payload <name> match ipv4
            lifetime <seconds>
            ipsec transform-set list <name1> . . . <name4>
            exit
         exit
      interface <s1-u_intf_name>
         ip address <ipv4_address>
```

```
      crypto-map <crypto_map_name>
    exit
  exit
  port ethernet <slot_number/port_number>
    no shutdown
    bind interface <s1_u_intf_name> <saegw_context_name>
    end
```

Notes:

- The type of crypto map used in this example is IKEv2-IPv4 for IPv4 addressing. An IKEv2-IPv6 crypto map can also be used for IPv6 addressing.
- The **ipsec transform-set list** command specifies up to four IPSec transform sets.
- Service names must be unique across all contexts within a chassis.

The following example configures an IKEv2 crypto map and applies it to the S5 interface:

```
configure
  context <sgw_egress_context_name>
    crypto map <crypto_map_name> ikev2-ipv4
      match address <acl_name>
      peer <ipv4_address>
      authentication local pre-shared-key key <text>
      authentication remote pre-shared-key key <text>
      payload <name> match ipv4
        lifetime <seconds>
        ipsec transform-set list <name1> . . . <name4>
        exit
      exit
    interface <s5_intf_name>
      ip address <ipv4_address>
        crypto map <crypto_map_name>
        exit
      exit
    port ethernet <slot_number/port_number>
      no shutdown
      bind interface <s5_intf_name> <sgw_egress_context_name>
      end
```

Notes:

- The type of crypto map used in this example is IKEv2-IPv4 for IPv4 addressing. An IKEv2-IPv6 crypto map can also be used for IPv6 addressing.
- The **ipsec transform-set list** command specifies up to four IPSec transform sets.
- Service names must be unique across all contexts within a chassis.

Configuring R12 Load Control Support

Load control enables a GTP-C entity (for example, an S-GW/P-GW) to send its load information to a GTP-C peer (e.g. an MME/SGSN, ePDG, TWAN) to adaptively balance the session load across entities supporting

the same function (for example, an S-GW cluster) according to their effective load. The load information reflects the operating status of the resources of the GTP-C entity.

Use the following example to configure this feature:

```
configure
  gtpc-load-control-profile profile_name
    inclusion-frequency advertisement-interval interval_in_seconds
    weightage system-cpu-utilization percentage system-memory-utilization
 percentage license-session-utilization percentage
    end
configure
  context context_name
    sgw-service sgw_service_name
      associate gtpc-load-control-profile profile_name
      exit
    saegw-service saegw_service_name
      associate sgw-service sgw_service_name
      end
```

Notes:

- The **inclusion-frequency** parameter determines how often the Load control information element is sent to the peer(s).
- The total of the three **weightage** parameters should not exceed 100.
- The **associate** command is used to associate the Load Control Profile with an existing S-GW service and to associate the S-GW service with the SAEGW service.
- On the SAEGW, both the P-GW and S-GW should use the same Load Control profile.

Configuring R12 Overload Control Support

Overload control enables a GTP-C entity becoming or being overloaded to gracefully reduce its incoming signaling load by instructing its GTP-C peers to reduce sending traffic according to its available signaling capacity to successfully process the traffic. A GTP-C entity is in overload when it operates over its signaling capacity, which results in diminished performance (including impacts to handling of incoming and outgoing traffic).

Use the following example to configure this feature.

```
configure
  gtpc-overload-control-profile profile_name
    inclusion-frequency advertisement-interval interval_in_seconds
    weightage system-cpu-utilization percentage system-memory-utilization
 percentage license-session-utilization percentage
    throttling-behavior emergency-events exclude
    tolerance initial-reduction-metric percentage
    tolerance threshold report-reduction-metric percentage
self-protection-limit percentage
    validity-period seconds
    end
configure
  context context_name
    sgw-service sgw_service_name
```

```
      associate gtpc-overload-control-profile profile_name
      exit
   saegw-service saegw_service_name
      associate sgw-service sgw_service_name
      end
```

Notes:

- The **inclusion-frequency** parameter determines how often the Overload control information element is sent to the peer(s).
- The total of the three **weightage** parameters should not exceed 100.
- **validity-period** configures how long the overload control information is valid. Valid entries are from 1 to 3600 seconds. The default is 600 seconds.
- The **associate** command is used to associate the Overload Control Profile with an existing S-GW and SAEGW service.
- On the SAEGW, both the P-GW and S-GW should use the same Overload Control profile.

Configuring S4 SGSN Handover Capability

This configuration example configures an S4 interface supporting inter-RAT handovers between the S-GW and a S4 SGSN. Use the following example to configure this feature.

Important If you modify the **interface-type** command, the parent service (service within which the eGTP/GTP-U service is configured) will automatically restart. Service restart results in dropping of active calls associated with the parent service.

```
configure
   context <saegw_context_name> -noconfirm
      interface <s4_interface_name>
         ip address <ipv4_address_primary>
         ip address <ipv4_address_secondary>
         exit
      exit
   port ethernet <slot_number/port_number>
      no shutdown
      bind interface <s4_interface_name> <saegw_context_name>
      exit
   context <saegw_context_name> -noconfirm
      gtpu-service <s4_gtpu_ingress_service_name>
         bind ipv4-address <s4_interface_ip_address>
         exit
      egtp-service <s4_egtp_ingress_service_name>
         interface-type interface-sgw-ingress
         validation-mode default
         associate gtpu-service <s4_gtpu_ingress_service_name>
         gtpc bind address <s4_interface_ip_address>
         exit
      sgw-service <sgw_service_name> -noconfirm
```

```
associate ingress egtp-service <s4_egtp_ingress_service_name>
end
```

Notes:

- The S4 interface IP address(es) can also be specified as IPv6 addresses using the **ipv6 address** command.
- Service names must be unique across all contexts within a chassis.

Configuring an eGTP P-GW Service

This section provides a high-level series of steps and the associated configuration file examples for configuring the system to perform as an eGTP P-GW in a test environment. Information provided in this section includes the following:

Information Required

The following sections describe the minimum amount of information required to configure and make the P-GW operational on the network. To make the process more efficient, it is recommended that this information be available prior to configuring the system.

There are additional configuration parameters that are not described in this section. These parameters deal mostly with fine-tuning the operation of the P-GW in the network. Information on these parameters can be found in the appropriate sections of the *Command Line Interface Reference*.

Required P-GW Context Configuration Information

The following table lists the information that is required to configure the P-GW context on a P-GW.

Table 5: Required Information for P-GW (SAEGW) Context Configuration

Required Information	Description
P-GW context name	An identification string from 1 to 79 characters (alpha and/or numeric) by which the P-GW context will be recognized by the system. **Important** The P-GW service must reside within the SAEGW context, thus this would be the SAEGW context name.
Accounting policy name	An identification string from 1 to 63 characters (alpha and/or numeric) by which the accounting policy will be recognized by the system. The accounting policy is used to set parameters for the Rf (off-line charging) interface.

Required Information	Description
S5/S8 Interface Configuration (To/from S-GW)	
Interface name	An identification string between 1 and 79 characters (alpha and/or numeric) by which the interface will be recognized by the system. Multiple names are needed if multiple interfaces will be configured.
IP address and subnet	IPv4 or IPv6 addresses assigned to the interface. Multiple addresses and subnets are needed if multiple interfaces will be configured.
Physical port number	The physical port to which the interface will be bound. Ports are identified by the chassis slot number where the line card resides followed by the number of the physical connector on the card. For example, port 17/1 identifies connector number 1 on the card in slot 17. A single physical port can facilitate multiple interfaces.
Gateway IP address	Used when configuring static IP routes from the interface(s) to a specific network.
GTP-U Service Configuration	
GTP-U service name	An identification string from 1 to 63 characters (alpha and/or numeric) by which the GTP-U service will be recognized by the system.
IP address	S5/S8 interface IPv4 address.
P-GW Service Configuration	
P-GW service name	An identification string from 1 to 63 characters (alpha and/or numeric) by which the P-GW service will be recognized by the system. Multiple names are needed if multiple P-GW services will be used.
PLMN ID	MCC number: The mobile country code (MCC) portion of the PLMN's identifier (an integer value between 100 and 999). MNC number: The mobile network code (MNC) portion of the PLMN's identifier (a 2 or 3 digit integer value between 00 and 999).
eGTP Service Configuration	
eGTP Service Name	An identification string from 1 to 63 characters (alpha and/or numeric) by which the eGTP service will be recognized by the system.

Required PDN Context Configuration Information

The following table lists the information that is required to configure the PDN context on a P-GW.

Table 6: Required Information for PDN Context Configuration

Required Information	Description
PDN context name	An identification string from 1 to 79 characters (alpha and/or numeric) by which the PDN context is recognized by the system.
IP Address Pool Configuration	
IPv4 address pool name and range	An identification string between 1 and 31 characters (alpha and/or numeric) by which the IPv4 pool is recognized by the system. Multiple names are needed if multiple pools will be configured. A range of IPv4 addresses defined by a starting address and an ending address.
IPv6 address pool name and range	An identification string between 1 and 31 characters (alpha and/or numeric) by which the IPv6 pool is recognized by the system. Multiple names are needed if multiple pools will be configured. A range of IPv6 addresses defined by a starting address and an ending address.
Access Control List Configuration	
IPv4 access list name	An identification string between 1 and 47 characters (alpha and/or numeric) by which the IPv4 access list is recognized by the system. Multiple names are needed if multiple lists will be configured.
IPv6 access list name	An identification string between 1 and 79 characters (alpha and/or numeric) by which the IPv6 access list is recognized by the system. Multiple names are needed if multiple lists will be configured.
Deny/permit type	The types are: • any • by host IP address • by IP packets • by source ICMP packets • by source IP address masking • by TCP/UDP packets
Readdress or redirect type	The types are • readdress server • redirect context • redirect css delivery-sequence • redirect css service • redirect nexthop

Required Information	Description
SGi Interface Configuration (To/from IPv4 PDN)	
Interface name	An identification string between 1 and 79 characters (alpha and/or numeric) by which the interface is recognized by the system. Multiple names are needed if multiple interfaces will be configured.
IP address and subnet	IPv4 addresses assigned to the interface. Multiple addresses and subnets are needed if multiple interfaces will be configured.
Physical port number	The physical port to which the interface will be bound. Ports are identified by the chassis slot number where the line card resides followed by the number of the physical connector on the card. For example, port 17/1 identifies connector number 1 on the card in slot 17. A single physical port can facilitate multiple interfaces.
Gateway IP address	Used when configuring static IP routes from the interface(s) to a specific network.
SGi Interface Configuration (To/from IPv6 PDN)	
Interface name	An identification string between 1 and 79 characters (alpha and/or numeric) by which the interface is recognized by the system. Multiple names are needed if multiple interfaces will be configured.
IP address and subnet	IPv6 addresses assigned to the interface. Multiple addresses and subnets are needed if multiple interfaces will be configured.
Physical port number	The physical port to which the interface will be bound. Ports are identified by the chassis slot number where the line card resides followed by the number of the physical connector on the card. For example, port 17/1 identifies connector number 1 on the card in slot 17. A single physical port can facilitate multiple interfaces.
Gateway IP address	Used when configuring static IP routes from the interface(s) to a specific network.

Required AAA Context Configuration Information

The following table lists the information that is required to configure the AAA context on a P-GW.

Table 7: Required Information for AAA Context Configuration

Required Information	Description
Gx Interface Configuration (to PCRF)	

Required Information	Description
Interface name	An identification string between 1 and 79 characters (alpha and/or numeric) by which the interface is recognized by the system. Multiple names are needed if multiple interfaces will be configured.
IP address and subnet	IPv4 or IPv6 addresses assigned to the interface. Multiple addresses and subnets are needed if multiple interfaces will be configured.
Physical port number	The physical port to which the interface will be bound. Ports are identified by the chassis slot number where the line card resides followed by the number of the physical connector on the card. For example, port 17/1 identifies connector number 1 on the card in slot 17. A single physical port can facilitate multiple interfaces.
Gateway IP address	Used when configuring static IP routes from the interface(s) to a specific network.
Gx Diameter Endpoint Configuration	
End point name	An identification string from 1 to 63 characters (alpha and/or numeric) by which the Gx Diameter endpoint configuration is recognized by the system.
Origin realm name	An identification string between 1 through 127 characters. The realm is the Diameter identity. The originator's realm is present in all Diameter messages and is typically the company or service name.
Origin host name	An identification string from 1 to 255 characters (alpha and/or numeric) by which the Gx origin host is recognized by the system.
Origin host address	The IP address of the Gx interface.
Peer name	The Gx endpoint name described above.
Peer realm name	The Gx origin realm name described above.
Peer address and port number	The IP address and port number of the PCRF.
Route-entry peer	The Gx endpoint name described above.
Gy Interface Configuration (to on-line charging server)	
Interface name	An identification string between 1 and 79 characters (alpha and/or numeric) by which the interface is recognized by the system. Multiple names are needed if multiple interfaces will be configured.
IP address and subnet	IPv4 or IPv6 addresses assigned to the interface. Multiple addresses and subnets are needed if multiple interfaces will be configured.

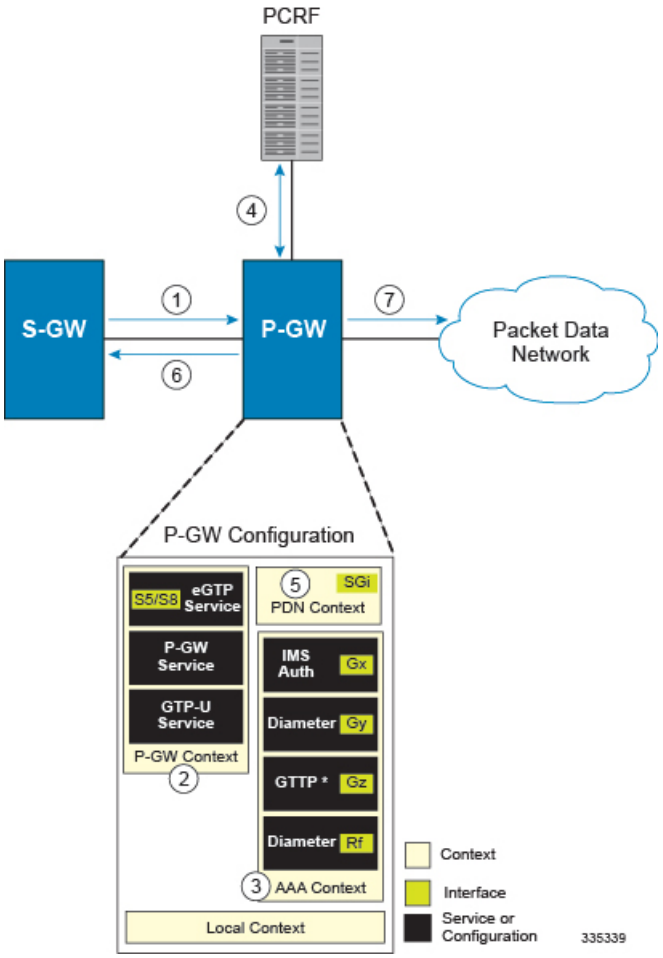
Required Information	Description
Physical port number	The physical port to which the interface will be bound. Ports are identified by the chassis slot number where the line card resides followed by the number of the physical connector on the card. For example, port 17/1 identifies connector number 1 on the card in slot 17. A single physical port can facilitate multiple interfaces.
Gateway IP address	Used when configuring static IP routes from the interface(s) to a specific network.
Gy Diameter Endpoint Configuration	
End point name	An identification string from 1 to 63 characters (alpha and/or numeric) by which the Gy Diameter endpoint configuration is recognized by the system.
Origin realm name	An identification string between 1 through 127 characters. The realm is the Diameter identity. The originator's realm is present in all Diameter messages and is typically the company or service name.
Origin host name	An identification string from 1 to 255 characters (alpha and/or numeric) by which the Gy origin host is recognized by the system.
Origin host address	The IP address of the Gy interface.
Peer name	The Gy endpoint name described above.
Peer realm name	The Gy origin realm name described above.
Peer address and port number	The IP address and port number of the OCS.
Route-entry peer	The Gy endpoint name described above.
Gz Interface Configuration (to off-line charging server)	
Interface name	An identification string between 1 and 79 characters (alpha and/or numeric) by which the interface is recognized by the system. Multiple names are needed if multiple interfaces will be configured.
IP address and subnet	IPv4 addresses assigned to the interface. Multiple addresses and subnets are needed if multiple interfaces will be configured.
Physical port number	The physical port to which the interface will be bound. Ports are identified by the chassis slot number where the line card resides followed by the number of the physical connector on the card. For example, port 17/1 identifies connector number 1 on the card in slot 17. A single physical port can facilitate multiple interfaces.
Gateway IP address	Used when configuring static IP routes from the interface(s) to a specific network.
Rf Interface Configuration (to off-line charging server)	

Required Information	Description
Interface name	An identification string between 1 and 79 characters (alpha and/or numeric) by which the interface is recognized by the system. Multiple names are needed if multiple interfaces will be configured.
IP address and subnet	IPv4 or IPv6 addresses assigned to the interface. Multiple addresses and subnets are needed if multiple interfaces will be configured.
Physical port number	The physical port to which the interface will be bound. Ports are identified by the chassis slot number where the line card resides followed by the number of the physical connector on the card. For example, port 17/1 identifies connector number 1 on the card in slot 17. A single physical port can facilitate multiple interfaces.
Gateway IP address	Used when configuring static IP routes from the interface(s) to a specific network.
Rf Diameter Endpoint Configuration	
End point name	An identification string from 1 to 63 characters (alpha and/or numeric) by which the Rf Diameter endpoint configuration is recognized by the system.
Origin realm name	An identification string between 1 through 127 characters. The realm is the Diameter identity. The originator's realm is present in all Diameter messages and is typically the company or service name.
Origin host name	An identification string from 1 to 255 characters (alpha and/or numeric) by which the Rf origin host is recognized by the system.
Origin host address	The IP address of the Rf interface.
Peer name	The Rf endpoint name described above.
Peer realm name	The Rf origin realm name described above.
Peer address and port number	The IP address and port number of the OFCS.
Route-entry peer	The Rf endpoint name described above.

How This Configuration Works

The following figure and supporting text describe how this configuration with a single source and destination context is used by the system to process a subscriber call originating from the GTP LTE network.

Figure 7: SAEGW Configuration with Single Source and Destination Context Processing a Subscriber Call Originating from the GTP LTE Network

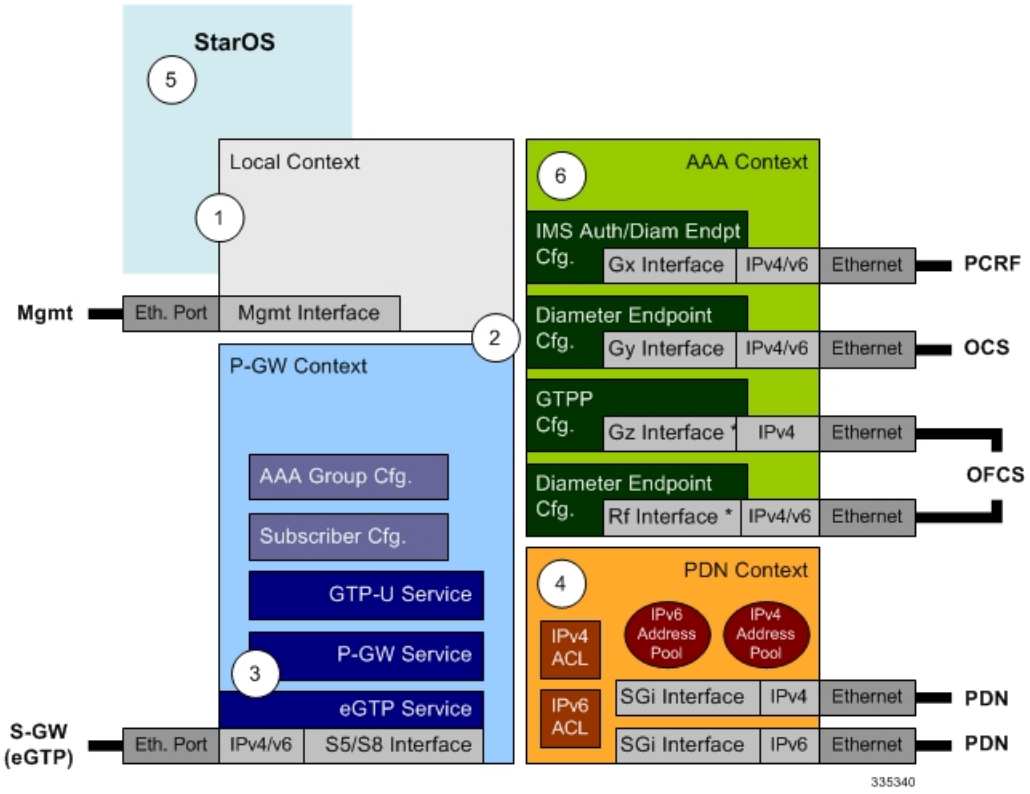


1. The S-GW establishes the S5/S8 connection by sending a Create Session Request message to the P-GW including an Access Point name (APN).
2. The P-GW service determines which context to use to provide AAA functionality for the session. This process is described in the *How the System Selects Contexts* section located in the *Understanding the System Operation and Configuration* chapter of the *System Administration Guide*.
3. The P-GW uses the configured Gx Diameter endpoint to establish the IP-CAN session.
4. The P-GW sends a CC-Request (CCR) message to the PCRF to indicate the establishment of the IP-CAN session and the PCRF acknowledges with a CC-Answer (CCA).
5. The P-GW uses the APN configuration to select the PDN context. IP addresses are assigned from the IP pool configured in the selected PDN context.
6. The P-GW responds to the S-GW with a Create Session Response message including the assigned address and additional information.
7. The S5/S8 data plane tunnel is established and the P-GW can forward and receive packets to/from the PDN.

eGTP P-GW Configuration

To configure the system to perform as an eGTP P-GW:

Figure 8: eGTP P-GW Configuration



Step 1 Set system configuration parameters such as activating PSCs by applying the example configurations found in the *System Administration Guide*.

Step 2 Set initial configuration parameters such as creating contexts and services by applying the example configurations found in Initial Configuration, on page 40.

Step 3 Configure the system to perform as an eGTP P-GW and set basic P-GW parameters such as eGTP interfaces and IP routes by applying the example configurations presented in P-GW Service Configuration, on page 44.

Step 4 Configure the PDN context by applying the example configuration in P-GW PDN Context Configuration, on page 44.

Step 5 Enable and configure the active charging service for Gx interface support by applying the example configuration in Active Charging Service Configuration, on page 45.

Step 6 Create a AAA context and configure parameters for policy by applying the example configuration in Policy Configuration, on page 47.

Step 7 Verify and save the configuration by following the steps found in Verifying and Saving the Configuration, on page 49.

Initial Configuration

Step 1 Create the context where the eGTP service will reside by applying the example configuration in Creating and Configuring an eGTP P-GW Context, on page 40.

Step 2 Create and configure APNs in the P-GW context by applying the example configuration in Creating and Configuring APNs in the P-GW Context, on page 41.

Step 3 Create and configure AAA server groups in the P-GW context by applying the example configuration in Creating and Configuring AAA Groups in the P-GW Context, on page 42.

Step 4 Create an eGTP service within the newly created context by applying the example configuration in Creating and Configuring an eGTP Service, on page 42.

Step 5 Create and configure a GTP-U service within the P-GW context by applying the example configuration in Creating and Configuring a GTP-U Service, on page 43.

Step 6 Create a context through which the interface to the PDN will reside by applying the example configuration in Creating a P-GW PDN Context, on page 43.

Creating and Configuring an eGTP P-GW Context

Use the following example to create a P-GW context, create an S5/S8 IPv4 interface (for data traffic to/from the S-GW), and bind the S5/S8 interface to a configured Ethernet port:

```
configure
   gtpp single-source
   context <saegw_context_name> -noconfirm
      interface <s5s8_interface_name>
         ip address <ipv4_address>
         exit
      gtpp group default
         gtpp charging-agent address <gz_ipv4_address>
         gtpp echo-interval <seconds>
         gtpp attribute diagnostics
         gtpp attribute local-record-sequence-number
         gtpp attribute node-id-suffix <string>
         gtpp dictionary <name>
         gtpp server <ipv4_address> priority <num>
         gtpp server <ipv4_address> priority <num> node-alive enable
         exit
      policy accounting <rf_policy_name> -noconfirm
         accounting-level {level_type}
         accounting-event-trigger interim-timeout action stop-start
         operator-string <string>
         cc profile <index> interval <seconds>
         exit
      exit
   subscriber default
   exit
   port ethernet <slot_number/port_number>
      no shutdown
```

```
bind interface <s5s8_interface_name> <saegw_context_name>
end
```

Notes:

- **gtpp single-source** is enabled to allow the system to generate requests to the accounting server using a single UDP port (by way of a AAA proxy function) rather than each AAA manager generating requests on unique UDP ports.
- The S5/S8 (P-GW to S-GW) interface IP address can also be specified as an IPv6 address using the **ipv6 address** command.
- Set the accounting policy for the Rf (off-line charging) interface. The accounting level types are: flow, PDN, PDN-QCI, QCI, and subscriber. Refer to the *Accounting Profile Configuration Mode Commands* chapter in the *Command Line Interface Reference* for more information on this command.
- Set the GTPP group setting for Gz accounting.
- Service names must be unique across all contexts within a chassis.

Creating and Configuring APNs in the P-GW Context

Use the following configuration to create an APN:

```
configure
  context <saegw_context_name> -noconfirm
    apn <name>
      accounting-mode radius-diameter
      associate accounting-policy <rf_policy_name>
      ims-auth-service <gx_ims_service_name>
      aaa group <rf-radius_group_name>
      dns primary <ipv4_address>
      dns secondary <ipv4_address>
      ip access-group <name> in
      ip access-group <name> out
      mediation-device context-name <saegw_context_name>
      ip context-name <pdn_context_name>
      ipv6 access-group <name> in
      ipv6 access-group <name> out
      active-charging rulebase <name>
      end
```

Notes:

- The IMS Authorization Service is created and configured in the AAA context.
- Multiple APNs can be configured to support different domain names.
- The associate accounting-policy command is used to associate a pre-configured accounting policy with this APN. Accounting policies are configured in the P-GW context. An example is located in Creating and Configuring an eGTP P-GW Context, on page 40.
- Service names must be unique across all contexts within a chassis.

Use the following configuration to create an APN that includes Gz interface parameters:

```
configure
  context <saegw_context_name> -noconfirm
```

```
apn <name>
  bearer-control-mode mixed
  selection-mode sent-by-ms
  accounting-mode gtpp
  gtpp group default accounting-context <aaa_context_name>
  ims-auth-service <gx_ims_service_name>
  ip access-group <name> in
  ip access-group <name> out
  ip context-name <pdn_context_name>
  active-charging rulebase <gz_rulebase_name>
  end
```

Notes:

- The IMS Authorization Service is created and configured in the AAA context.
- Multiple APNs can be configured to support different domain names.
- The accounting-mode GTPP and GTPP group commands configure this APN for Gz accounting.
- Service names must be unique across all contexts within a chassis.

Creating and Configuring AAA Groups in the P-GW Context

Use the following example to create and configure AAA groups supporting RADIUS and Rf accounting:

```
configure
  context <saegw_context_name> -noconfirm
    aaa group <rf-radius_group_name>
      radius attribute nas-identifier <id>
      radius accounting interim interval <seconds>
      radius dictionary <name>
      radius mediation-device accounting server <address> key <key>
      diameter authentication dictionary <name>
      diameter accounting dictionary <name>
      diameter accounting endpoint <rf_cfg_name>
      diameter accounting server <rf_cfg_name> priority <num>
      exit
    aaa group default
      radius attribute nas-ip-address address <ipv4_address>
      radius accounting interim interval <seconds>
      diameter authentication dictionary <name>
      diameter accounting dictionary <name>
      diameter accounting endpoint <rf_cfg_name>
      diameter accounting server <rf_cfg_name> priority <num>
      end
```

Notes:

- Service names must be unique across all contexts within a chassis.

Creating and Configuring an eGTP Service

Use the following configuration example to create the eGTP service.

Important If you modify the **interface-type** command, the parent service (service within which the eGTP/GTP-U service is configured) will automatically restart. Service restart results in dropping of active calls associated with the parent service.

```
configure
  context <saegw_context_name>
    egtp-service <egtp_service_name> -noconfirm
      interface-type interface-pgw-ingress
      validation mode default
      associate gtpu-service <gtpu_service_name>
      gtpc bind address <s5s8_interface_address>
      end
```

Notes:

- Co-locating a GGSN service on the same ASR 5500 requires that the **gtpc bind address** command uses the same IP address that the GGSN service is bound to.
- Service names must be unique across all contexts within a chassis.

Creating and Configuring a GTP-U Service

Use the following configuration example to create the GTP-U service:

```
configure
  context <saegw_context_name>
    gtpu-service <gtpu_service_name> -noconfirm
      bind ipv4-address <s5s8_interface_address>
      end
```

Notes:

- The **bind** command can also be specified as an IPv6 address using the **ipv6-address** command.
- Service names must be unique across all contexts within a chassis.

Creating a P-GW PDN Context

Use the following example to create a P-GW PDN context and Ethernet interface, and bind the interface to a configured Ethernet port.

```
configure
  context <pdn_context_name> -noconfirm
    interface <sgi_ipv4_interface_name>
      ip address <ipv4_address>
      exit
    interface <sgi_ipv6_interface_name>
      ipv6 address <address>
      end
```

Notes:

- Service names must be unique across all contexts within a chassis.

P-GW Service Configuration

Step 1 Configure the P-GW service by applying the example configuration in Configuring the P-GW Service, on page 44.

Step 2 Specify an IP route to the eGTP Serving Gateway by applying the example configuration in Configuring a Static IP Route, on page 44.

Configuring the P-GW Service

Use the following example to configure the P-GW service:

```
configure
   context <saegw_context_name>
      pgw-service <pgw_service_name> -noconfirm
         plmn id mcc <id> mnc <id>
         associate egtp-service <egtp_service_name>
         associate qci-qos-mapping <name>
         end
```

Notes:

- QCI-QoS mapping configurations are created in the AAA context. Refer to Configuring QCI-QoS Mapping, on page 49.
- Co-locating a GGSN service on the same ASR 5500 requires the configuration of the **associate ggsn-service** *name* command within the P-GW service.
- Service names must be unique across all contexts within a chassis.

Configuring a Static IP Route

Use the following example to configure an IP Route for control and user plane data communication with an eGTP Serving Gateway:

```
configure
   context <saegw_context_name>
      ip route <sgw_ip_addr/mask> <sgw_next_hop_addr> <pgw_intrfc_name>
      end
```

Notes:

- Service names must be unique across all contexts within a chassis.

P-GW PDN Context Configuration

Use the following example to configure an IP Pool and APN, and bind a port to the interface in the PDN context:

```
configure
   context <pdn_context_name> -noconfirm
      interface <sgi_ipv4_interface_name>
         ip address <ipv4_address>
         exit
      interface <sgi_ipv6_interface_name>
         ip address <ipv6_address>
```

```
    exit
  ip pool <name> range <start_address end_address> public <priority>
  ipv6 pool <name> range <start_address end_address> public <priority>
  subscriber default
  exit
  ip access-list <name>
    redirect css service <name> any
    permit any
    exit
  ipv6 access-list <name>
    redirect css service <name> any
    permit any
    exit
  aaa group default
  exit
exit
port ethernet <slot_number/port_number>
  no shutdown
  bind interface <sgi_ipv4_interface_name> <pdn_context_name>
  exit
port ethernet <slot_number/port_number>
  no shutdown
  bind interface <sgi_ipv6_interface_name> <pdn_context_name>
  end
```

Notes:

- Service names must be unique across all contexts within a chassis.

Active Charging Service Configuration

Use the following example to enable and configure active charging:

```
configure
  require active-charging optimized-mode
  active-charging service <name>
    ruledef <name>
      <rule_definition>
              .
              .
      <rule_definition>
      exit
    ruledef default
      ip any-match = TRUE
      exit
    ruledef icmp-pkts
      icmp any-match = TRUE
      exit
    ruledef qci3
      icmp any-match = TRUE
      exit
    ruledef static
      icmp any-match = TRUE
      exit
```

```
      charging-action <name>
        <action>
              .
              .
        <action>
        exit
      charging-action icmp
        billing-action egcdr
        exit
      charging-action qci3
        content-id <id>
        billing-action egcdr
        qos-class-identifier <id>
        allocation-retention-priority <priority>
        tft-packet-filter qci3
        exit
      charging-action static
        service-identifier <id>
        billing-action egcdr
        qos-class-identifier <id>
        allocation-retention-priority <priority>
        tft-packet-filter qci3
        exit
      packet-filter <packet_filter_name>
        ip remote address = { ipv4/ipv6_address | ipv4/ipv6_address/mask }
        ip remote-port = { port_number | range start_port_number to end_port_number
}
        exit
        rulebase default
        exit
      rulebase <name>
        <rule_base>
              .
              .
        <rule_base>
        exit
      rulebase <gx_rulebase_name>
        dynamic-rule order first-if-tied
        egcdr tariff minute <minute> hour <hour>(optional)
        billing-records egcdr
        action priority 5 dynamic-only ruledef qci3 charging-action qci3
        action priority 100 ruledef static charging-action static
        action priority 500 ruledef default charging-action icmp
        action priority 570 ruledef icmp-pkts charging-action icmp
        egcdr threshold interval <interval>
        egcdr threshold volume total <bytes>
        end
```

Notes:

- A rule base is a collection of rule definitions and associated charging actions.
- As depicted above, multiple rule definitions, charging actions, and rule bases can be configured to support a variety of charging scenarios.

- Charging actions define the action to take when a rule definition is matched.
- Routing and/or charging rule definitions can be created/configured. The maximum number of routing rule definitions that can be created is 256. The maximum number of charging rule definitions is 2048.
- The billing-action egcdr command in the charging-action *qc13*, *icmp*, and *static* examples is required for Gz accounting.
- The Gz rulebase example supports the Gz interface for off-line charging. The **billing-records egcdr** command is required for Gz accounting. All other commands are optional.
- Service names must be unique across all contexts within a chassis.

Important If an uplink packet is coming on the dedicated bearer, only rules installed on the dedicated bearer are matched. Static rules are not matched and packets failing to match the same will be dropped.

Important After you configure **configure**, **require active-charging optimized-mode**, **active-charging service** <*name*>, **ruledef** <*name*>, and <*rule_definition*> CLI commands, you must save the configuration and then reload the chassis for the command to take effect. For information on saving the configuration file and reloading the chassis, refer to the *System Administration Guide* for your deployment.

Policy Configuration

Step 1 Configure the policy and accounting interfaces by applying the example configuration in Creating and Configuring the AAA Context, on page 47.

Step 2 Create and configure QCI to QoS mapping by applying the example configuration in Configuring QCI-QoS Mapping, on page 49.

Creating and Configuring the AAA Context

Use the following example to create and configure a AAA context including diameter support and policy control, and bind Ethernet ports to interfaces supporting traffic between this context and a PCRF, an OCS, and an OFCS:

```
configure
  context <aaa_context_name> -noconfirm
    interface <gx_interface_name>
      ipv6 address <address>
      exit
    interface <gy_interface_name>
      ipv6 address <address>
      exit
    interface <gz_interface_name>
      ip address <ipv4_address>
      exit
    interface <rf_interface_name>
```

```
      ip address <ipv4_address>
      exit
    subscriber default
      exit
    ims-auth-service <gx_ims_service_name>
      p-cscf discovery table <> algorithm round-robin
      p-cscf table <#> row-precedence <> ipv6-address <pcrf_ipv6_adr>
      policy-control
        diameter origin endpoint <gx_cfg_name>
        diameter dictionary <name>
        diameter host-select table <> algorithm round-robin
        diameter host-select row-precedence <> table <> host <gx_cfg_name>
        exit
      exit
    diameter endpoint <gx_cfg_name>
      origin realm <realm_name>
      origin host <name> address <aaa_ctx_ipv6_address>
      peer <gx_cfg_name> realm <name> address <pcrf_ipv4_or_ipv6_addr>
      route-entry peer <gx_cfg_name>
      exit
    diameter endpoint <gy_cfg_name>
      origin realm <realm_name>
      origin host <name> address <gy_ipv6_address>
      connection retry-timeout <seconds>
      peer <gy_cfg_name> realm <name> address <ocs_ipv4_or_ipv6_addr>
      route-entry peer <gy_cfg_name>
      exit
    diameter endpoint <rf_cfg_name>
      use-proxy
      origin realm <realm_name>
      origin host <name> address <rf_ipv4_address>
      peer <rf_cfg_name> realm <name> address <ofcs_ipv4_or_ipv6_addr>
      route-entry peer <rf_cfg_name>
      exit
    exit
  port ethernet <slot_number/port_number>
    no shutdown
    bind interface <gx_interface_name> <aaa_context_name>
    exit
  port ethernet <slot_number/port_number>
    no shutdown
    bind interface <gy_interface_name> <aaa_context_name>
    exit
  port ethernet <slot_number/port_number>
    no shutdown
    bind interface <gz_interface_name> <aaa_context_name>
    exit
  port ethernet <slot_number/port_number>
    no shutdown
    bind interface <rf_interface_name> <aaa_context_name>
    end
```

Notes:

- The **p-cscf table** command under **ims-auth-service** can also specify an IPv4 address to the PCRF.
- The Gx interface IP address can also be specified as an IPv4 address using the **ip address** command.
- The Gy interface IP address can also be specified as an IPv4 address using the **ip address** command.
- The Rf interface IP address can also be specified as an IPv6 address using the **ipv6 address** command.
- Service names must be unique across all contexts within a chassis.

Configuring QCI-QoS Mapping

Use the following example to create and map QCI values to enforceable QoS parameters:

```
configure
  qci-qos-mapping <name>
    qci 1 user-datagram dscp-marking <hex>
    qci 3 user-datagram dscp-marking <hex>
    qci 9 user-datagram dscp-marking <hex>
    end
```

Notes:

- The SAEGW does not support non-standard QCI values. QCI values 1 through 9 are standard values and are defined in 3GPP TS 23.203; the SAEGW supports these standard values.
- The above configuration only shows one keyword example. Refer to the *QCI - QOS Mapping Configuration Mode Commands* chapter in the *Command Line Interface Reference* for more information on the **qci** command and other supported keywords.

Verifying and Saving the Configuration

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*.

DHCP Service Configuration

The system can be configured to use the Dynamic Host Control Protocol (DHCP) to assign IP addresses for PDP contexts. IP address assignment using DHCP is done using the following method, as configured within an APN:

DHCP-proxy: The system acts as a proxy for client (MS) and initiates the DHCP Discovery Request on behalf of client (MS). Once it receives an allocated IP address from DHCP server in response to DHCP Discovery Request, it assigns the received IP address to the MS. This allocated address must be matched with the an address configured in an IP address pool on the system. This complete procedure is not visible to MS.

As the number of addresses in memory decreases, the system solicits additional addresses from the DHCP server. If the number of addresses stored in memory rises above the configured limit, they are released back to the DHCP server.

There are parameters that must first be configured that specify the DHCP servers to communicate with and how the IP address are handled. These parameters are configured as part of a DHCP service.

Important This section provides the minimum instruction set for configuring a DHCP service on system for DHCP-based IP allocation. For more information on commands that configure additional DHCP server parameters and working of these commands, refer to the *DHCP Service Configuration Mode Commands* chapter of *Command Line Interface Reference*.

These instructions assume that you have already configured the system level configuration as described in *System Administration Guide* and P-GW service as described in *eGTP P-GW Configuration* section of this chapter.

To configure the DHCP service:

Step 1 Create the DHCP service in system context and bind it by applying the example configuration in DHCP Service Creation, on page 50.

Step 2 Configure the DHCP servers and minimum and maximum allowable lease times that are accepted in responses from DHCP servers by applying the example configuration in DHCP Server Parameter Configuration, on page 50.

Step 3 Verify your DHCP Service configuration by following the steps in DHCP Service Configuration Verification, on page 51.

Step 4 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

DHCP Service Creation

Use the following example to create the DHCP service to support DHCP-based address assignment:

```
configure
   context <dest_ctxt_name>
      dhcp-service <dhcp_svc_name>
       bind address <ip_address> [ nexthop-forwarding-address <nexthop_ip_address>
 [ mpls-label input <in_mpls_label_value> output <out_mpls_label_value1>
[out_mpls_label_value2]]]
         end
```

Notes:

- To ensure proper operation, DHCP functionality should be configured within a destination context.
- Optional keyword **nexthop-forwarding-address** <*nexthop_ip_address*> **[mpls-label input** <*in_mpls_label_value*> **output** <*out_mpls_label_value1*> **[** *out_mpls_label_value2* **]]** applies DHCP over MPLS traffic.
- Service names must be unique across all contexts within a chassis.

DHCP Server Parameter Configuration

Use the following example to configure the DHCP server parameters to support DHCP-based address assignment:

```
configure
   context <dest_ctxt_name>
      dhcp-service <dhcp_svc_name>
```

```
dhcp server <ip_address> [ priority <priority> ]
dhcp server selection-algorithm {first-server | round-robin}
lease-duration min <minimum_dur> max <max_dur>
dhcp deadtime <max_time>
dhcp detect-dead-server consecutive-failures <max_number>
max-retransmissions <max_number>
retransmission-timeout <dur_sec>
end
```

Notes:

- Multiple DHCP can be configured by entering **dhcp server** command multiple times. A maximum of 20 DHCP servers can be configured.
- The **dhcp detect-dead-server** command and **max-retransmissions** command work in conjunction with each other.
- The retransmission-timeout command works in conjunction with **max-retransmissions** command.
- Service names must be unique across all contexts within a chassis.

DHCP Service Configuration Verification

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

show dhcp service all

This command produces an output similar to that displayed below where DHCP name is *dhcp1*:

```
Service name:                   dhcp1
Context:                                isp
Bind:                                   Done
Local IP Address:                       150.150.150.150
Next Hop Address:                       192.179.91.3
              MPLS-label:
                Input:                  5000
               Output:                  1566    1899
Service Status:                         Started
Retransmission Timeout:                 3000 (milli-secs)
Max Retransmissions:                    2
Lease Time:                             600 (secs)
Minimum Lease Duration:                 600 (secs)
Maximum Lease Duration:                 86400 (secs)
DHCP Dead Time:                         120 (secs)
DHCP Dead consecutive Failure:5
DHCP T1 Threshold Timer:                50
DHCP T2 Threshold Timer:                88
DHCP Client Identifier:                 Not Used
DHCP Algorithm:                         Round Robin
DHCP Servers configured:
 Address: 150.150.150.150               Priority: 1
DHCP server rapid-commit:               disabled
DHCP client rapid-commit:               disabled
DHCP chaddr validation:                 enabled
```

Step 2 Verify the DHCP service status by entering the following command in Exec Mode:

```
show dhcp service status
```

DHCPv6 Service Configuration

The system can be configured to use the Dynamic Host Control Protocol (DHCP) for IPv6 to enable the DHCP servers to pass the configuration parameters such as IPv6 network addresses to IPv6 nodes. DHCPv6 configuration is done within an APN.

These instructions assume that you have already configured the system level configuration as described in *System Administration Guide* and APN as described in *P-GW PDN Context Configuration* section of this chapter.

To configure the DHCPv6 service:

Step 1 Create the DHCPv6 service in system context and bind it by applying the example configuration in DHCPv6 Service Creation, on page 52.

Step 2 Configure the DHCPv6 server and other configurable values for Renew Time, Rebind Time, Preferred Lifetime, and Valid Lifetime by applying the example configuration in DHCPv6 Server Parameter Configuration, on page 53.

Step 3 Configure the DHCPv6 client and other configurable values for Maximum Retransmissions, Server Dead Tries, and Server Resurrect Time by applying the example configuration in DHCPv6 Client Parameter Configuration, on page 53.

Step 4 Configure the DHCPv6 profile by applying the example configuration in the *DHCPv6 Profile Configuration* section.

Step 5 Associate the DHCPv6 profile configuration with the APN by applying the example configuration in Associate DHCPv6 Configuration, on page 55.

Step 6 Verify your DHCPv6 Service configuration by following the steps in Associate DHCPv6 Configuration, on page 55.

Step 7 Save your configuration as described in the *Verifying and Saving Your Configuration* chapter.

DHCPv6 Service Creation

Use the following example to create the DHCPv6 service to support DHCP-based address assignment:

```
configure
   context <dest_ctxt_name>
      dhcpv6-service <dhcpv6_svc_name>
         bind address <ipv6_address> port <port>
         end
```

Notes:

- To ensure proper operation, DHCPv6 functionality should be configured within a destination context.
- The Port specifies the listen port and is used to start the DHCPv6 server bound to it. It is optional and if unspecified, the default port is 547.
- Service names must be unique across all contexts within a chassis.

DHCPv6 Server Parameter Configuration

Use the following example to configure the DHCPv6 server parameters to support DHCPv6-based address assignment:

```
configure
   context <dest_ctxt_name>
      dhcpv6-service <dhcpv6_svc_name>
         dhcpv6-server
         renew-time <renewal_time>
         rebind-time <rebind_time>
         preferred-lifetime <pref_lifetime>
         valid-lifetime <valid_lifetime>
         end
```

Notes:

- Multiple DHCP can be configured by entering **dhcp server** command multiple times. A maximum of 256 services (regardless of type) can be configured per system.
- **renew-time** configures the renewal time for prefixes assigned by dhcp-service. Default is 900 seconds.
- **rebind-time** configures the rebind time for prefixes assigned by dhcp-service. Default is 900 seconds.
- **preferred-lifetime** configures the preferred lifetime for prefixes assigned by dhcp-service. Default is 900 seconds.
- **valid-lifetime** configures the valid lifetime for prefixes assigned by dhcp-service. Default is 900 seconds.
- Service names must be unique across all contexts within a chassis.

DHCPv6 Client Parameter Configuration

Use the following example to configure the DHCPv6 client parameters to support DHCPv6-based address assignment:

```
configure
   context <dest_ctxt_name>
      dhcpv6-service <dhcpv6_svc_name>
         dhcpv6-client
            server-ipv6-address <ipv6_addr> port <port> priority <priority>
            max-retransmissions <max_number>
            server-dead-time <dead_time>
            server-resurrect-time <revive_time>
            end
```

Notes:

- DHCPv6 client configuration requires an IPv6 address, port, and priority. The port is used for communicating with the DHCPv6 server. If not specified, default port 547 is used. The Priority parameter defines the priority in which servers should be tried out.
- **max-retransmissions** configures the max retransmission that DHCPV6-CLIENT will make towards DHCPV6-SERVER. Default is 20.
- **server-dead-time**: PDN DHCPV6-SERVER is considered to be dead if it does not respond after given tries from client. Default is 5.

- **server-resurrect-time**: PDN DHCPV6-SERVER is considered alive after it has been dead for given seconds. Default is 20.
- Service names must be unique across all contexts within a chassis.

DHCPv6 Profile Configuration

Use the following example to configure the DHCPv6 profile:

```
configure
   context <dest_ctxt_name>
      dhcp-server-profile <server_profile>
         enable rapid-commit-dhcpv6
         process dhcp-option-from { AAA | LOCAL | PDN-DHCP } priority
<priority>
         dhcpv6-server-preference <pref_value>
         enable dhcpv6-server-unicast
         enable dhcpv6-server-reconf
         exit
      dhcp-client-profile <client_profile>
         dhcpv6-client-unicast
         client-identifier { IMSI | MSISDN }
         enable rapid-commit-dhcpv6
         enable dhcp-message-spray
         request dhcp-option dns-address
         request dhcp-option netbios-server-address
         request dhcp-option sip-server-address
         end
```

Notes:

- **dhcp-server-profile** command creates a server profile and then enters the DHCP Server Profile configuration mode.
- **enable rapid-commit-dhcpv6** command enables rapid commit on the DHCPv6 server. By default it is disabled. This is done to ensure that if there are multiple DHCPv6 servers in a network, with rapid-commit-option, they would all end up reserving resources for the UE.
- **process dhcp-option-from** command configures in what order the configuration options should be processed for a given client request. For a given client configuration, values can be obtained from either AAA, PDN-DHCP-SERVER, or LOCAL. By default, AAA is preferred over PDN-DHCP, which is preferred over LOCAL configuration.
- **dhcpv6-server-preference**: According to RFC-3315, DHCPv6-CLIENT should wait for a specified amount of time before considering responses to its queries from DHCPv6-SERVERS. If a server responds with a preference value of 255, DHCPv6-CLIENT need not wait any longer. Default value is 0 and it may have any configured integer between 1 and 255.
- **enable dhcpv6-server-unicast** command enables server-unicast option for DHCPv6. By default, it is disabled.
- **enable dhcpv6-server-reconf** command configures support for reconfiguration messages from the server. By default, it is disabled.
- **dhcpv6-client-unicast** command Enables client to send messages on unicast address towards the server.

- **dhcp-client-profile** command creates a client profile and then enters the DHCP Client Profile configuration mode.
- **client identifier** command configures the client-identifier, which is sent to the external DHCP server. By default, IMSI is sent. Another available option is MSISDN.
- **enable rapid-commit-dhcpv6** command configures the rapid commit for the client. By default, rapid-commit option is enabled for both DHCPv4 & DHCPv6.
- **enable dhcp-message-spray** command enables dhcp-client to spray a DHCP message to all configured DHCP servers in the PDN. By default this is disabled. With Rapid-Commit, there can only be one server to which this can be sent.
- **request dhcp-option** command configures DHCP options which can be requested by the dhcp-client. It supports the following options:
  - dns-address
  - netbios-server-address
  - sip-server-address
  -
- Service names must be unique across all contexts within a chassis.

Associate DHCPv6 Configuration

Use the following example to associate the DHCPv6 profile with an APN:

```
configure
   context <dest_ctxt_name>
      apn <apn_name>
         dhcpv6 service-name <dhcpv6_svc_name> server-profile <server_profile>
client-profile <client_profile>
         dhcpv6 ip-address-pool-name <dhcpv6_ip_pool>
         dhcpv6 context-name <dest_ctxt>
         end
```

Notes:

- Service names must be unique across all contexts within a chassis.

DHCPv6 Service Configuration Verification

Step 1 Verify that your DHCPv6 servers configured properly by entering the following command in Exec Mode:

show dhcpv6-service all

This command produces an output similar to that displayed below where DHCPv6 service name is *dhcp6-service*:

```
Service name:               dhcpv6-service
Context:                            A
Bind Address:                       2092::192:90:92:40
Bind :                              Done
Service Status:                     Started
Server Dead Time:                   120 (secs)
Server Dead consecutive Failure:5
```

```
Server Select Algorithm:            First Server
Server Renew Time:                  400 (secs)
Server Rebind Time:                 500 (secs)
Server Preferred Life Time:         600 (secs)
Server Valid Life Time:             700 (secs)
Max Retransmissions:                3 (secs)
Server Dead Tries:                  4 (secs)
Server Resurrect Time:              10 (secs)
ipv6_nd_flag:                       O_FLAG
DHCPv6 Servers configured:
          Address:                  2092::192:90:92:40 Priority: 1  enabled
```

Step 2 Verify the DHCPv6 service status by entering the following command in Exec Mode:

show dhcpv6 status service *dhcpv6_service_name*

Configuring Optional Features on the P-GW

The configuration examples in this section are optional and provided to cover the most common uses of the P-GW in a live network. The intent of these examples is to provide a base configuration for testing.

The following optional configurations are provided in this section:

Configuring ACL-based Node-to-Node IP Security on the S5 Interface

The configuration example in this section creates an IKEv2/IPSec ACL-based node-to-node tunnel endpoint on the S5 interface.

Important Use of the IP Security feature requires that a valid license key be installed. Contact your local Sales or Support representative for information on how to obtain a license.

The following configuration examples are included in this section:

• Creating and Configuring an IKEv2 Transform Set, on page 57
• Creating and Configuring a Crypto Map, on page 58

Creating and Configuring a Crypto Access Control List

The following example configures a crypto ACL (Access Control List), which defines the matching criteria used for routing subscriber data packets over an IPSec tunnel:

```
configure
  context <saegw_context_name> -noconfirm
    ip access-list <acl_name>
      permit tcp host <source_host_address> host <dest_host_address>
      end
```

Notes:

- The **permit** command in this example routes IPv4 traffic from the server with the specified source host IPv4 address to the server with the specified destination host IPv4 address.
- Service names must be unique across all contexts within a chassis.

Creating and Configuring an IPSec Transform Set

The following example configures an IPSec transform set, which is used to define the security association that determines the protocols used to protect the data on the interface:

```
configure
  context <saegw_context_name> -noconfirm
    ipsec transform-set <ipsec_transform-set_name>
      encryption aes-cbc-128
      group none
      hmac sha1-96
      mode tunnel
      end
```

Notes:

- The encryption algorithm, **aes-cbc-128**, or Advanced Encryption Standard Cipher Block Chaining, is the default algorithm for IPSec transform sets configured on the system.
- The **group none** command specifies that no crypto strength is included and that Perfect Forward Secrecy is disabled. This is the default setting for IPSec transform sets configured on the system.
- The **hmac** command configures the Encapsulating Security Payload (ESP) integrity algorithm. The **sha1-96** keyword uses a 160-bit secret key to produce a 160-bit authenticator value. This is the default setting for IPSec transform sets configured on the system.
- The **mode tunnel** command specifies that the entire packet is to be encapsulated by the IPSec header including the IP header. This is the default setting for IPSec transform sets configured on the system.
- Service names must be unique across all contexts within a chassis.

Creating and Configuring an IKEv2 Transform Set

The following example configures an IKEv2 transform set:

```
configure
  context <saegw_context_name> -noconfirm
    ikev2-ikesa transform-set <ikev2_transform-set_name>
      encryption aes-cbc-128
      group 2
      hmac sha1-96
      lifetime <sec>
      prf sha1
      end
```

Notes:

- The encryption algorithm, **aes-cbc-128**, or Advanced Encryption Standard Cipher Block Chaining, is the default algorithm for IKEv2 transform sets configured on the system.
- The **group 2** command specifies the Diffie-Hellman algorithm as Group 2, indicating medium security. The Diffie-Hellman algorithm controls the strength of the crypto exponentials. This is the default setting for IKEv2 transform sets configured on the system.
- The **hmac** command configures the Encapsulating Security Payload (ESP) integrity algorithm. The **sha1-96** keyword uses a 160-bit secret key to produce a 160-bit authenticator value. This is the default setting for IKEv2 transform sets configured on the system.
- The **lifetime** command configures the time the security key is allowed to exist, in seconds.
- The **prf** command configures the IKE Pseudo-random Function which produces a string of bits that cannot be distinguished from a random bit string without knowledge of the secret key. The **sha1** keyword uses a 160-bit secret key to produce a 160-bit authenticator value. This is the default setting for IKEv2 transform sets configured on the system.
- Service names must be unique across all contexts within a chassis.

Creating and Configuring a Crypto Map

The following example configures an IKEv2 crypto map:

```
configure
  context <saegw_context_name>
    crypto map <crypto_map_name> ikev2-ipv4
      match address <acl_name>
      peer <ipv4_address>
      authentication local pre-shared-key key <text>
      authentication remote pre-shared-key key <text>
      ikev2-ikesa transform-set list  <name1> . . . name6>
      payload <name> match ipv4
        lifetime <seconds>
        ipsec transform-set list <name1> . . . <name4>
        exit
      exit
    interface <s5_intf_name>
      ip address <ipv4_address>
      crypto-map <crypto_map_name>
      exit
    exit
  port ethernet <slot_number/port_number>
```

```
no shutdown
bind interface <s5_intf_name> <saegw_context_name>
end
```

Notes:

- The type of crypto map used in this example is IKEv2/IPv4 for IPv4 addressing. An IKEv2/IPv6 crypto map can also be used for IPv6 addressing.
- The **ipsec transform-set list** command specifies up to four IPSec transform sets.
- Service names must be unique across all contexts within a chassis.

Configuring APN as Emergency

The configuration example in this section configures an emergency APN for VoLTE based E911 support.

In APN Configuration Mode, specify the name of the emergency APN and set the emergency inactivity timeout as follows. You may also configure the P-CSCF FQDN server name for the APN.

```
configure
  context <saegw_context_name> -noconfirm
    apn <name>
      emergency-apn
      timeout emergency-inactivity <seconds>
      p-cscf fqdn <fqdn>
      end
```

Notes:

- By default, an APN is assumed to be non-emergency.
- The **timeout emergency-inactivity** command specifies the timeout duration, in seconds, to check inactivity on the emergency session. <*seconds*> must be an integer value from 1 through 3600.
- By default, emergency inactivity timeout is disabled (0).
- The **p-cscf fqdn** command configures the P-CSCF FQDN server name for the APN. <*fqdn*> must be a string from 1 to 256 characters in length.
- P-CSCF FQDN has more significance than CLI-configured P-CSCF IPv4 and IPv6 addresses.
- Service names must be unique across all contexts within a chassis.

Configuring Dynamic Node-to-Node IP Security on the S5 Interface

The configuration example in this section creates an IPSec/IKEv2 dynamic node-to-node tunnel endpoint on the S5 interface.

Important Use of the IP Security feature requires that a valid license key be installed. Contact your local Sales or Support representative for information on how to obtain a license.

The following configuration examples are included in this section:

Creating and Configuring an IPSec Transform Set

The following example configures an IPSec transform set, which is used to define the security association that determines the protocols used to protect the data on the interface:

```
configure
   context <saegw_context_name> -noconfirm
       ipsec transform-set <ipsec_transform-set_name>
         encryption aes-cbc-128
         group none
         hmac sha1-96
         mode tunnel
         end
```

Notes:

- The encryption algorithm, **aes-cbc-128**, or Advanced Encryption Standard Cipher Block Chaining, is the default algorithm for IPSec transform sets configured on the system.
- The **group none** command specifies that no crypto strength is included and that Perfect Forward Secrecy is disabled. This is the default setting for IPSec transform sets configured on the system.
- The **hmac** command configures the Encapsulating Security Payload (ESP) integrity algorithm. The **sha1-96** keyword uses a 160-bit secret key to produce a 160-bit authenticator value. This is the default setting for IPSec transform sets configured on the system.
- The **mode tunnel** command specifies that the entire packet is to be encapsulated by the IPSec header, including the IP header. This is the default setting for IPSec transform sets configured on the system.
- Service names must be unique across all contexts within a chassis.

Creating and Configuring an IKEv2 Transform Set

The following example configures an IKEv2 transform set:

```
configure
   context <saegw_context_name> -noconfirm
      ikev2-ikesa transform-set <ikev2_transform-set_name>
         encryption aes-cbc-128
         group 2
         hmac sha1-96
         lifetime <sec>
         prf sha1
         end
```

Notes:

- The encryption algorithm, **aes-cbc-128**, or Advanced Encryption Standard Cipher Block Chaining, is the default algorithm for IKEv2 transform sets configured on the system.

- The **group 2** command specifies the Diffie-Hellman algorithm as Group 2, indicating medium security. The Diffie-Hellman algorithm controls the strength of the crypto exponentials. This is the default setting for IKEv2 transform sets configured on the system.
- The **hmac** command configures the Encapsulating Security Payload (ESP) integrity algorithm. The **sha1-96** keyword uses a 160-bit secret key to produce a 160-bit authenticator value. This is the default setting for IKEv2 transform sets configured on the system.
- The **lifetime** command configures the time the security key is allowed to exist, in seconds.
- The **prf** command configures the IKE Pseudo-random Function, which produces a string of bits that cannot be distinguished from a random bit string without knowledge of the secret key. The **sha1** keyword uses a 160-bit secret key to produce a 160-bit authenticator value. This is the default setting for IKEv2 transform sets configured on the system.
- Service names must be unique across all contexts within a chassis.

Creating and Configuring a Crypto Template

The following example configures an IKEv2 crypto template:

```
configure
  context <saegw_context_name> -noconfirm
    crypto template <crypto_template_name> ikev2-dynamic
      ikev2-ikesa transform-set list <name1> . . . <name6>
        ikev2-ikesa rekey
        payload <name> match childsa match ipv4
          ipsec transform-set list <name1> . . . <name4>
          rekey
          end
```

Notes:

- The **ikev2-ikesa transform-set list** command specifies up to six IKEv2 transform sets.
- The **ipsec transform-set list** command specifies up to four IPSec transform sets.
- Service names must be unique across all contexts within a chassis.

Binding the S5 IP Address to the Crypto Template

The following example configures the binding of the S5 interface to the crypto template.

Important If you modify the **interface-type** command, the parent service (service within which the eGTP/GTP-U service is configured) will automatically restart. Service restart results in dropping of active calls associated with the parent service.

```
configure
  context <saegw_context_name> -noconfirm
    gtpu-service <gtpu_ingress_service_name>
      bind ipv4-address <s5_interface_ip_address> crypto-template
<sgw_s5_crypto_template>
      exit
    egtp-service <egtp_ingress_service_name>
```

```
      interface-type interface-pgw-ingress
      associate gtpu-service <gtpu_ingress_service_name>
      gtpc bind ipv4-address <s5_interface_ip_address>
      exit
   pgw-service <pgw_service_name> -noconfirm
      plmn id mcc <id> mnc <id> primary
      associate egtp-service <egtp_ingress_service_name>
      end
```

Notes:

- The **bind** command in the GTP-U and eGTP service configuration can also be specified as an IPv6 address using the **ipv6-address** command.
- Service names must be unique across all contexts within a chassis.

Configuring the GTP Echo Timer

The GTP echo timer on the ASR5x00 P-GW can be configured to support two different types of path management: default and dynamic. This timer can be configured on the GTP-C and/or the GTP-U channels.

Default GTP Echo Timer Configuration

The following examples describe the configuration of the default eGTP-C and GTP-U interface echo timers:

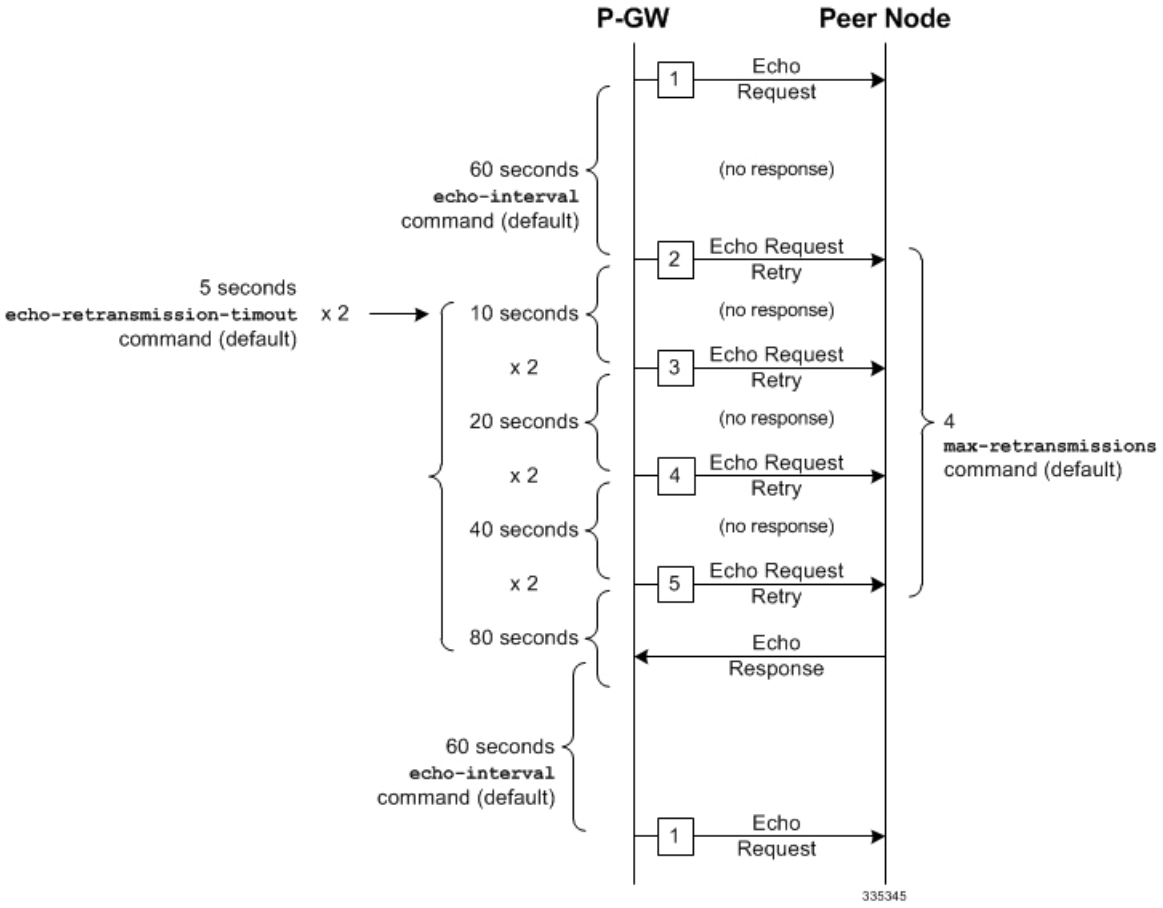
eGTP-C

```
configure
   context <context_name>
      egtp-service <egtp_service_name>
         gtpc echo-interval <seconds>
         gtpc echo-retransmission-timeout <seconds>
         gtpc max-retransmissions <num>
         end
```

Notes:

- The following diagram describes a failure and recovery scenario using default settings of the three **gtpc** commands in the example above:

Figure 9: Failure and Recovery Scenario - Example 1



- The multiplier (x2) is system-coded and cannot be configured.
- Service names must be unique across all contexts within a chassis.

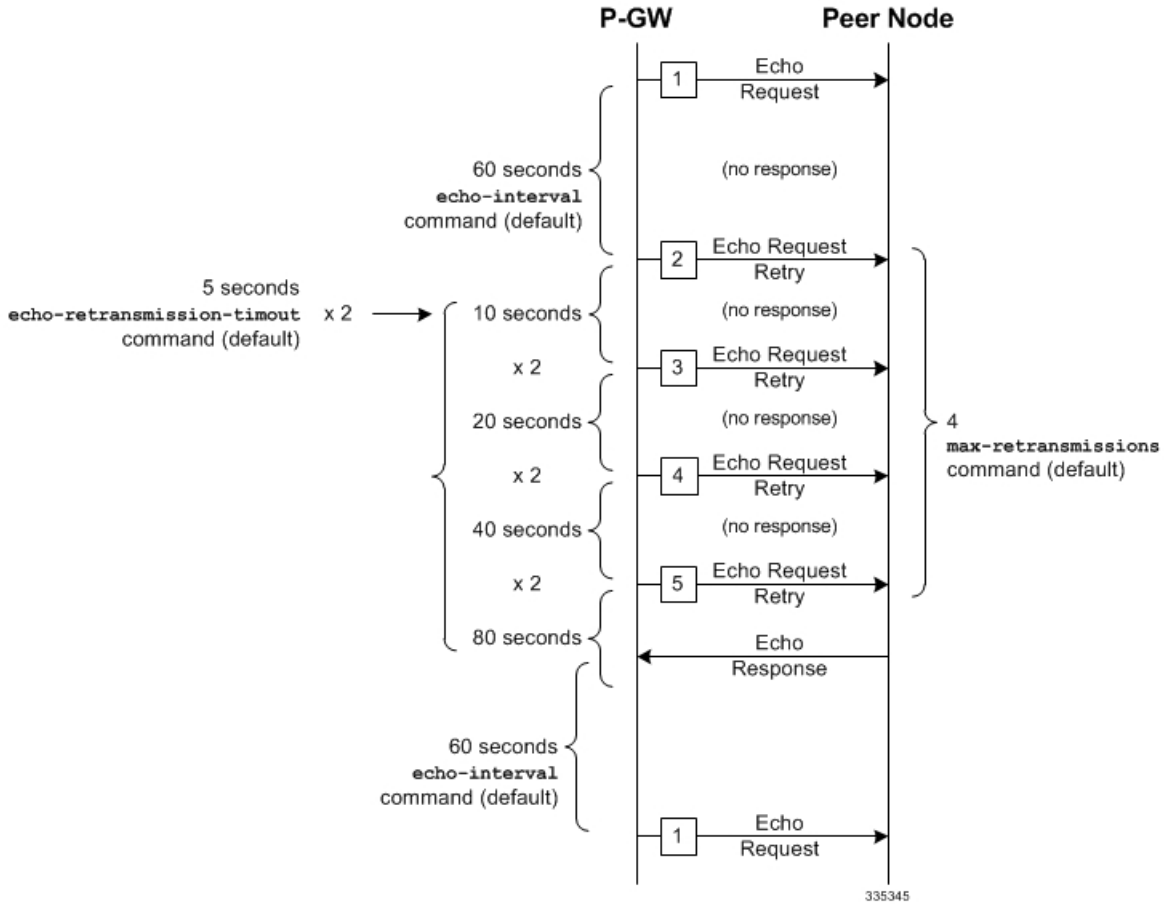
GTP-U

```
configure
   context <context_name>
      gtpu-service <gtpu_service_name>
         echo-interval <seconds>
         echo-retransmission-timeout <seconds>
         max-retransmissions <num>
         end
```

Notes:

- The following diagram describes a failure and recovery scenario using default settings of the three GTP-U commands in the example above:

Figure 10: Failure and Recovery Scenario - Example 2



- The multiplier (x2) is system-coded and cannot be configured.
- Service names must be unique across all contexts within a chassis.

Dynamic GTP Echo Timer Configuration

The following examples describe the configuration of the dynamic eGTP-C and GTP-U interface echo timers:

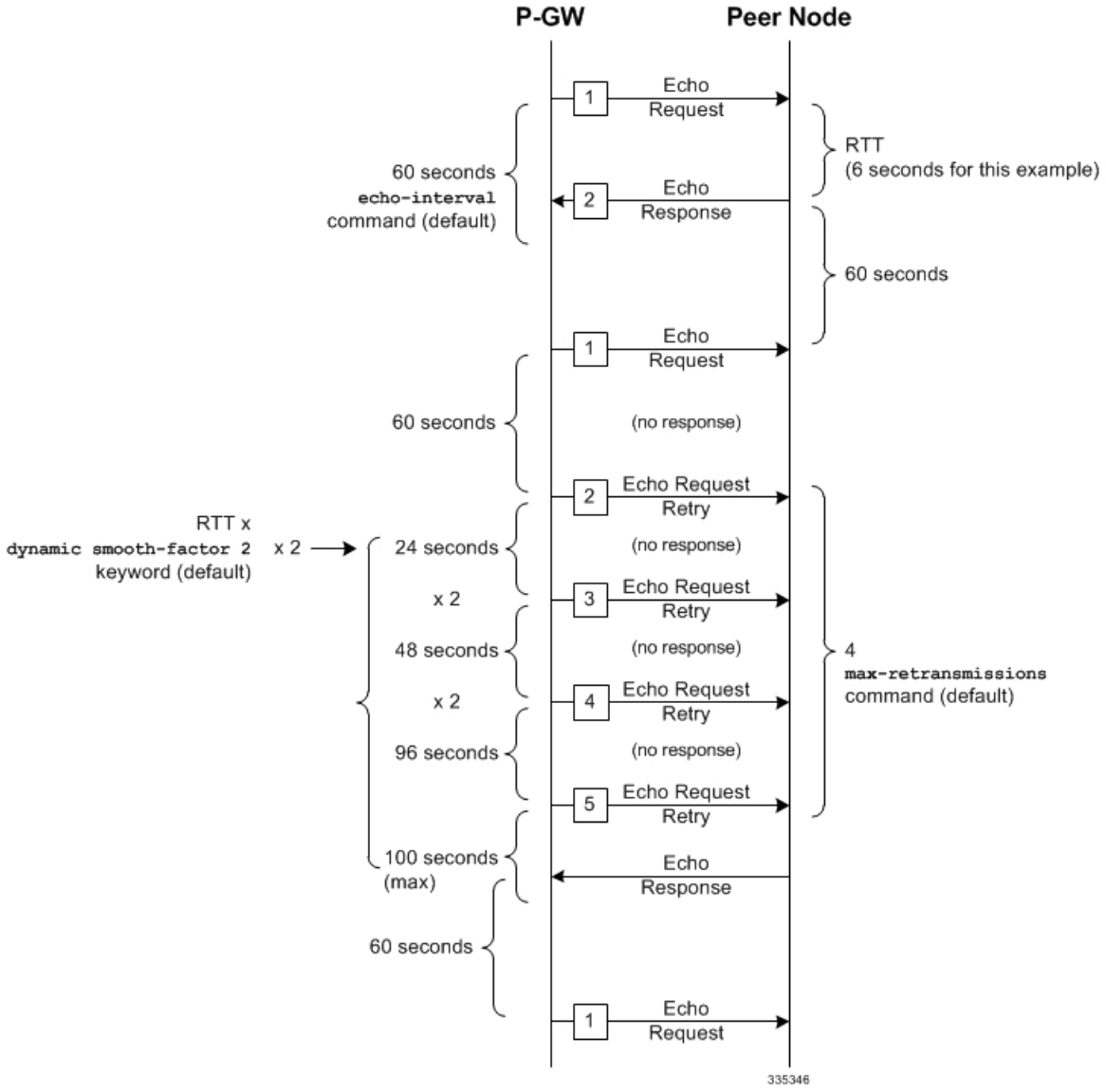
eGTP-C

```
configure
  context <context_name>
    egtp-service <egtp_service_name>
      gtpc echo-interval <seconds> dynamic smooth-factor <multiplier>
      gtpc echo-retransmission-timeout <seconds>
      gtpc max-retransmissions <num>
      end
```

Notes:

- The following diagram describes a failure and recovery scenario using default settings of the three **gtpc** commands in the example above and an example round trip timer (RTT) of six seconds:

Figure 11: Failure and Recovery Scenario - Example 3



- The multiplier (x2) and the 100 second maximum are system-coded and cannot be configured.
- Service names must be unique across all contexts within a chassis.

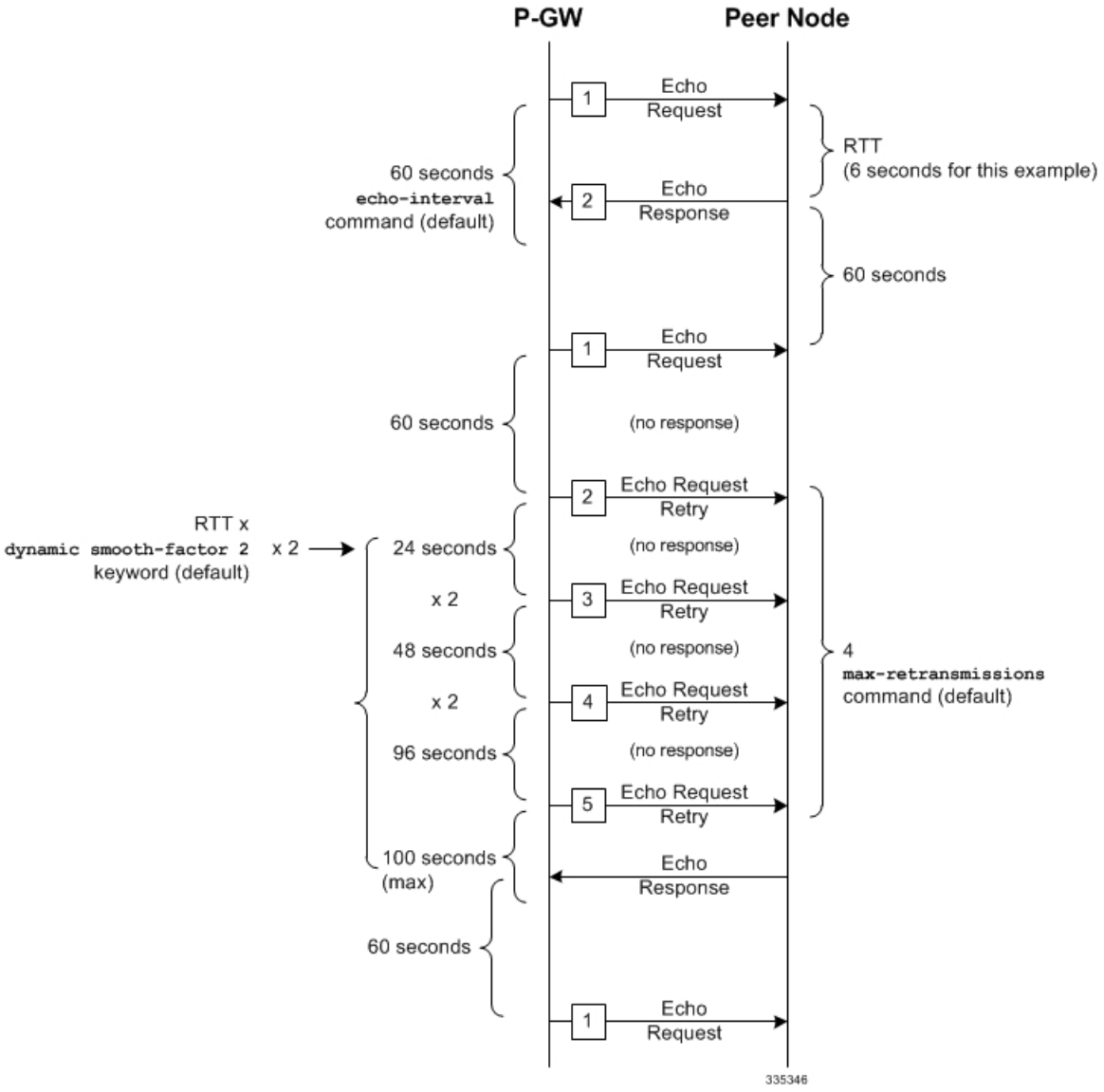
GTP-U

```
configure
  context <context_name>
    gtpu-service <gtpu_service_name>
    echo-interval <seconds> dynamic smooth-factor <multiplier>
    echo-retransmission-timeout <seconds>
    max-retransmissions <num>
    end
```

Notes:

- The following diagram describes a failure and recovery scenario using default settings of the three **gtpc** commands in the example above and an example round trip timer (RTT) of six seconds:

Figure 12: Failure and Recovery Scenario - Example 4



- The multiplier (x2) and the 100 second maximum are system-coded and cannot be configured.
- Service names must be unique across all contexts within a chassis.

Configuring GTPP Offline Accounting on the P-GW

By default the P-GW service supports GTPP accounting. To provide GTPP offline charging, configure the P-GW with the example parameters below:

```
configure
  gtpp single-source
```

```
context <saegw_context_name>
  subscriber default
    accounting mode gtpp
    exit
  gtpp group default
    gtpp charging-agent address <gz_ipv4_address>
    gtpp echo-interval <seconds>
    gtpp attribute diagnostics
    gtpp attribute local-record-sequence-number
    gtpp attribute node-id-suffix <string>
    gtpp dictionary <name>
    gtpp server <ipv4_address> priority <num>
    gtpp server <ipv4_address> priority <num> node-alive enable
    exit
  policy accounting <gz_policy_name>
    accounting-level {type}
    operator-string <string>
    cc profile <index> buckets <num>
    cc profile <index> interval <seconds>
    cc profile <index> volume total <octets>
    exit
  exit
context <saegw_context_name>
  apn apn_name
    associate accounting-policy <gz_policy_name>
    exit
  interface <gz_interface_name>
    ip address <address>
    exit
  exit
port ethernet <slot_number/port_number>
  no shutdown
  bind interface <gz_interface_name> <saegw_context_name>
  end
```

Notes:

- **gtpp single-source** is enabled to allow the system to generate requests to the accounting server using a single UDP port (by way of a AAA proxy function) rather than each AAA manager generating requests on unique UDP ports.
- **gtpp** is the default option for the **accounting mode** command.
- An accounting mode configured for the call-control profile will override this setting.
- **accounting-level** types are: flow, PDN, PDN-QCI, QCI, and subscriber. Refer to the *Accounting Profile Configuration Mode Commands* chapter in the *Command Line Interface Reference* for more information on this command.
- Service names must be unique across all contexts within a chassis.

Configuring Local QoS Policy

The configuration examples in this section creates a local QoS policy. A local QoS policy service can be used to control different aspects of a session, such as QoS, data usage, subscription profiles, or server usage, by means of locally defined policies.

Important Local QoS Policy is a licensed-controlled feature. Contact your Cisco account or support representative for detailed licensing information.

The following configuration examples are included in this section:

- Creating and Configuring a Local QoS Policy, on page 68
- Binding a Local QoS Policy, on page 69
- Verifying Local QoS Policy, on page 70

Creating and Configuring a Local QoS Policy

The following configuration example enables a local QoS policy on the P-GW:

```
configure
   local-policy-service <name> -noconfirm
      ruledef <ruledef_name> -noconfirm
         condition priority <priority> <variable> match <string_value>
         condition priority <priority> <variable> match <int_value>
         condition priority <priority> <variable> nomatch <regex>
         exit
      actiondef <actiondef_name> -noconfirm
         action priority <priority> <action_name> <arguments>
         action priority <priority> <action_name> <arguments>
         exit
      actiondef <actiondef_name> -noconfirm
         action priority <priority> <action_name> <arguments>
         action priority <priority> <action_name> <arguments>
         exit
      eventbase <eventbase_name> -noconfirm
         rule priority <priority> event <list_of_events> ruledef <ruledef_name>
actiondef <actiondef_name>
         end
```

Notes:

- A maximum of 16 local QoS policy services are supported.
- A maximum 256 ruledefs are suggested in a local QoS policy service for performance reasons.
- The **condition** command can be entered multiple times to configure multiple conditions for a ruledef. The conditions are examined in priority order until a match is found and the corresponding condition is applied.
- A maximum of 256 actiondefs are suggested in a local QoS policy service for performance reasons.
- The **action** command can be entered multiple times to configure multiple actions for an actiondef. The actions are examined in priority order until a match is found and the corresponding action is applied.

- Currently, only one eventbase is supported and must be named "default".
- The **rule** command can be entered multiple times to configure multiple rules for an eventbase.
- A maximum of 256 rules are suggested in an eventbase for performance reasons.
- Rules are executed in priority order, and if the rule is matched the action specified in the actiondef is executed. If an event qualifier is associated with a rule, the rule is matched only for that specific event. If a qualifier of **continue** is present at the end of the rule, the subsequent rules are also matched; otherwise, rule evaluation is terminated on first match.
- Service names must be unique across all contexts within a chassis.

Binding a Local QoS Policy

Option 1: The following configuration example binds the previously configured local QoS policy:

```
configure
  context <saegw_context_name> -noconfirm
    apn <name>
      ims-auth-service <local-policy-service name>
      end
```

Notes:

- A maximum of 30 authorization services can be configured globally in the system. There is also a system limit for the maximum number of total configured services.
- Useful in case of emergency calls; PCRF is not involved.
- Service names must be unique across all contexts within a chassis.

Option 2: The following configuration example may also be used to bind the previously configured local QoS policy or a failure handling template:

```
configure
  context <saegw_context_name> -noconfirm
    ims-auth-service <auth_svc_name>
      policy-control
        associate failure-handling-template <template_name>
        associate local-policy-service <service_name>
        end
```

Notes:

- Only one failure handling template can be associated with the IMS authorization service. The failure handling template should be configured prior to issuing this command.
- The failure handling template defines the action to be taken when the Diameter application encounters a failure supposing a result-code failure, tx-expiry or response-timeout. The application will take the action given by the template. For more information on failure handling template, refer to the *Diameter Failure Handling Template Configuration Mode Commands* chapter in the *Command Line Interface Reference*.
- You must select "local-fallback" in the failure handling template to support fallback to local policy.

- To support fallback to local policy in case of failure at PCRF, the local policy service should be associated with an IMS authorization service. In case of any failures, the local policy template associated with the ims-auth service will be chosen for fallback.
- Service names must be unique across all contexts within a chassis.

Verifying Local QoS Policy

The following configuration example verifies if local QoS service is enforced:

```
logging filter active facility local-policy level debug
logging active
show local-policy statistics all
```

Notes:

- Please take extreme caution not to use logging feature in console port and in production nodes.

Configuring X.509 Certificate-based Peer Authentication

The configuration example in this section enables X.509 certificate-based peer authentication, which can be used as the authentication method for IP Security on the P-GW.

Important Use of the IP Security feature requires that a valid license key be installed. Contact your local Sales or Support representative for information on how to obtain a license.

The following configuration example enables X.509 certificate-based peer authentication on the P-GW.

In Global Configuration Mode, specify the name of the X.509 certificate and CA certificate, as follows:

```
configure
   certificate name <cert_name> pem url <cert_pem_url> private-key pem url
<private_key_url>
   ca-certificate name <ca_cert_name> pem url <ca_cert_url>
   end
```

Notes:

- The **certificate name** and **ca-certificate list ca-cert-name** commands specify the X.509 certificate and CA certificate to be used.
- The PEM-formatted data for the certificate and CA certificate can be specified, or the information can be read from a file via a specified URL as shown in this example.
- Service names must be unique across all contexts within a chassis.

When creating the crypto template for IPSec in Context Configuration Mode, bind the X.509 certificate and CA certificate to the crypto template and enable X.509 certificate-based peer authentication for the local and remote nodes, as follows:

```
configure
   context <saegw_context_name> -noconfirm
      crypto template <crypto_template_name> ikev2-dynamic
         certificate name <cert_name>
```

```
ca-certificate list ca-cert-name <ca_cert_name>
authentication local certificate
authentication remote certificate
end
```

Notes:

- A maximum of 16 certificates and 16 CA certificates are supported per system. One certificate is supported per service, and a maximum of four CA certificates can be bound to one crypto template.
- The **certificate name** and **ca-certificate list ca-cert-name** commands bind the certificate and CA certificate to the crypto template.
- The **authentication local certificate** and **authentication remote certificate** commands enable X.509 certificate-based peer authentication for the local and remote nodes.
- Service names must be unique across all contexts within a chassis.

Configuring R12 Load Control Support

Load control enables a GTP-C entity (for example, an S-GW/P-GW) to send its load information to a GTP-C peer (e.g. an MME/SGSN, ePDG, TWAN) to adaptively balance the session load across entities supporting the same function (for example, an S-GW cluster) according to their effective load. The load information reflects the operating status of the resources of the GTP-C entity.

Use the following example to configure this feature:

```
configure
  gtpc-load-control-profile profile_name
    inclusion-frequency advertisement-interval interval_in_seconds
    weightage system-cpu-utilization percentage system-memory-utilization
 percentage license-session-utilization percentage
    end
configure
  context  context_name
    pgw-service pgw_service_name
      associate gtpc-load-control-profile profile_name
      exit
    saegw-service saegw_service_name
      associate pgw-service pgw_service_name
      end
```

Notes:

- The **inclusion-frequency** parameter determines how often the Load control information element is sent to the peer(s).
- The total of the three **weightage** parameters should not exceed 100.
- The **associate** command is used to associate the Load Control Profile with an existing S-GW service and to associate the P-GW service with the SAEGW service.
- On the SAEGW, both the P-GW and S-GW should use the same Load Control profile.

Configuring R12 Overload Control Support

Overload control enables a GTP-C entity becoming or being overloaded to gracefully reduce its incoming signaling load by instructing its GTP-C peers to reduce sending traffic according to its available signaling capacity to successfully process the traffic. A GTP-C entity is in overload when it operates over its signaling capacity, which results in diminished performance (including impacts to handling of incoming and outgoing traffic).

Use the following example to configure this feature.

```
configure
   gtpc-overload-control-profile profile_name
      inclusion-frequency advertisement-interval interval_in_seconds
      weightage system-cpu-utilization percentage system-memory-utilization
percentage license-session-utilization percentage
      throttling-behavior emergency-events exclude
      tolerance initial-reduction-metric percentage
      tolerance threshold report-reduction-metric percentage
self-protection-limit percentage
      validity-period seconds
      end
configure
   context context_name
      pgw-service pgw_service_name
         associate gtpc-overload-control-profile profile_name
         exit
      saegw-service saegw_service_name
         associate pgw-service pgw_service_name
         end
```

Notes:

- The **inclusion-frequency** parameter determines how often the Overload control information element is sent to the peer(s).
- The total of the three **weightage** parameters should not exceed 100.
- **validity-period** configures how long the overload control information is valid. Valid entries are from 1 to 3600 seconds. The default is 600 seconds.
- The **associate** command is used to associate the Overload Control Profile with an existing S-GW and P-GW service.
- On the SAEGW, both the P-GW and S-GW should use the same Overload Control profile.

Configuring Guard Timer on Create Session Request Processing

The P-GW has an existing timer "session setup-timeout" which is hard coded to 60 seconds, which is used as a guard timer for session creation. This timer is used for all APNs and is started when a Create Session Request is received for any session creation.

Internal or external processing issues or delay at external interfaces, for example, Gx/Gy, can cause Create Session Request processing to run longer than time expected in end to end call setup. If the session processing is not complete when the timer expires, the Create Session Request processing is stopped and the P-GW performs an internal cleanup by stopping all other corresponding sessions, for example Gx/Gy. The P-GW

responds with a Create Session Failure response stating that no resources are available to S-GW. In successful cases when there's no delay, the timer is stopped while sending out the Create Session Response.

A new CLI command has been introduced to allow a configurable value to override the previously hardcoded default session setup timeout value of 60 seconds. This will help to fine tune the call setup time at P-GW with respect to end to end call setup time.

Configuring Session Timeout

The following configuration example makes a P-GW session setup timeout configurable.

```
configure
   context context_name
      pgw-service  service_name
         setup-timeout  timer-value
         [  default | no ] setup-timeout
         end
```

Notes:

- **setup-timeout**: Specifies the session setup timeout period, in seconds. If P-GW is able to process the Create Session Request message before the timer expires, P-GW stops the timer and sends a successful Create Session Response.

  timer_value must be an integer from 1 to 120.

  Default: 60 seconds

- **default**: Default value is 60 seconds. If no value is set, the P-GW service sets the timer to the default value.
- **no**: Sets the timer to the default value of 60 seconds.

Configuring RLF Bypass

The Bypass Rate Limit Function is an enhancement to the existing GTP Throttling feature. The RLF feature allows the operator to control the bypassing of some messages being throttled.

A new command option **throttling-override-policy** has been added to the existing CLI command **gtpc overload-protection egress rlf-template rlf-temp** which allows you to selectively by-pass throttling for a configured message type or for all messages in emergency call or priority call or call for the configured APN. A new CLI command mode **throttling-override-policy** has been also been introduced for Generic syntax for throttling override policy.

Configuring the Throttling Override Policy Mode

The following configuration helps to create a GTP-C Throttling Override Policy and to enter GTP-C Throttling Override Policy mode.

```
configure
   throttling-override-policy throttling-override-policy_name
```

Notes:

Entering the above command sequence results in the following prompt:

```
[local]  host_name  (config-throttling-override-policy)
```

Configuring RLF Bypass Feature

The following configuration configures message types which can bypass the rate limiting function.

```
configure
   throttling-override-policy  throttling-override-policy_name
      [ default | no ] egress bypass-rlf pgw { msg-type { cbr | dbr | ubr
 | emergency-call | earp-pl-list {1 | 2 | 3 | 4 | 5 ...  | 15 }+ |
apn-names  <apn-name1> <apn-name2> <apn-name3> }
      end
```

Notes:

- If an empty throttling-override-policy is created, then the default values for all the configurables are zeros/disabled.
- If no throttling-override-policy is associated, then **show service configuration** for P-GW will show it as "n/a".
- Maximum number of throttling-override-policy that can be added are 1024. This limit is the same as max RLF templates.

Example

The following command configures Create Bearer Request message type at the P-GW node to bypass throttling.

```
egress bypass-rlf pgw msg-type cbr
```