ePDG Administration Guide, StarOS Release 18
Last Updated: October 30, 2015

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
     800 553-NETS (6387)
Fax: 408 527-0883
THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB’s public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS” WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental. 

ePDG Administration Guide, StarOS Release 18

© 2015 Cisco Systems, Inc. All rights reserved.
# CONTENTS

**About this Guide** .................................................................................................................. vii
  Conventions Used .................................................................................................................. viii
  Supported Documents and Resources .............................................................................. ix
  Related Common Documentation .................................................................................... ix
  Related Product Documentation ...................................................................................... ix
  Obtaining Documentation ................................................................................................... ix
  Contacting Customer Support ............................................................................................ x

**Evolved Packet Data Gateway Overview** .......................................................................... 11
  Product Description ............................................................................................................ 12
  Platform Requirements ....................................................................................................... 12
  Licenses ............................................................................................................................... 12
  Network Deployment(s) and Interfaces .............................................................................. 13
    Network Elements ............................................................................................................ 13
      ePDG ............................................................................................................................... 13
      eNodeB ............................................................................................................................ 14
      MME ............................................................................................................................... 14
      S-GW ............................................................................................................................. 14
      P-GW ............................................................................................................................. 14
      3GPP AAA Server ......................................................................................................... 14
      HSS ............................................................................................................................... 14
      PCRF .............................................................................................................................. 14
    Logical Network Interfaces ............................................................................................ 15
    Transport Combinations ............................................................................................... 15
  Features and Functionality ............................................................................................... 17
    ePDG Service ................................................................................................................... 18
    IKEv2 and IPSec Encryption ........................................................................................... 18
      Supported Algorithms ................................................................................................. 19
      x.509 Digital Certificate Handling .............................................................................. 19
      Timers ............................................................................................................................ 19
    Dead Peer Detection ....................................................................................................... 20
    Child SA Rekeying .......................................................................................................... 20
    Support for MAC Address of WiFi Access Points ......................................................... 20
    AAA Server Groups ........................................................................................................ 20
    EAP Authentication ......................................................................................................... 21
    IPv6 Capabilities ............................................................................................................ 21
    General Call Flow .......................................................................................................... 21
    P-GW Selection ............................................................................................................. 24
      Static Selection ............................................................................................................. 24
      Dynamic Selection ....................................................................................................... 26
      P-GW Initiated Bearer Modification ............................................................................ 27
      Topology/Weight-based Selection .............................................................................. 28
    Dual Stack Support ........................................................................................................ 28
    Inter-access Handover Support .................................................................................... 29
    Mobile Access Gateway Function .................................................................................. 29
    IPv6 Router Advertisement Support ............................................................................. 29
Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating the ePDG Service</td>
<td>99</td>
</tr>
<tr>
<td>Egress Context and MAG Service Configuration</td>
<td>101</td>
</tr>
<tr>
<td>Configuring the Egress Context and MAG Service</td>
<td>101</td>
</tr>
<tr>
<td>Egress Context and EGTP Service Configuration</td>
<td>103</td>
</tr>
<tr>
<td>Configuring the Egress Context and EGTP Service</td>
<td>103</td>
</tr>
<tr>
<td>Bulk Statistics Configuration</td>
<td>105</td>
</tr>
<tr>
<td>Logging Configuration</td>
<td>106</td>
</tr>
<tr>
<td>Non UICC device support for certificate and multi authentication config</td>
<td>106</td>
</tr>
<tr>
<td>Saving the Configuration</td>
<td>107</td>
</tr>
<tr>
<td>Monitoring the Evolved Packet Data Gateway</td>
<td>109</td>
</tr>
<tr>
<td>Monitoring ePDG Status and Performance</td>
<td>110</td>
</tr>
<tr>
<td>Clearing Statistics and Counters</td>
<td>116</td>
</tr>
<tr>
<td>Evolved Packet Data Gateway Engineering Rules</td>
<td>117</td>
</tr>
<tr>
<td>IKEv2/IPSec Restrictions</td>
<td>118</td>
</tr>
<tr>
<td>X.509 Certificate (CERT) Restrictions</td>
<td>119</td>
</tr>
<tr>
<td>GTPv2 Restrictions</td>
<td>120</td>
</tr>
<tr>
<td>S2b Interface Rules</td>
<td>121</td>
</tr>
<tr>
<td>MAG-to-LMA Rules</td>
<td>121</td>
</tr>
<tr>
<td>EGTP Service Rules</td>
<td>121</td>
</tr>
<tr>
<td>ePDG Service Rules</td>
<td>121</td>
</tr>
<tr>
<td>ePDG Subscriber Rules</td>
<td>122</td>
</tr>
<tr>
<td>IKEv2 Error Codes and Notifications</td>
<td>125</td>
</tr>
</tbody>
</table>
About this Guide

This preface describes the ePDG Administration Guide, how it is organized, and its document conventions.

The guide describes the ePDG (Evolved Packet Data Gateway) and includes network deployments and interfaces, feature descriptions, session flows, configuration instructions, and CLI commands for monitoring and troubleshooting the system. It also contains a sample ePDG configuration file and ePDG engineering rules.
Conventions Used

The following tables describe the conventions used throughout this documentation.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Notice Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Information Note" /></td>
<td>Information Note</td>
<td>Provides information about important features or instructions.</td>
</tr>
<tr>
<td><img src="image" alt="Caution" /></td>
<td>Caution</td>
<td>Alerts you of potential damage to a program, device, or system.</td>
</tr>
<tr>
<td><img src="image" alt="Warning" /></td>
<td>Warning</td>
<td>Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Typeface Conventions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text represented as a screen display</td>
<td>This typeface represents displays that appear on your terminal screen, for example: Login:</td>
</tr>
<tr>
<td>Text represented as commands</td>
<td>This typeface represents commands that you enter, for example: show ip access-list</td>
</tr>
<tr>
<td></td>
<td>This document always gives the full form of a command in lowercase letters. Commands are not case sensitive.</td>
</tr>
<tr>
<td>Text represented as a command variable</td>
<td>This typeface represents a variable that is part of a command, for example: show card slot_number</td>
</tr>
<tr>
<td></td>
<td>slot_number is a variable representing the desired chassis slot number.</td>
</tr>
<tr>
<td>Text represented as menu or sub-menu names</td>
<td>This typeface represents menus and sub-menus that you access within a software application, for example: Click the File menu, then click New</td>
</tr>
</tbody>
</table>
Supported Documents and Resources

Related Common Documentation

The most up-to-date information for this product is available in the product release notes provided with each product release.

The following common documents are available:

- AAA Interface Administration and Reference
- Command Line Interface Reference
- Installation Guide (platform dependent)
- Release Change Reference
- SNMP MIB Reference,
- Statistics and Counters Reference
- System Administration Guide (platform dependent)
- Thresholding Configuration Guide

Related Product Documentation

The following product documents are also available and can be used in conjunction with the ePDG documentation:

- Packet Data Network Gateway Administration Guide
- Serving Gateway Administration Guide
- Mobility Management Entity Administration Guide

Obtaining Documentation

The most current Cisco documentation is available on the following website:

http://www.cisco.com/cisco/web/psa/default.html

Use the following path selections to access the ePDG documentation:

Contacting Customer Support

Use the information in this section to contact customer support.

Refer to the support area of http://www.cisco.com for up-to-date product documentation or to submit a service request. A valid username and password are required to access this site. Please contact your Cisco sales or service representative for additional information.
Chapter 1
Evolved Packet Data Gateway Overview

This chapter contains an overview of the ePDG (evolved Packet Data Gateway), including:

- Product Description
- Network Deployment(s) and Interfaces
- Features and Functionality
- How the ePDG Works
- Supported Standards
Product Description

The Cisco® ePDG (evolved Packet Data Gateway) enables mobile operators to provide secure access to the 3GPP E-UTRAN/EPC (Evolved UTRAN/Evolved Packet Core) network from untrusted non-3GPP IP access networks. The ePDG functions as a security gateway to provide network security and internet working control via IPSec tunnel establishment based on information obtained during 3GPP AAA (Authentication, Authorization, and Accounting). The ePDG enables mobile operators to extend wireless service coverage, reduce the load on the macro wireless network, and make use of existing backhaul infrastructure to reduce the cost of carrying wireless calls.

The ePDG has the following key features:

- Support for the IPSec/IKEv2-based SWu interface between the ePDG and the WLAN (Wireless LAN) UEs.
- Routing of packets between the WLAN UEs and the Cisco P-GW (Packet Data Network Gateway) over the S2b interface via GTPv2 or PMIPv6 (Proxy Mobile IP version 6) protocol.
- P-GW selection via DNS client functionality to provide PDN (Packet Data Network) connectivity to the WLAN UEs.
- Support for passing assigned IPv4/IPv6 address configurations from the P-GW to the WLAN UEs.
- Support for the Diameter-based SWm interface between the ePDG and the external 3GPP AAA server.
- Tunnel authentication and authorization for IPSec/PMIPv6/GTPv2 tunnels using the EAP-AKA (Extensible Authentication Protocol - Authentication and Key Agreement) authentication method between the 3GPP AAA server and the WLAN UEs.
- Encapsulation and decapsulation of packets sent over the IPSec/PMIPv6/GTPv2 tunnels.
- Hosts a MAG (Mobile Access Gateway) function, which acts as a proxy mobility agent in the E-UTRAN/EPC network and uses PMIPv6 signaling to provide network-based mobility management on behalf of the WLAN UEs attached to the network.

Platform Requirements

The ePDG service runs on a Cisco ASR 5000/ASR 5500 chassis running the StarOS operating system and Virtualized Packet Core (VPC) platforms. The chassis can be configured with a variety of components to meet specific network deployment requirements. For additional information, see the installation guide for the chassis and/or contact your Cisco account representative.

Licenses

The ePDG is a licensed Cisco product. Separate session and feature licenses may be required. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, see “Managing License Keys” in the System Administration Guide.
Network Deployment(s) and Interfaces

This section describes the ePDG as it provides secure access from the WLAN UEs to the Cisco P-GW and a connection to the PDN (Packet Data Network) in the E-UTRAN/EPC (Evolved UTRAN/Evolved Packet Core) network. The figure below shows the ePDG terminating the SWu interface from the untrusted non-3GPP IP access network and providing secure access to the Cisco P-GW and a connection to the PDN via the PMIPv6/GTPv2 S2b interface. It also shows the network interfaces used by the Cisco MME, S-GW, and P-GW in the E-UTRAN/EPC network.

Figure 1. The ePDG in the E-UTRAN/EPC Network

Network Elements

This section provides a description of the network elements that work with the ePDG in the E-UTRAN/EPC network. For untrusted non-3GPP IP access, note that the network architecture assumes the access network elements do not perform any function other than delivering packets.

ePDG

The ePDG is responsible for interworking between the EPC and untrusted non-3GPP networks that require secure access, such as a WiFi, LTE metro, and femtocell access networks.
eNodeB

The eNodeB (evolved Node B) is the termination point for all radio-related protocols. As a network, E-UTRAN is simply a mesh of eNodeBs connected to neighboring eNodeBs via the X2 interface.

MME

The Cisco MME (Mobility Management Entity) is the key control node for the LTE access network. It works in conjunction with the eNodeB and the Cisco S-GW to control bearer activation and deactivation. The MME is typically responsible for selecting the Cisco P-GW for the UEs to access the PDN, but for secure access from untrusted non-3GPP IP access networks, the ePDG is responsible for selecting the P-GW.

S-GW

The Cisco S-GW (Serving Gateway) routes and forwards data packets from the 3GPP UEs and acts as the mobility anchor during inter-eNodeB handovers. The S-GW receives signals from the MME that control the data traffic. Every 3GPP UE accessing the EPC is associated with a single S-GW.

P-GW

The Cisco P-GW (Packet Data Network Gateway) is the network node that terminates the SGi interface towards the PDN. The P-GW provides connectivity to external PDNs for the subscriber UEs by being the point of entry and exit for all subscriber UE traffic. A subscriber UE may have simultaneous connectivity with more than one P-GW for accessing multiple PDNs. The P-GW performs policy enforcement, packet filtering, charging support, lawful interception, and packet screening. The P-GW is the mobility anchor for both trusted and untrusted non-3GPP IP access networks. For PMIP-based S2a and S2b interfaces, the P-GW hosts the LMA (Local Mobility Anchor) function.

3GPP AAA Server

The 3GPP AAA (Authentication, Authorization, and Accounting) server provides UE authentication via the EAP-AKA (Extensible Authentication Protocol - Authentication and Key Agreement) authentication method.

HSS

The HSS (Home Subscriber Server), is the master user database that supports the IMS (IP Multimedia Subsystem) network entities. It contains subscriber profiles, performs subscriber authentication and authorization, and provides information about the subscriber's location and IP information.

PCRF

The PCRF (Policy and Charging Rules Function) determines policy rules in the IMS network. The PCRF operates in the network core, accesses subscriber databases and charging systems, and makes intelligent policy decisions for subscribers.
Logical Network Interfaces

The following table provides descriptions of the logical network interfaces supported by the ePDG in the E-UTRAN/EPC network.

Table 1. Logical Network Interfaces on the ePDG

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWu Interface</td>
<td>The secure interface to the WLAN UEs in the untrusted non-3GPP IP access network, the SWu interface carries IPSec tunnels. The ePDG uses IKEv2 signaling to establish IPSec tunnels between the UEs and the ePDG. It also supports the negotiation of configuration attributes such as IP address, DNS, and P-CSCF in the CP (Configuration Parameters) payload of IKE_AUTH Request and Response messages.</td>
</tr>
<tr>
<td>SWm Diameter Interface</td>
<td>The interface to the 3GPP Diameter AAA server, the SWm interface is used for WLAN UE authentication. It supports the transport of mobility parameters, tunnel authentication, and authorization data. The EAP-AKA (Extensible Authentication Protocol - Authentication and Key Agreement) method is used for authenticating the WLAN UEs over this interface.</td>
</tr>
<tr>
<td>S2b Interface</td>
<td>The interface to the P-GW, the S2b interface runs PMIPv6 (Proxy Mobile IP version 6)/GTPv2 protocol to establish WLAN UE sessions with the P-GW. It also supports the transport of P-CSCF attributes and DNS attributes in PBU (Proxy-MIP Binding Update)/Create Session Request and PBA (Proxy-MIP Binding Acknowledgement)/Create Session Response messages as part of the P-CSCF discovery performed by the WLAN UEs.</td>
</tr>
</tbody>
</table>

Transport Combinations

The table below lists the IPv4/IPv6 transport combinations for the ePDG and whether each combination is supported for deployment in this release.

Table 2. Transport Combinations for the ePDG

<table>
<thead>
<tr>
<th>IP Address Allocated by the P-GW for the WLAN UEs</th>
<th>IPSec Tunnels (between the WLAN UEs and the ePDG)</th>
<th>GTPv2</th>
<th>Combination Supported for Deployment?</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4</td>
<td>IPv4</td>
<td>IPv4</td>
<td>Yes</td>
</tr>
<tr>
<td>IPv4</td>
<td>IPv6</td>
<td>IPv6</td>
<td>Yes</td>
</tr>
<tr>
<td>IPv4</td>
<td>IPv6</td>
<td>IPv4</td>
<td>Yes</td>
</tr>
<tr>
<td>IPv4</td>
<td>IPv4</td>
<td>IPv6</td>
<td>Yes</td>
</tr>
<tr>
<td>IPv6</td>
<td>IPv4</td>
<td>IPv4</td>
<td>Yes</td>
</tr>
<tr>
<td>IPv6</td>
<td>IPv6</td>
<td>IPv6</td>
<td>Yes</td>
</tr>
<tr>
<td>IPv6</td>
<td>IPv6</td>
<td>IPv4</td>
<td>Yes</td>
</tr>
<tr>
<td>IPv6</td>
<td>IPv4</td>
<td>IPv6</td>
<td>Yes</td>
</tr>
<tr>
<td>IPvv6</td>
<td>IPv4</td>
<td>IPv4</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Network Deployment(s) and Interfaces

<table>
<thead>
<tr>
<th>IP Address Allocated by the P-GW for the WLAN UEs</th>
<th>IPSec Tunnels (between the WLAN UEs and the ePDG)</th>
<th>GTPv2</th>
<th>Combination Supported for Deployment?</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4v6</td>
<td>IPv6</td>
<td>IPv6</td>
<td>Yes</td>
</tr>
<tr>
<td>IPv4v6</td>
<td>IPv4</td>
<td>IPv6</td>
<td>Yes</td>
</tr>
<tr>
<td>IPv4v6</td>
<td>IPv6</td>
<td>IPv4</td>
<td>Yes</td>
</tr>
</tbody>
</table>

PMIPv6 S2b IPv6 transport is qualified.
Features and Functionality

This section describes the ePDG features and functions, as follows:

- ePDG Service
- IKEv2 and IPSec Encryption
- Dead Peer Detection
- Child SA Rekeying
- Support for MAC Address of WiFi Access Points
- AAA Server Groups
- EAP Authentication
- IPv6 Capabilities
- P-GW Selection
- Dual Stack Support
- Inter-access Handover Support
- Mobile Access Gateway Function
- IPv6 Router Advertisement Support
- DNS Request Support
- P-CSCF Request Support
- Multiple PDN Support
- Default APN Support
- Congestion Control
- Session Recovery Support
- DSCP and 802.1P Marking
- IPSec Cookie Threshold
- Threshold Crossing Alerts
- Bulk Statistics Support
- ePDG ICSR Support
- IKEv2 RFC 5996 Support
- IPv6 Support on IPSec SWU Interface
- Narrowing Traffic Selectors
- Static IP address allocation Support
- Ref - ePDG and PGW Support on the Same Chassis (with GTPv2)
- ICSR-VoLTE Support
- Local PGW Resolution Support
- Non UICC Device Support Using Certificate Based Authentication
- EAP-MSCHAPv2-EAP-TLSEAP-TTLS based support for NON UICC devices
- Emergency APN Support on ePDG
- Passing on UE tunnel Endpoint Address over SWm support
- Custom SWm to SWu error code mapping
- ePDGBearer Duration KPIs
- Data buffering support for DL packets before session establishment
- Downlink DSCP Marking(SWu)
- ePDG Fast Re-Auth Support
- ePDG Offline charging

**ePDG Service**

The ePDG service enables the WLAN UEs in the untrusted non-3GPP IP access network to connect to the E-UTRAN/EPC network via a secure IPSec interface.

During configuration, you create the ePDG service in an ePDG context, which is a routing domain in the system. Context and service configuration for the ePDG includes the following main steps:

- **Configure the IPv4/IPv6 address for the service:** This is the IP address of the ePDG to which the WLAN UEs attempt to connect, sending IKEv2 messages to this address to establish IPSec tunnels.

- **Configure the name of the crypto template for IKEv2/IPSec:** A crypto template is used to define an IKEv2/IPSec policy. It includes IKEv2 and IPSec parameters for keepalive, lifetime, NAT-T, and cryptographic and authentication algorithms. There must be one crypto template per ePDG service.

- **The name of the EAP profile:** The EAP profile defines the EAP authentication method and associated parameters.

- **IKEv2 and IPSec transform sets:** Transform sets define the negotiable algorithms for IKE SAs (Security Associations) and Child SAs to enable calls to connect to the ePDG.

- **The setup timeout value:** This parameter specifies the session setup timeout timer value. The ePDG terminates a UE connection attempt if the UE does not establish a successful connection within the specified timeout period. The default value is 60 seconds.

- **Max-sessions:** This parameter sets the maximum number of subscriber sessions allowed by the ePDG service. The default value is 1,000,000 and is subject to license limitations.

- **DNS client:** DNS client configuration is needed for P-GW selection.

**IKEv2 and IPSec Encryption**

The ePDG supports IKEv2 (Internet Key Exchange version 2) and IPSec (IP Security) ESP (Encapsulating Security Payload) encryption as per RFCs 4303 and 5996. IKEv2 and IPSec encryption enables network domain security for all IP packet-switched networks in order to provide confidentiality, integrity, authentication, and anti-replay protection. These capabilities are ensured through use of cryptographic techniques.
The data path from the ePDG supports mixed inner IPv4 and IPv6 addresses in the same Child SA for ESP (Encapsulating Security Payload) encapsulation and decapsulation when the Any option is configured in the payload, regardless of the IP version of the outer protocol.

**Supported Algorithms**

The ePDG supports the protocols in the table below, which are specified in RFC 5996.

**Table 3. Supported Algorithms**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Type</th>
<th>Supported Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet Key Exchange version 2</td>
<td>IKEv2 Encryption</td>
<td>DES-CBC, 3DES-CBC, AES-CBC-128, AES-CBC-256</td>
</tr>
<tr>
<td></td>
<td>IKEv2 Pseudo Random Function</td>
<td>PRF-HMAC-SHA1, PRF-HMAC-MD5, AES-XCBC-PRF-128</td>
</tr>
<tr>
<td></td>
<td>IKEv2 Integrity</td>
<td>HMAC-SHA1-96, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512, HMAC-MD5-96, AES-XCBC-96</td>
</tr>
<tr>
<td></td>
<td>IKEv2 Diffie-Hellman Group</td>
<td>Group 1 (768-bit), Group 2 (1024-bit), Group 5 (1536-bit), Group 14 (2048-bit)</td>
</tr>
<tr>
<td></td>
<td>Extended Sequence Number</td>
<td>Value of 0 or off is supported (ESN itself is not supported)</td>
</tr>
<tr>
<td></td>
<td>IPSec Integrity</td>
<td>NULL, HMAC-SHA1-96, HMAC-MD5-96, AES-XCBC-96</td>
</tr>
</tbody>
</table>

**x.509 Digital Certificate Handling**

A digital certificate is an electronic credit card that establishes a subscriber’s credentials when doing business or other transactions on the Internet. The digital certificates used by the ePDG conform to ITU-T standard X.509 for a PKI (Public Key Infrastructure) and PMI (Privilege Management Infrastructure). X.509 specifies standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm.

The ePDG is capable of authenticating itself to the UE using certificates and does so in the response to the first IKE_AUTH Request message from the UE.

ePDG also supports hash and URL based encoding of certificate payloads in IKE exchanges.

The ePDG generates an SNMP notification when the certificate is within 30 days of expiration and approximately once a day until a new certificate is provided. Operators need to generate a new certificate and then configure the new certificate using the system’s CLI. The certificate is then used for all new sessions.

**Timers**

The ePDG includes the following timers for IPSec tunnels:

- **IKE Session Setup Timer**: This timer ensures that an IKE session set up is completed within a configured period. The ePDG tears down the call if it is still in progress when the timer expires. The default value is 120 seconds, and the range is between 1 and 3600 seconds.
• **IKEv2 and IPSec SA Lifetime Timers:** The ePDG maintains separate SA lifetime timers for both IKEv2 SAs and IPSec SAs. All timers are started when an SA is successfully set up. If there is traffic through the SA, the ePDG may initiate rekeying. If there is no traffic and rekey keepalive is not required, the ePDG deletes the SA without rekeying. If there is no traffic and rekey keepalive is required, the ePDG attempts to rekey. The default value of the IKEv2 SA lifetime timer is 86400 seconds and the range is between 60 and 86400 seconds. The default value of the IPSec SA lifetime timer is 86400 seconds and the range is between 60 and 86400 seconds.

• **DPD Timers:** By default, DPD (Dead Peer Detection) is disabled. When enabled, the ePDG may initiate DPD via IKEv2 keepalive messages to check the liveliness of the WLAN UEs. When enabled, the ePDG always respond to DPD checks from the UEs. The default value of the DPD timers is 3600 seconds and the range is between 1 and 65535 seconds. The default DPD retry interval is 10 seconds, and the range is between 1 and 65535 seconds. The default number of DPD retries is 2, and the range is between 0 and 65535.

**Dead Peer Detection**

The ePDG supports DPD (Dead Peer Detection) protocol messages originating from the ePDG and the WLAN UEs. DPD is performed when no IKE/IPSec packets reach the ePDG within the configured DPD interval. DPD is configured in the crypto template in the ePDG service. The administrator can also disable DPD. However, the ePDG always responds to DPD availability checks initiated by the UE, regardless of the ePDG idle timer configuration.

**Child SA Rekeying**

Rekeying of an IKEv2 Child SA (Security Association) occurs for an already established Child SA whose lifetime is about to exceed a maximum limit. The ePDG initiates rekeying to replace the existing Child SA. The ePDG-initiated rekeying is disabled by default. This is the recommended setting, although rekeying can be enabled using the Crypto Configuration Payload Mode commands.

**Support for MAC Address of WiFi Access Points**

The ePDG can propagate the MAC (Media Access Control) address of each WiFi access point to the P-GW. The ePDG sends this information using the PMIP Location AVP (Attribute-Value Pair) in the User-Location-Info Vendor Specific Option of PBU (Proxy-MIP Binding Update) messages over the S2b interface. In case the protocol used on S2b is GTPv2 then this information is communicated using the Private Extension IE in Create Session Request message.

The WLAN UEs send the MAC address of each WiFi access point to the ePDG embedded in the NAI (Network Access Identifier). When the ePDG receives an NAI that includes a MAC address, the ePDG checks the MAC address and if the validation is successful, the ePDG removes the MAC address from the NAI before sending it to the AAA server in the User-Name AVP of DER (Diameter EAP Request) messages.

Note that the ePDG can be configured to allow IPSec connection establishment without the MAC address present. If the MAC address is not present and the ePDG is configured to check for the MAC address, the ePDG fails the IKE negotiation and returns Notify payload 24 (AUTHENTICATION_FAILED).

**AAA Server Groups**

A value-added feature to enable VPN service provisioning for enterprise or MVNO customers. Enables each corporate customer to maintain its own AAA servers with its own unique configurable parameters and custom dictionaries. This feature provides support for up to 800 AAA server groups and 800 NAS IP addresses that can be provisioned within a
single context or across the entire chassis. A total of 128 servers can be assigned to an individual server group. Up to 1,600 accounting, authentication, and/or mediation servers are supported per chassis.

EAP Authentication

Enables secure user and device level authentication with a 3GPP AAA server or via 3GPP2 AAA proxy and the authenticator in the ePDG.

The ePDG uses the Diameter-based SWm interface to authenticate subscriber traffic with the 3GPP AAA server. Following completion of the security procedures (IKEv2) between the UE and ePDG, the ePDG selects EAP-AKA as the method for authenticating the subscriber session. EAP-AKA uses symmetric cryptography and pre-shared keys to derive the security keys between the UE and EAP server. The ePDG represents the EAP authenticator and triggers the identity challenge-response signaling between the UE and back-end 3GPP AAA server. On successful verification of user credentials, the 3GPP AAA server obtains the Cipher Key and Integrity Key from the HSS. It uses these keys to derive the MSK (Master Session Key) that are returned on EAP-Success to the ePDG. The ePDG uses the MSK to derive the authentication parameters.

After the user credentials are verified by the 3GPP AAA and HSS, the ePDG returns the PDN address obtained from the P-GW (using PMIPv6/GTPv2) to the UE. In the connection establishment procedures, the PDN address is triggered based on subscription information conveyed over the SWm reference interface. Based on the subscription information and requested PDN-Type signaled by the UE, the ePDG informs the P-GW of the type of required address (IPv6 and/or IPv4 Home Address Option for dual IPv4/IPv6 PDNs).

IPv6 Capabilities

IPv6 addressing enables increased address efficiency and relieves pressures caused by the rapidly approaching IPv4 address exhaustion problem.

The ePDG offers the following IPv6 capabilities:

- Support for any combination of IPv4, IPv6, or dual stack IPv4/IPv6 address assignment from address pools on the P-GW.
- Support for native IPv6 transport and service addresses on the PMIPv6/GTPv2 S2b interface with the P-GW.

IPv6 transport is supported on the SWm Diameter AAA interface with the external 3GPP AAA server. Note that the ePDG supports IPv6 transport for the UE-ePDG tunnel endpoints on the SWu interface.

General Call Flow

The following section explains the basic ePDG call flows.
Figure 2. General Call Flow

The UE and the ePDG exchange the first pair of messages, known as IKE_SA_INIT and RSP, in which the ePDG and UE negotiate cryptographic algorithms, exchange nonces and perform a Diffie_Hellman exchange.

Table 4. General Call Flow

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The UE sends IKE_SA_INIT Message.</td>
</tr>
<tr>
<td>2.</td>
<td>ePDG responds with IKE_SA_INIT_RSP Message.</td>
</tr>
<tr>
<td>Step</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>3.</td>
<td>The UE sends the user identity (in the IDi payload) and the APN information (in the IDr payload) in this first message of the IKE_AUTH phase, and begins negotiation of child security associations. The UE omits the AUTH parameter in order to indicate to the ePDG that it wants to use EAP over IKEv2. The user identity shall be compliant with Network Access Identifier (NAI) format specified in TS 23.003 containing the IMSI, as defined for EAP-AKA in RFC 4187. The UE shall send the configuration payload (CFG_REQUEST) within the IKE_AUTH request message to obtain an IPv4 home IP Address and/or a Home Agent Address. When the MAC ULI feature is enabled, the root NAI used will be of the form “0&lt;IMSI&gt;@AP_MAC_ADDR:epc.mnc&lt;MNC&gt;.mcc&lt;MCC&gt;.3gppnetwork.org”.</td>
</tr>
<tr>
<td>4.</td>
<td>The ePDG sends the Authentication and Authorization Request message to the 3GPP AAA Server, containing the user identity and APN.</td>
</tr>
<tr>
<td>5.</td>
<td>The 3GPP AAA Server shall fetch the user profile and authentication vectors from HSS/HLR (if these parameters are not available in the 3GPP AAA Server). The 3GPP AAA Server shall lookup the IMSI of the authenticated user based on the received user identity (root NAI) and include the EAP-AKA as requested authentication method in the request sent to the HSS. The HSS shall then generate authentication vectors with AMF separation bit = 0 and send them back to the 3GPP AAA server. The 3GPP AAA Server checks in user's subscription if he/she is authorized for non-3GPP access. The counter of IKE SAs for that APN is stepped up. If the maximum number of IKE SAs for that APN is exceeded, the 3GPP AAA Server shall send an indication to the ePDG that established the oldest active IKE SA (it could be the same ePDG or a different one) to delete the oldest established IKE SA. The 3GPP AAA Server shall update accordingly the information of IKE SAs active for the APN. The 3GPP AAA Server initiates the authentication challenge. The user identity is not requested again.</td>
</tr>
<tr>
<td>6.</td>
<td>The ePDG responds with its identity, a certificate, and sends the AUTH parameter to protect the previous message it sent to the UE (in the IKE_SA_INIT exchange). It completes the negotiation of the child security associations if any. The EAP message received from the 3GPP AAA Server (EAP-Request/AKA-Challenge) is included in order to start the EAP procedure over IKEv2.</td>
</tr>
<tr>
<td>7.</td>
<td>The UE checks the authentication parameters and responds to the authentication challenge. The only payload (apart from the header) in the IKEv2 message is the EAP message.</td>
</tr>
<tr>
<td>8.</td>
<td>The ePDG forwards the EAP-Response/AKA-Challenge message to the 3GPP AAA Server.</td>
</tr>
<tr>
<td>8a</td>
<td>The AAA checks, if the authentication response is correct.</td>
</tr>
<tr>
<td>9.</td>
<td>When all checks are successful, the 3GPP AAA Server sends the final Authentication and Authorization Answer (with a result code indicating success) including the relevant service authorization information, an EAP success and the key material to the ePDG. This key material shall consist of the MSK generated during the authentication process. When the SWm and SWd interfaces between ePDG and 3GPP AAA Server are implemented using Diameter, the MSK shall be encapsulated in the EAP-Master-Session-Key-AVP, as defined in RFC 4072.</td>
</tr>
<tr>
<td>10.</td>
<td>The MSK shall be used by the ePDG to generate the AUTH parameters in order to authenticate the IKE_SA_INIT phase messages, as specified for IKEv2 in RFC 4306. These two first messages had not been authenticated before as there was no key material available yet. According to RFC 4306 [3], the shared secret generated in an EAP exchange (the MSK), when used over IKEv2, shall be used to generated the AUTH parameters.</td>
</tr>
<tr>
<td>11.</td>
<td>The EAP Success/Failure message is forwarded to the UE over IKEv2.</td>
</tr>
<tr>
<td>12.</td>
<td>The UE takes its own copy of the MSK as input to generate the AUTH parameter to authenticate the first IKE_SA_INIT message. The AUTH parameter is sent to the ePDG.</td>
</tr>
<tr>
<td>12a</td>
<td>The ePDG checks the correctness of the AUTH received from the UE. At this point the UE is authenticated.</td>
</tr>
</tbody>
</table>
## Features and Functionality

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.</td>
<td>On successful authentication the ePDG selects the P-GW based on Node Selection options. The ePDG sends Create Session Request (IMSI, [MSISDN], Serving Network, RAT Type (WLAN), Indication Flags, Sender F-TEID for C-plane, APN, Selection Mode, PAA, APN-AMBR,Bearer Contexts, [Recovery], [Charging characteristics], [Additional Protocol Configuration Options (APCO)], Private IE (P-CSCF, AP MAC address). Indication Flags shall have Dual Address Bearer Flag set if PDN Type is IPv4v6. Handover flag shall be set to Initial or Handover based on the presence of IP addresses in the IPv4/IPv6_Address configuration requests. Selection Mode shall be set to “MS or network provided APN, subscribed verified”. The MSISDN, Charging characteristics, APN-AMBR and bearer QoS shall be provided on S2b interface by ePDG when these are received from AAA on SWm interface. The control plane TEID shall be per PDN connection and the user plane TEID shall be per bearer created.</td>
</tr>
<tr>
<td>14.</td>
<td>The P-GW allocates the requested IP address session and responds back to the ePDG with a Create Session Response (Cause, P-GW S2b Address C-plane, PAA, APN-AMBR, [Recovery], Bearer Contexts Created, [Additional Protocol Configuration Options (APCO)], Private IE (P-CSCF)) message.</td>
</tr>
<tr>
<td>15.</td>
<td>The ePDG calculates the AUTH parameter which authenticates the second IKE_SA_INIT message.</td>
</tr>
<tr>
<td>16.</td>
<td>The ePDG sends the assigned Remote IP address in the configuration payload (CFG_REPLY). The AUTH parameter is sent to the UE together with the configuration payload, security associations and the rest of the IKEv2 parameters and the IKEv2 negotiation terminates.</td>
</tr>
<tr>
<td>17.</td>
<td>Router Advertisement will be sent for IPv6 address assignments, based on configuration.</td>
</tr>
</tbody>
</table>

### P-GW Selection

The P-GW selection function enables the ePDG to allocate a P-GW to provide PDN connectivity to the WLAN UEs in the untrusted non-3GPP IP access network. The P-GW selection function can employ either static or dynamic selection.

### Static Selection

The PDN-GW-Allocation-Type AVP indicates whether the P-GW address is statically allocated or dynamically selected by other nodes, and is considered only if MIP6-Agent-Info is present. When the PDN-GW-Allocation-Type AVP is absent or is STATIC, and an initial attach occurs, or is DYNAMIC and a handoff attach occurs, the ePDG performs static selection of the P-GW.

The figure below shows the message exchange for static selection. The table that follows the figure describes each step in the flow.
Table 5. P-GW Static Selection

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The AAA server sends the P-GW FQDN (Fully Qualified Domain Name) to the ePDG.</td>
</tr>
<tr>
<td>2.</td>
<td>The ePDG receives the P-GW FQDN from the AAA server as part of the MIP-Home-Agent-Host AVP in a Diameter EAP Answer message. The ePDG removes the first two labels of the received P-GW FQDN (if the FQDN starts with 'topon') to obtain the Canonical Node Name ID of the P-GW. The ePDG uses this P-GW ID to send an S-NAPTR (Server-Name Authority Pointer) query to the DNS proxy.</td>
</tr>
<tr>
<td>3.</td>
<td>The DNS proxy sends the S-NAPTR query to the DNS.</td>
</tr>
<tr>
<td>4.</td>
<td>The DNS may return multiple NAPTR resource records with an ‘A’ flag (for an address record) with the same or different service parameters.</td>
</tr>
<tr>
<td>5.</td>
<td>The DNS proxy forwards the two NAPTR resource records to the ePDG.</td>
</tr>
<tr>
<td>6.</td>
<td>The ePDG selects the replacement string (the P-GW FQDN) that matches the service parameter if ePDG is configured as MAG for PMIPv6 protocol or service parameter 'x-3gpp-pgw:x-s2b-gtp' when ePDG is configured for GTP protocol support. The ePDG then performs an A/AAAA query with the selected replacement string (the P-GW FQDN).</td>
</tr>
<tr>
<td>7.</td>
<td>The DNS proxy sends the A/AAAA query to the DNS.</td>
</tr>
<tr>
<td>8.</td>
<td>The DNS returns the IP address of the P-GW.</td>
</tr>
<tr>
<td>9.</td>
<td>The DNS proxy forwards the P-GW IP address to the ePDG.</td>
</tr>
</tbody>
</table>
Dynamic Selection

For a given APN, when the HSS returns Dynamic Allocation Allowed for the P-GW ID and the selection is not for a 3GPP-to-non-3GPP handover, the ePDG ignores the P-GW ID and instead performs dynamic selection.

The figure below shows the message exchange for dynamic selection. The table that follows the figure describes each step in the flow.

Figure 4. P-GW Dynamic Selection

Table 6. P-GW Dynamic Selection

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The WLAN UE sends the APN name to the ePDG.</td>
</tr>
<tr>
<td>2.</td>
<td>The ePDG constructs the APN FQDN from the received APN name. The ePDG uses this query string to send an S-NAPTR (Server-Name Authority Pointer) query to the DNS proxy.</td>
</tr>
<tr>
<td>3.</td>
<td>The DNS proxy sends the S-NAPTR query to the DNS.</td>
</tr>
<tr>
<td>4.</td>
<td>The DNS may return multiple NAPTR resource records with an ‘S’ flag (for SRV records) with the same or different service parameters.</td>
</tr>
<tr>
<td>5.</td>
<td>The DNS proxy forwards the NAPTR resource records to the ePDG.</td>
</tr>
</tbody>
</table>
Step | Description |
--- | --- |
6. | The ePDG selects the replacement strings (the APN FQDNs) that matches the service parameter if ePDG is configured as MAG for PMIPv6 protocol or service parameter ‘x-3gpp-pgw:x-s2b-gtp’ when ePDG is configured for GTP protocol support. The ePDG then performs a DNS SRV query with a replacement string (the APN FQDN) for each of the selected replacement strings. |
7. | The DNS proxy sends each DNS SRV query to the DNS. |
8. | For each SRV query, the DNS returns the SRV resource records with the target strings. |
9. | The DNS proxy forwards the SRV response to the ePDG. The ePDG compares the P-GW FQDNs against the configured ePDG FQDN and selects longest suffix matching entry. |
10. | The ePDG performs an A/AAAA query with the selected P-GW FQDN. |
11. | The DNS proxy sends the A/AAAA query to the DNS. |
12. | The DNS returns the IP address of the P-GW. |
13. | The DNS proxy forwards the P-GW IP address to the ePDG. |

**P-GW Initiated Bearer Modification**

The following section covers the P-GW initiated default/dedicated bearer modification procedure.

**Figure 5. P-GW initiated bearer modification**

![Diagram of P-GW initiated bearer modification]

**Table 7. P-GW initiated bearer modification**

| Step | Description |
--- | --- |
Evolved Packet Data Gateway Overview

Features and Functionality

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>If dynamic PCC is deployed, the PCRF sends a PCC decision provision (QoS policy) message to the PDN GW. This corresponds to the initial steps of the PCRF-Initiated IP CAN Session Modification procedure or to the PCRF response in the PCEF initiated IP-CAN Session Modification procedure, up to the point that the PDN GW requests IP CAN Bearer Signalling. If dynamic PCC is not deployed, the PDN GW may apply local QoS policy.</td>
</tr>
<tr>
<td>2.</td>
<td>The PDN GW uses this QoS policy to determine that a service data flow shall be aggregated to or removed from an active S2b bearer or that the authorized QoS of a service data flow has changed. The PDN GW generates the TFT and updates the EPS Bearer QoS to match the traffic flow aggregate. The PDN GW then sends the Update Bearer Request (APN AMBR, Bearer Context (EPS Bearer Identity, EPS Bearer QoS, TFT)) message to the ePDG.</td>
</tr>
<tr>
<td>3.</td>
<td>The ePDG uses the uplink packet filter (UL TFT) to determine the mapping of traffic flows to the S2b bearer and acknowledges the S2b bearer modification to the P-GW by sending an Update Bearer Response (EPS Bearer Identity) message. Also the QCI values received in QoS shall be updated and utilized for the UL traffic DSCP mapping/marking.</td>
</tr>
</tbody>
</table>

Topology/Weight-based Selection

Topology/weight-based selection uses DNS requests to enable P-GW load balancing based on topology and/or weight.

For topology-based selection, once the DNS procedure outputs a list of P-GW hostnames for the APN FQDN, the ePDG performs a longest-suffix match and selects the P-GW that is topologically closest to the ePDG and subscriber. If there are multiple matches with the same suffix length, the Weight and Priority fields in the NAPTR resource records are used to sort the list. The record with the lowest number in the Priority field is chosen first, and the Weight field is used for those records with the same priority.

For weight-based selection, once the DNS procedure outputs a list of P-GW hostnames for the APN FQDN, if there are multiple entries with same priority, calls are distributed to these P-GWs according to the Weight field in the resource records. The Weight field specifies a relative weight for entries with the same priority. Larger weights are given a proportionately higher probability of being selected. The ePDG uses the value of (65535 minus NAPTR preference) as the statistical weight for NAPTR resource records in the same way as the SRV weight is used for SRV records, as defined in RFC 2782.

When both topology-based and weight-based selection are enabled on the ePDG, topology-based selection is performed first, followed by weight-based selection. A candidate list of P-GWs is constructed based on these, and the ePDG selects a P-GW from this list for call establishment. If the selected P-GW does not respond, the ePDG selects the alternate P-GW(s) from the candidate list.

Dual Stack Support

The ePDG supports PDN type IPv4v6. The ePDG handles traffic originating from both IPv4 and IPv6 UE addresses based on configured traffic selectors. Here Dual stack is mentioned for subscriber traffic (inner IP packets).

The ePDG determines the PDN type based on the requested IP address versions sent from the UE in the CP payload (CFG_REQUEST) within the IKE_AUTH Request message. The ePDG sets the IPv6 Home Network Prefix option and IPv4 Home Address Request option parameters when sending the PBU (Proxy-MIP Binding Update) message to the P-GW, specifying the PDN type as IPv4v6. In case the protocol used on S2b is GTPv2 then the ePDG sets the PDN Type inside PAA (PDN Address Allocation) as IPv4v6 and sends the same in Create Session Request Message to the P-GW. The ePDG sends the addresses allocated by the P-GW in the PBA (Proxy-MIP Binding Acknowledgement) / Create Session Response message to the UE via the CP payload (CFG_REPLY) in the IKE_AUTH Response message.
Inter-access Handover Support

The ePDG supports inter-access handovers between two different interfaces, such as a handover between a 3GPP network and an untrusted non-3GPP IP access network, or between two untrusted non-3GPP IP access networks.

When a UE sends an IKE_AUTH Request message with a NULL IPv4/IPv6 address in the CP payload, the ePDG determines that the request is for an initial attach. When a message contains non-null IP address values, the ePDG determines that the request is for a handover attach. On the SWu interface, the UE populates the INTERNAL_IP4_ADDRESS and/or INTERNAL_IP6_ADDRESS parameter with the previously-assigned IP addresses to indicate that UE supports IP address preservation for handovers.

In case the protocol used on S2b is PMIPv6, per 3GPP TS 29.275, the ePDG indicates an inter-access handover in the S2b Handoff Indicator option of PBU (Proxy-MIP Binding Update) messages. Per RFC 5213, the ePDG indicates the RAT (Radio Access Technology) of untrusted non-3GPP access network in the Access Technology Type option.

In case the protocol used on S2b is GTPv2 then per 3GPP TS 29.274, the ePDG indicates an inter-access handover in the indication flags IE.

Mobile Access Gateway Function

The ePDG hosts a MAG (Mobile Access Gateway) function, which acts as a proxy mobility agent in the E-UTRAN/EPC network and uses Proxy Mobile IPv6 signaling to provide network-based mobility management on behalf of the UEs attached to the network. The P-GW also uses Proxy Mobile IPv6 signaling to host an LMA (Local Mobility Anchor) function to provide network-based mobility management. With this approach, the attached UEs are no longer involved in the exchange of signaling messages for mobility.

The MAG function on the ePDG and the LMA function on the P-GW maintain a single shared tunnel. To distinguish between individual subscriber sessions, separate GRE keys are allocated in the PBU (Proxy-MIP Binding Update) and PBA (Proxy-MIP Binding Acknowledgement) messages between the ePDG and the P-GW. If the Proxy Mobile IP signaling contains PCOs (Protocol Configuration Options), it can also be used to transfer P-CSCF or DNS addresses.

The S2b interface uses IPv6 for both control and data. During PDN connection establishment, the P-GW uses Proxy Mobile IPv6 signaling to allocate the IPv6 HNP (Home Network Prefix) to the ePDG, and the ePDG returns the HNP to the UE in an IPv6 router advertisement.

Note that the MAG function on the ePDG does not support multiple PDN connections for the same APN and UE combination. The ePDG establishes each subsequent connection from the same UE to the same APN via a new session and deletes the previous session before the new session gets established.

IPv6 Router Advertisement Support

The ePDG provides router advertisement support for IPv6 and dual stack PDNs to allow the WLAN UEs to initialize the IPv6 protocol stack. The ePDG sends an unsolicited router advertisement to the UE for an IPv6 PDN connection after sending the final IKE_AUTH Response message. When the ePDG receives a Router Solicitation Request message from the UE, the ePDG intercepts the message and responds to it. This is needed for some UEs that perform address auto-configuration despite receiving the IP address information through the CP payload of the IKE_AUTH Response message.
DNS Request Support

During IPSec tunnel establishment, the WLAN UEs can request an IP address for the DNS in the CP payload (CFG_REQUEST). The ePDG retrieves the request from the CFG_REQUEST attribute of the first IKE_AUTH message exchange and includes it in the PBU (Proxy-MIP Binding Update) message sent to the P-GW.

The ePDG sends the PBU message by framing the MIPv6 APCO VSE (Additional Protocol Configuration Options Vendor Specific Extension) with an IPv6 and/or IPv4 DNS request to the P-GW. Once the response is received from the P-GW with the list of IPv6 and/or IPv4 DNS addresses in the returned MIPv6 APCO VSE, the ePDG includes the final address(es) in the CP payload (CFG_REPLY) of the final IKE_AUTH Response message sent to the UE.

In case the Protocol used on S2b is GTPv2 then APCO is used in Create Session Request message for requesting the IPv4 or IPv6 DNS server address request and then P-GW communicates the DNS server addresses in the APCO IE in the Create Session Response Message, the ePDG includes the final address(es) in the CP payload (CFG_REPLY) of the final IKE_AUTH Response message sent to the UE.

Note that the ePDG includes a maximum of two IPv4 DNS addresses and/or a maximum of two IPv6 DNS addresses in the CP payload (CFG_REPLY).

P-CSCF Request Support

To connect to the IMS core network, the WLAN UEs perform P-CSCF discovery as part of session establishment. This feature supports P-CSCF attributes in CFG_REQUEST and CFG_REPLY messages as part of the CP payload in the IKE_AUTH Request and Response messages the ePDG sends and receives from the UEs. The P-CSCF attribute can be sent on SWu as private or with standard value.

The WLAN UEs request a P-CSCF address in IKE_AUTH messages to establish IMS PDN connections. The ePDG receives the P-CSCF attribute in the CP payload (CFG_REQUEST) of the first IKE_AUTH message exchange and includes a P-CSCF Request message in the PBU (Proxy-MIP Binding Update) message to the P-GW. The ePDG sends the PBU message by framing the MIPv6 PCO VSE (Protocol Configuration Options Vendor Specific Extension) within the P-CSCF Request message to the P-GW. Once the ePDG receives the response from the PGW with the list of P-CSCF addresses, the ePDG shall include the P-CSCF addresses in the CP payload (CFG_REPLY) of the final IKE_AUTH Response message sent to the UE.

In case protocol used on S2b is GTPv2 ePDG has flexibility to use either APCO IE or Private Extension IE based on ePDG configuration. Once the ePDG receives the response from the P-GW with the list of P-CSCF addresses, the ePDG includes the P-CSCF addresses in the CP payload (CFG_REPLY) of the final IKE_AUTH Response message sent to the UE.

On SWu interface the ePDG is able to handle the private attribute value for the P-CSCF address and this private attribute value is configurable on ePDG. By default 16384 is used for P-CSCF IPv4 address and 16390 is used for the IPv6 P-CSCF address. The values 16384-32767 are for private use among mutually consenting parties.

The P-CSCF v4 and v6 are recently assigned values by IANA so ePDG shall be supporting those values as well in addition to the private configured value. ePDG should respond to UE with same attribute value as received in the request. Private values are maintained for the devices which are already in market as they may not comply to standard values.

UE should include P-CSCF_V4_ADDR attribute only once in IKE_AUTH request and no specific P-CSCF address is included because it is a request. ePDG is enhanced to support both IPv4 and IPv6 P-CSCF address handling together. ePDG also supports maximum of 3 IPv4 and 3 IPv6 P-CSCF addresses. The exceeding P-CSCF address will be ignored. In case of invalid P-CSCF address are received the P-CSCF address is ignored and have no impact on the call establishment.

On S2b interface the P-CSCF is enhanced to support both APCO IE and private Extension IE. ePDG continues to use existing "vendor-specific-attribute" configuration present under epdg-service to decide whether to use APCO IE or...
private extension IE. The feature scope shall be limited to GTPv2 and shall not cover PMIPv6 as most of the customers are showing interest in GTPv2 based deployment.

**Multiple PDN Support**

The multiple PDN feature enables the WLAN UEs to simultaneously establish multiple PDN connections towards the P-GW. Each PDN connection has a separate IKE tunnel established between the UE and the ePDG.

Note that the ePDG supports multiple PDN connections to different APNs only and multiple PDN connections from the same UE to the same APN are not allowed. The ePDG establishes each subsequent connection from the same UE to the same APN via a new session and deletes the previous session before the new session gets established. These new PDN connections use different IPSec/PMIPv6/GTPv2 tunnels.

To request a new session, the UE sends the APN information (in the IDr payload) along with the user identity (in the IDi payload) in this first IKE_AUTH Request message, and begins negotiation of Child SAs. The ePDG sends the new APN information in the Service Selection Mobility Option towards the P-GW, which treats each MN-ID+APN combination as a separate binding and allocates a new IP address/prefix for each new binding.

In case of S2b protocol being used as GTPv2 IMSI + APN is used for identifying the unique session.

**Default APN Support**

The ePDG supports a default APN when APN information is not available from the WLAN UEs over the SWu interface.

When the APN information is received from the WLAN UEs, the information is sent towards the AAA server via DER (Diameter EAP Request) messages. When the APN information is absent, the AAA server provides the default APN to the ePDG in a DEA (Diameter EAP Answer) message.

**Congestion Control**

The congestion control feature allows you to set policies and thresholds and specify how the system reacts when faced with a heavy load condition.

The congestion control feature monitors the system for conditions that could potentially degrade performance when the system is under heavy load. Typically, these conditions are temporary (for example, high CPU or memory utilization) and are quickly resolved. However, continuous or large numbers of these conditions within a specific time interval may have an impact on the system’s ability to service subscriber sessions. Congestion control helps identify such conditions and invokes policies for addressing the situation.

Congestion control operation is based on configuring the following:

- **Congestion Condition Thresholds**: Thresholds dictate the conditions for which congestion control is enabled and establishes limits for defining the state of the system (congested or clear). These thresholds function in a way similar to operation thresholds that are configured for the system as described in the Thresholding Configuration Guide. The primary difference is that when congestion thresholds are reached, a service congestion policy and an SNMP trap, starCongestion, are generated. A threshold tolerance dictates the percentage under the configured threshold that must be reached in order for the condition to be cleared. An SNMP trap, starCongestionClear, is then triggered.

- **Port Utilization Thresholds**: If you set a port utilization threshold, when the average utilization of all ports in the system reaches the specified threshold, congestion control is enabled.
- **Port-specific Thresholds**: If you set port-specific thresholds, when any individual port-specific threshold is reached, congestion control is enabled system-wide.

- **Service Congestion Policies**: Congestion policies are configurable for each service. These policies dictate how services respond when the system detects that a congestion condition threshold has been crossed. The ePDG supports congestion policies to either drop or reject new calls when congestion is detected in the system.

The congestion control overload disconnect feature can also be enabled for disconnecting passive calls during an overload situation. The ePDG selects passive calls based on the overload disconnect configuration options.

The following table lists the `congestion-control threshold` command options supported on the ePDG in this release.

### Table 8. Supported Congestion Control Threshold Command Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>license-utilization <code>percent</code></td>
<td>The percent utilization of licensed session capacity as measured in 10 second intervals. <em>percent</em> can be configured to any integer value from 0 to 100. Default: 100</td>
</tr>
<tr>
<td>max-sessions-per-service-utilization <code>percent</code></td>
<td>The percent utilization of the maximum sessions allowed per service as measured in real time. This threshold is based on the maximum number of sessions or PDP contexts configured for the a particular service. <em>percent</em> can be an integer from 0 through 100. Default: 80</td>
</tr>
<tr>
<td>port-rx-utilization <code>percent</code></td>
<td>The average percent utilization of port resources for all ports by received data as measured in 5 minute intervals. <em>percent</em> can be an integer from 0 through 100. Default: 80</td>
</tr>
</tbody>
</table>
| port-specific \{ slot/port | all \} \{ rx-utilization `percent` \} \{ tx-utilization `percent` \} | Sets port-specific thresholds. If you set port-specific thresholds, when any individual port-specific threshold is reached, congestion control is applied system-wide.
`slot/port`: Specifies the port for which port-specific threshold monitoring is being configured. The slot and port must refer to an installed card and port.
`all`: Set port specific threshold monitoring for all ports on all cards.
`rx-utilization `percent`: Default 80%. The average percent utilization of port resources for the specified port by received data as measured in 5 minute intervals. *percent* must an integer from 0 through 100.
`tx-utilization `percent`: Default 80%. The average percent utilization of port resources for the specified port by transmitted data as measured in 5 minute intervals. *percent* must an integer from 0 through 100. Default: Disabled |
<p>| port-tx-utilization <code>percent</code> | The average percent utilization of port resources for all ports by transmitted data as measured in 5 minute intervals. <em>percent</em> can be an integer from 0 through 100. Default: 80 |
| service-control-cpu-utilization <code>percent</code> | The average percent utilization of CPUs on which a Demux Manager software task instance is running as measured in 10-second intervals. <em>percent</em> can be an integer from 0 through 100. Default: 80 |</p>
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>system-cpu-utilization</strong> percent</td>
<td>The average percent utilization for all PSC2 CPUs available to the system as measured in 10-second intervals. <em>percent</em> can be an integer from 0 through 100. Default: 80</td>
</tr>
<tr>
<td><strong>system-memory-utilization</strong> percent</td>
<td>The average percent utilization of all CPU memory available to the system as measured in 10-second intervals. <em>percent</em> can be an integer from 0 through 100. Default: 80</td>
</tr>
</tbody>
</table>

**Important:** For more information on congestion control, including configuration instructions, see the System Administration Guide. For more information on the congestion-control threshold command, see the eHRPD/LTE Command Line Interface Reference.

### Session Recovery Support

Session recovery provides seamless failover and reconstruction of subscriber session information in the event of a hardware or software fault within the system, preventing a fully connected user session from being disconnected. The ePDG supports session recovery for IPv4, IPv6, and IPv4/v6 sessions and ensures that data and control planes are re-established as they were before the recovery procedure.

When session recovery occurs, the system reconstructs the following subscriber information:

- Data and control state information required to maintain correct call behavior, including DNS, P-GW, and P-CSCF addresses.
- Subscriber data statistics that are required to ensure that accounting information is maintained.
- A best-effort attempt to recover various timer values, such as call duration, absolute time, and others.

Note that for the recovered sessions, the ePDG recreates counters only and not statistics.

Session recovery is also useful for in-service software patch upgrade activities. If session recovery is enabled during the software patch upgrade, it helps to preserve existing sessions on the active hardware during the upgrade process.

**Important:** For more information on session recovery support, see the System Administration Guide.

### ICSR Support

The ePDG supports ICSR with fault detection and automatic switch over. The subscriber session details for all ePDG interfaces are replicated in stand by, In case of a switchover, the new chassis processes all subsequent control and data traffic for the subscriber session.

The SWu, SWm and S2b interface are not impacted by the switchovers.

ePDG release 18.0 supports upgrade/down grade from release 18 (N) to 16 (N-2).

**Important:** For more information on ICSR, see the System Administration Guide.
P-GW selection

The ePDG selects P-GW node based one of the logic:

- eDNS
- DNS over TCP
- P-GW re-selection on session timeout
- PGW re-selection on call attempt failure due to PGW reject

eDNS

The ePDG supports extended DNS client to handle DNS response larger than 512 bytes.

RFC 1035 limits the size of DNS responses over UDP to 512 bytes. If P-GW discovery is done via DNS, there is a chance of 512 byte limit is hit as there are multiple P-GWs supporting an APN consequently having multiple responses to the DNS query, resulting in truncation of the RRs.

Extended the DNS (RFC 2671) allows the client to advertise a bigger re-assemble buffer size to the DNS server so that the server can send a response bigger than 512 bytes. An interim solution to the truncation issue is to arrange the RRs hierarchically so that the limit is never hit.

DNS over TCP

By default DNS client communicates with the server over UDP port. The client can support eDNS, DNS responses up to 4 K Bytes in size from the server. If FQDN resolves too many RRs, the 4 KB limit could be exhausted.

Use the following approach to resolve this issue:

Use TCP port when the server needs to send bigger responses (up to 64 KB), this needs to be driven by the client. When the server indicates that it is not able to send all the answers to a query by setting the truncation bit in the response header. The client on seeing this would switch to TCP port and re-sends the same query. The client continues to use UDP port for new requests.

P-GW re-selection on session timeout

During dynamic P-GW node selection by ePDG, if the selected P-GW is unreachable, the ePDG will select the next P-GW entry from the P-GW candidate list returned during the S-NAPTR procedure to set up the PDN connection.

PGW re-selection on call attempt failure due to PGW reject

ePDG attempts to select alternate PGW when the first PGW has rejected the call with the below error causes. Maximum alternate PGW selection attempts(0-64) can be configured per APN profile using CLI, default is 3.

- EGTP_CAUSE_ALL_DYNAMIC_ADDR_OCCUPIED (0x54)
- EGTP_CAUSE_NO_RESOURCES_AVAILABLE(73)
- EGTP_CAUSE_SERVICE_DENIED (0x59),
- EGTP_CAUSE_PEER_NOT_RESPONDING (100)
- EGTP_CAUSE_SERVICE_NOT_SUPPORTED (0x44)

S2b GTPv2 support

ePDG supports PDN connection, session establishment and release, along with support for dedicated bearer creation, deletion and modification that is initiated by the P-GW.
During the initial attachment, the ePDG “default EPS QOS”, and “APN-AMBR” values are populated in the create session request based on the values received from the SWm interface. If these values are missing in the messages received on the SWm interface, ePDG encodes the mandatory or conditional IE with the values set to zero.

When a new PDN connection is established, ePDG allocates and sends a default EPS bearer ID to the PDN gateway. After the initial attach, a default bearer is created for the session, and the IP address is allocated and communicated to the UE.

A GTP-C and GTP-U tunnel is successfully established between the ePDG and P-GW, and an IPSec tunnel is established between the UE and ePDG. Traffic is allowed to flow between these established tunnels.

ePDG sends a “delete session request” message to P-GW, and handles the corresponding “delete session response” message from the P-GW during the following scenarios:

- UE/ePDG initiated detach with GTP on S2b
- UE requested PDN disconnection with GTP on S2b
- AAA initiated detach with GTP on S2b

ePDG handles the received “create bearer request” message and sends a “create bearer response” message for the dedicated bearer creation triggered from the P-GW.

After the dedicated bearer is created, a new GTP-U tunnel is established between ePDG and P-GW, and traffic mapping to the TFT of this bearer occurs. ePDG supports up to 16 packet filters per bearer.

ePDG also stores mapping information between the uplink packet filters received from the P-GW (For example; in the Create Bearer Request message), and the corresponding S2b bearer. ePDG matches these filters and decides if the uplink packets should be allowed or dropped.

ePDG receives the “delete bearer request” message and sends a “delete bearer response” message for the dedicated bearer deletion triggered by the P-GW.

ePDG clears the bearer path (GTP-U tunnel) corresponding to the EBI received. In the case of a linked EBI, the PDN connection and its associated bearers are deleted. The TFT mapping for the deleted bearer is also deleted.

ePDG handles the received “update bearer request” message and sends a “update bearer response” message for dedicated bearer modification triggered from the P-GW. ePDG updates the UL TFT mapping for the associated bearer using the “bearer context” information.

ePDG supports path failure detection for control plane by using Echo Request and Echo Response messages. A peer's IP address-specific counter is reset every time an Echo Response message is received from the peer's IP address. The counter is incremented when the T3-RESPONSE timer expires for an Echo Request message sent to the peer's IP address. The path is considered as down if the counter exceeds the value of N3-REQUESTS.

ePDG initiates the Echo requests once retransmission timeout occurs for the request sent to the P-GW. The retransmission for GTP messages is handled by running the retransmission timer (T3-RESPONSE) and for N3-REQUESTS timer, the message is retransmitted after the retransmission timer expires. After all the retransmissions are over, echo handling is initiated.

The GTPC configuration has the configuration command, no gtpc path-failure detection-policy <CR> using which on path failure detection, SNMP traps/alarms are generated notifying that P-GW has gone down, but the sessions are not deleted. The SNMP trap is sent only once per peer, and not for every session. When this command is not configured, path failure detection and the subsequent cleanup action is enabled by default.

Detection of path failure for user plane is supported using the Echo Request/ Echo Response messages. A path counter is reset every time an Echo Response is received and incremented when the T3-RESPONSE timer expires for any Echo Request message sent. The path is considered as down if the counter exceeds the value of N3-REQUESTS.
**DSCP and 802.1P Marking**

The ePDG can assign DSCP levels to specific traffic patterns in order to ensure that the data packets can be delivered according to the precedence with which they are tagged. The DiffServ markings can be applied to the IP header of the every subscriber data packet transmitted over the SWu and the S2b[GTPv2] interface.

The specific traffic patterns are classified as per their associated QCI/ARP value on the GTP-tunnel. Data packets falling under the category of each of the traffic patterns are tagged with a DSCP marking.

For uplink traffic, i.e. traffic from ePDG to P-GW through GTP tunnel, DSCP markings can be configured using global qci-qos mapping configuration association in ePDG service. In this case, only outer IP header is used for routing the packet over GTP-u’ interface. Hence TOS field of only outer IP header is changed, i.e. subscriber packet is not marked with DSCP value at ePDG.

ePDG service does have configuration for association of the global configured qci-qos mapping and further in global qci-qos mapping configuration its expected that encaps-header configuration for dscp marking shall be used for setting the TOS value in the outer IP header.

Following is the global configuration under qci-qos mapping:

```
qci num [ uplink { encaps-header { copy-inner | dscp-marking hex } | 802.1p-value num }]`
```

The 802.1p marking shall be done on the uplink traffic per the qci-qos mapping global configuration corresponding to the map configured under ePDG service. This is similar configuration as described above for DSCP marking.

The 802.1p marking shall be done in the “user priority” bits of the “TAG” field in the 802.1q tagged frame.

ePDG also supports:

- DSCP marking of Data Packets in uplink (UE->ePDG->PGW) using qci-qos mapping configuration which can be associated to epdg-service
- ePDG marking the inner IP packet DSCP value received from PGW to the outer ESP header in SWu interface
- DSCP marking of Signaling packets (GTPC, on S2b interface) using CLI in egtp-service configuration
- DSCP marking of diameter packets using CLI in Diameter Endpoint configuration

**IPSec Cookie Threshold**

The ePDG supports IKEv2 Cookie challenge payload, this feature helps protect against opening too many half opened IPSec sessions.

The IKEv2 Cookie feature when enabled will invoke a cookie challenge payload mechanism which ensures that only legitimate subscribers are initiating the IKEv2 tunnel request and not a spoofed attack. Note that this configuration is per ipsecmgr.

The Cookie Challenge mechanism is disabled by default, the number of half open connections over which cookie challenge gets activated is also configurable.
Threshold Crossing Alerts

Thresholding on the system is used to monitor the system for conditions that could potentially cause errors or outages. Typically, these conditions are temporary (high CPU utilization or packet collisions on a network, for example) and are quickly resolved. However, continuous or large numbers of these error conditions within a specific time interval may be indicative of larger, more severe issues. The purpose of thresholding is to help identify potentially severe conditions so that immediate action can be taken to minimize and/or avoid system downtime.

The system supports threshold crossing alerts for certain key resources such as CPU, memory, etc. With this capability, the operator can configure a threshold on these resources whereby, should the resource depletion cross the configured threshold, an SNMP trap would be sent.

The following thresholding models are supported by the system:

- **Alert**: A value is monitored and an alert condition occurs when the value reaches or exceeds the configured high threshold within the specified polling interval. The alert is generated, then generated and/or sent again at the end of the polling interval.

- **Alarm**: Both high and low threshold are defined for a value. An alarm condition occurs when the value reaches or exceeds the configured high threshold within the specified polling interval. The alert is generated, then generated and/or sent again at the end of the polling interval.

Thresholding reports conditions using one of the following mechanisms:

- **SNMP traps**: SNMP traps have been created that indicate the condition (high threshold crossing and/or clear) of each of the monitored values. Generation of specific traps can be enabled or disabled on the chassis, ensuring that only important faults get displayed. SNMP traps are supported in both Alert and Alarm modes.

- **Logs**: The system provides a facility for which active and event logs can be generated. As with other system facilities, logs are generated messages pertaining to the condition of a monitored value and are generated with a severity level of WARNING. Logs are supported in both the Alert and the Alarm models.

- **Alarm System**: High threshold alarms generated within the specified polling interval are considered outstanding until a condition no longer exists or a condition clear alarm is generated. Outstanding alarms are reported to the system.
system’s alarm subsystem and are viewable through the Alarm Management menu in the Web Element Manager.

**Important:** For more information about threshold crossing alerts, see the *Thresholding Configuration Guide*.

## Bulk Statistics Support

The system's support for bulk statistics allows operators to choose to view not only statistics that are of importance to them, but also to configure the format in which it is presented. This simplifies the post-processing of statistical data since it can be formatted to be parsed by external, back-end processors.

The system can be configured to collect bulk statistics and send them to a collection server called a receiver. Bulk statistics are collected in a group. The individual statistics are grouped by schema. The following is a partial list of supported schema:

- **ePDG:** Provides statistics to support the ePDG.
- **ePDG-APN:** Provides statistics to support the ePDG APN level statistics
- **System:** Provides system-level statistics.
- **Card:** Provides card-level statistics.
- **Port:** Provides port-level statistics.

The system supports the configuration of up to four sets of receivers. Each set can have primary and secondary receivers. Each set can be configured to collect specific sets of statistics from the various schema. Bulk statistics can be periodically transferred, based on the transfer interval, using ftp/tftp/sftp mechanisms.

Bulk statistics are stored on the receivers in files. The format of the bulk statistic data files can be configured by the user. Users can specify the format of the file name, file headers, and/or footers to include information such as the date, system host name, system uptime, the IP address of the system generating the statistics (available for headers and footers only), and/or the time that the file was generated.

When the Web Element Manager is used as the receiver, it is capable of further processing the statistics data through XML parsing, archiving, and graphing.

The Bulk Statistics Server component of the Web Element Manager parses collected statistics and stores the information in the PostgreSQL database. If XML file generation and transfer is required, this element generates the XML output and can send it to a northbound NMS or an alternate bulk statistics server for further processing.

Additionally, if archiving of the collected statistics is desired, the Bulk Statistics Server writes the files to an alternative directory on the server. A specific directory can be configured by the administrative user or the default directory can be used. Regardless, the directory can be on a local file system or on an NFS-mounted file system on the Web Element Manager server.

**Important:** For more information on bulk statistics, see the *System Administration Guide*.

## IKEv2 RFC 5996 Support

StarOS IKEv2 stack currently complies to RFC 4306. In Release 15.0, StarOS IKEv2 is enhanced to comply with newer version of IKEV2 RFC 5996. As part of new version support below features are introduced:
- **New notification payloads:** RFC 5996 introduces two new notification payloads TEMPORARY.FAILURE and CHILD_SA.NOT.FOUND using which certain conditions of the sender can be notified to the receiver.

- **Exchange collisions:** ePDG supports collision handling mechanism as defined in RFC 5996, it makes use of the new notify payloads in RFC5996 to do the same. Collision handling can be enabled using CLI, by default. Collision handling is supported as specified in RFC 4306/4718.

- **Integrity with combined mode ciphers:** StarOS IPSec is enhanced to graciously handle SA payloads containing combined mode cipher. In case an SA payload contains matching payload along with combined mode cipher, the one with combined mode cipher is ignored. Otherwise no proposal chosen is sent.

- **Negotiation parameters in CHILDSA REKEY:** Negotiation parameters in CHILDSA REKEY: According to RFC 5996 on rekeying of a CHILD SA, the traffic selectors and algorithms match the ones negotiated during the setting up of child SA. StarOS IKEv2 is enhanced to not send any new parameters in CREATE_CHILD_SA for a childsa being rekeyed. However StarOS IKEv2 does not enforce any restrictions on the peer for the same; this is done to minimize impact on IOT’s with existing peer vendor products, which may not be compliant to RFC 5996.

- **NAT traversal:** The Crypto engine accepts inbound udp-encapsulated IPSec ESP packets even if IKEv2 did not detect NATT. Inbound packets with udp_encap are accepted for processing.

- **Certificates:** RFC 5996 mandates configurability for sending and receiving HTTP method for hash-and-URL lookup with CERT/CERTREQ payloads. If configured and if peer requests for CERT using encoding type as “Hash and URL of X.509 certificate” and send HTTP_CERT_LOOKUP_SUPPORTED using notify payload in the first IKE_AUTH, ASR shall send the URL in the CERT payload instead of sending the entire certificate in the payload. If not configured and CERTREQ is received with encoding type as “hash and URL for X.509 certificate”, ASR should respond with entire certificate even if peer had sent HTTP_CERT_LOOKUP_SUPPORTED.

**IPv6 Support on IPSec SWU Interface**

When a UE attaches to a WiFi Access Point, the WiFi Access Point does assigns the UE an IP Address. Prior to this feature development the IP address assigned was always an IPv4 address. With this feature now the UE shall be provided an IPv4 or IPv6 address by the WiFi Access Point for initiating the IPsec connection to the ePDG over IPv4/IPv6 transport accordingly. For IPv6 transport the IPv6 UDP checksum is mandatory and is supported for IKEv2 establishment.

The ePDG now supports incoming IKEv2 requests from UE over an IPv6 transport as well. One epdg-service can now bound to one IPv4 and IPv6 address which acts as IPsec tunnel endpoint addresses. ePDG continues to support the inner IPv4, IPv6 and IPv4v6 traffic in both IPv4 and IPv6 outer IP SWu transport.

IPv6 NAT support is not standardized and there is no requirement to support the IPv6 NAT. If at all NAT related parameters are present in the crypto template during configuration, it should not have any impact on the tunnel setup and the data flow.

**Narrowing Traffic Selectors**

During traffic selector negotiation, ePDG by default responds with wildcard IP address, even if the UE is requesting specific range in the Tsr. The ePDG should allow to use specific sets of TSs to send traffic to specific sets of address ranges for specific client policies. The ePDG also should respect the range requested by UE and it should (according to the IKEv2 spec) be able to narrow down the UE’s request.

IKE Responder performs narrowing As per RFC5996 as shown below:
1. If the responder's policy does not allow it to accept any part of the proposed Traffic Selectors, it responds with a
   TS_UNACCEPTABLE Notify message.
2. If the responder's policy allows the entire set of traffic covered by TSi and TSr, no narrowing is necessary, and
   the responder can return the same TSi and TSr values.
3. If the responder's policy allows it to accept the first selector of TSi and TSr, then the responder MUST narrow
   the Traffic Selectors to a subset that includes the initiator's first choices.
4. If the responder's policy does not allow it to accept the first selector of TSi and TSr, the responder narrows to an
   acceptable subset of TSi and TSr.

All these 4 cases will be supported with the exception that at any point of time maximum of four traffic selector per
protocol (combination of IPv4 and/or IPv6) will be supported in a single CHILD SA.

When narrowing is done, if there are several subsets are acceptable, GW will respond back with first 4 acceptable
subsets and it will not support ADDITIONAL_TS_POSSIBLE notification.

**Static IP Address Allocation Support**

ePDG supports the static UE IP address communicated by AAA to ePDG over SWm interface (as Served-Party-IP-
Address AVP in DEA) and ePDG communicates the same to PGW over S2b interface (as PAA IE of create session
request GTP message and Home Network Prefix/IPv4 Home Address in PBU for PMIPv6 case).

This feature is applicable for both GTPv2 and PMIPv6 based implementation.

It shall be AAA server functionality to provide the static PGW IP address, when the UE IP address is provided statically
so that same PGW is selected which have the static IP pool corresponding to UE address. ePDG will continue with call
establishment and will not be validating the AAA provided PGW allocation type. It is the discretion of PGW to
accept/reject call in case the requested static IP address is not available at the PGW.

During handoff calls the priority should be given to UE provided IP address over the ones statically provided by AAA
server as the subscribed QoS profile at AAA may not be updated. When UE is offloaded from LTE the IP address
provided in LTE to UE should be given priority in WIFI over the AAA provided values. WIFI to WIFI handoff is not a
requirement so inter ePDG service handoff is not a valid use-case.

All the three PDN Types UE static IP address are supported including the IPv4, IPv6 and IPv4v6.

**Table 9. ePDG Static IP Address support failure matrix**

<table>
<thead>
<tr>
<th>S.N</th>
<th>UE requested PDN Type</th>
<th>AAA provided PDN type</th>
<th>AAA provided Static IP address type</th>
<th>ePDG Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>v4</td>
<td>v4</td>
<td>v4</td>
<td>Call established for v4 PDN type using the AAA provided static IP address.</td>
</tr>
<tr>
<td>2</td>
<td>v4</td>
<td>v4</td>
<td>v6</td>
<td>Call established for v4 PDN type but ignoring the AAA provided IP address.</td>
</tr>
<tr>
<td>3</td>
<td>v4</td>
<td>v4</td>
<td>v4v6</td>
<td>Call established for v4 PDN type and using v4 address provided by AAA.</td>
</tr>
<tr>
<td>4</td>
<td>v4v6</td>
<td>v4</td>
<td></td>
<td>Call established for v4 PDN type and using v4 address provided by AAA.</td>
</tr>
<tr>
<td>5</td>
<td>v4</td>
<td>v4v6</td>
<td>v4v6</td>
<td>Call established for v4 PDN type and using v4 address provided by AAA.</td>
</tr>
<tr>
<td>S.N</td>
<td>UE requested PDN Type</td>
<td>AAA provided PDN type</td>
<td>AAA provided Static IP address type</td>
<td>ePDG Action</td>
</tr>
<tr>
<td>-----</td>
<td>----------------------</td>
<td>-----------------------</td>
<td>------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>6</td>
<td>v4</td>
<td>v4v6</td>
<td>v6</td>
<td>Call established for v4 PDN type but ignoring the AAA provided IP address.</td>
</tr>
<tr>
<td>7</td>
<td>v4</td>
<td>v6</td>
<td>v6</td>
<td>Call released due to invalid-pdn-type reason.</td>
</tr>
<tr>
<td>8</td>
<td>v4</td>
<td>v6</td>
<td>v4v6</td>
<td>Call released due to invalid-pdn-type reason.</td>
</tr>
<tr>
<td>9</td>
<td>v4</td>
<td>v6</td>
<td>v4</td>
<td>Call released due to invalid-pdn-type reason.</td>
</tr>
<tr>
<td>10</td>
<td>v6</td>
<td>v4</td>
<td>v4v6</td>
<td>Call released due to invalid-pdn-type reason.</td>
</tr>
<tr>
<td>11</td>
<td>v6</td>
<td>v4</td>
<td>v4</td>
<td>Call released due to invalid-pdn-type reason.</td>
</tr>
<tr>
<td>12</td>
<td>v6</td>
<td>v4</td>
<td>v6</td>
<td>Call released due to invalid-pdn-type reason.</td>
</tr>
<tr>
<td>13</td>
<td>v6</td>
<td>v6</td>
<td>v4</td>
<td>Call established but ignoring the AAA provided IP address.</td>
</tr>
<tr>
<td>14</td>
<td>v6</td>
<td>v6</td>
<td>v4v6</td>
<td>Call established for v6 PDN type and using v6 address provided by AAA and v4 address is ignored.</td>
</tr>
<tr>
<td>15</td>
<td>v6</td>
<td>v6</td>
<td>v6</td>
<td>Call established for v6 PDN type and using v6 address provided by AAA.</td>
</tr>
<tr>
<td>16</td>
<td>v6</td>
<td>v4v6</td>
<td>v6</td>
<td>Call established for v6 pdn and using v6 address provided by AAA.</td>
</tr>
<tr>
<td>17</td>
<td>v6</td>
<td>v4v6</td>
<td>v4v6</td>
<td>Call established for v6 PDN and using v6 address provided by AAA and ignoring the v4 address.</td>
</tr>
<tr>
<td>18</td>
<td>v6</td>
<td>v4v6</td>
<td>v4</td>
<td>Call established but ignoring the AAA provided IP address.</td>
</tr>
<tr>
<td>19</td>
<td>v4v6</td>
<td>v4</td>
<td>v6</td>
<td>Call established using PDN type v4 and the static address provided by AAA is ignored.</td>
</tr>
<tr>
<td>20</td>
<td>v4v6</td>
<td>v4</td>
<td>v4</td>
<td>Call established using PDN type v4 and the static address provided by AAA is used.</td>
</tr>
<tr>
<td>21</td>
<td>v4v6</td>
<td>v4</td>
<td>v4v6</td>
<td>Call established using PDN type v4 and the static address provided by AAA is ignored.</td>
</tr>
<tr>
<td>22</td>
<td>v4v6</td>
<td>v6</td>
<td>v4</td>
<td>Call established using PDN type v6 and the static address provided by AAA is ignored.</td>
</tr>
<tr>
<td>23</td>
<td>v4v6</td>
<td>v6</td>
<td>v6</td>
<td>Call established using PDN type v6 and the static address provided by AAA is used.</td>
</tr>
<tr>
<td>24</td>
<td>v4v6</td>
<td>v6</td>
<td>v4v6</td>
<td>Call established using PDN type v6 and the static address v6 provided by AAA is used.</td>
</tr>
<tr>
<td>25</td>
<td>v4v6</td>
<td>v4v6</td>
<td>v4</td>
<td>Call established using PDN type v4v6 and static IP address provided by AAA is used.</td>
</tr>
<tr>
<td>26</td>
<td>v4v6</td>
<td>v4v6</td>
<td>v6</td>
<td>Call established using PDN type v4v6 and static v6 IP address provided by AAA is communicated to PGW over S2b.</td>
</tr>
<tr>
<td>27</td>
<td>v4v6</td>
<td>v4v6</td>
<td>v4v6</td>
<td>Call established using PDN type v4v6 and static IP address v4v6 both are communicated to PGW over S2b.</td>
</tr>
</tbody>
</table>
In case of mismatch in the PDN type between UE requested and the one provided by AAA server the call shall be released by ePDG with “invalid-pdn-type” as the disconnect reason.

**ePDG and PGW Support on the Same Chassis (with GTPv2)**

ePDG and PGW services does work together in combo mode (both enabled on the same chassis) with common component resources like IPsec being utilized in best effort manner. Session recovery including card migration is supported for the combo mode

**ICSR-VoLTE Support**

The ePDG does supports VoLTE call marking when the dedicated bearer corresponding to the QCI configured as VoLTE is created. The VoLTE call does have special handling of allowing data during the ICSR pending standby state and during the ICSR audit phase (at new active) which helps in reducing the data outage for the VoLTE calls during planned ICSR switchover.

Currently, when sessions are created on the ePDG, there is period of 60 seconds (configurable, explained below) lag before the sessions are check-pointed to the standby chassis. If chassis failure occurs during this period, the sessions that were not check-pointed are lost. Also, in some ICSR switchovers, a large number sessions that were not check-pointed need to be flushed resulting in additional delay in the switchover. This causes significant issues for VoLTE service. This is critical for IMS sessions. If an IMS session is not synchronized with the standby chassis and an ICSR switchover event occurs, the newly active chassis does not have any information of this session and the ePDG is out of sync with other network elements. This situation cannot be corrected until the UE registers again (max 2 hours) and VoLTE calls cannot be delivered to the UE. Therefore, it is critical to minimize the interval in which the session is not synchronized with the peer.

In maintenance mode it's required that ePDG should automatically delete the VoLTE calls when the VoLTE bearer gets teared down or subscriber becomes non-VoLTE after deletion of all VoLTE bearers.

In earlier release, "clear subs all non-volte" was implemented to clear non volte calls. Now "clear subs all non-volte auto-del" shall be implemented to delete non volte calls and mark the volte calls for auto deletion when the volte bearer is torn down. This helps in avoiding manual intervention from admin to cleanup calls again when volte bearer is torn down and the call becomes non-VoLTE. Once the call is marked for auto-deletion it cannot be reverted.

**Non VoLTE sessions data outage reduction**

ePDG does allows the data for non-VoLTE calls during ICSR switchover to reduce the data-outage for non-VoLTE calls and is configuration controlled to either allow data traffic for both VoLTE and non-VoLTE calls or only VoLTE calls.

**Local PGW Resolution Support**

In the current implementation of PGW selection, ePDG uses PGW address provided by AAA or uses DNS resolution. With local PGW resolution support, PGW address can be configured locally. If the above two methods (static and dynamic) PGW selection fails, or if PGW address were available but not reachable, then only locally configured addresses are referred and used. Also, if there is no PGW address received from AAA or, if no DNS setup is present, then also locally configured PGW addresses are referred. This way the existing functionality of PGW selection is retained, and added an additional backup-mode with local PGW address configuration resolution.

A new CLI is introduced in *ePDG Service Config* mode where epdg-service is associated with “subscriber-map”, which is also an indication that “Local PGW Resolution Support” is enabled for epdg-service. The local PGW resolution will
take into effect only if the CLI is configured and none of the existing method of PGW resolution method results in session creation.

Below are the Local PGW Resolution Support scenarios:

- PGW address received from AAA, but unreachable
- PGW addresses received by DNS resolution, but all are unreachable
- DNS server is not reachable, or rejects the DNS query
- None of the PGW selection mechanisms (Static/Dynamic) are present, i.e. neither DNS resolution is configured, nor AAA sends any PGW address

In all of the above scenarios, if local PGW address is configured and ePDG-Service is associated with Subscriber-Map, then PGW address is selected based on weight. In this algorithm the sessions are created approximately in the same ratio of the weights configured with the PGW addresses. For example if the weights are 10, 20 and 30, then 1000 sessions will be distributed in ration 1:2:3 respectively. (same algorithm used as DNS resolution based PGW selection mechanism.)

Only first PGW is selected based on weight based selection algorithm and if the call does not gets established with this selected PGW, rest of the addresses are selected on Round Robin method starting from next available PGW configured rounding upto PGW address configured just before the PGW address selected based on weight. This way none of the addresses are repeated. For example if ten PGW address are configured, based on weight 7th one is selected as first address, and if it is unreachable then address at 8th index is selected, then 9th, 10th, 1st, 2nd and so on until address present at 6th index.

In a case where PGW resolution is enabled and the existing DNS/AAA server PGW resolution mechanism failed and there is no disconnect reason already set from previous mechanism, further the local PGW resolution failed due to configuration error then new disconnect reason shall be set “ePDG-local-pgw-resolution-failed” for identifying the case.

Also in the case of HO, even if the local PGW resolution is enabled and there is no or unreachable PGW address provided by AAA server, or PGW FQDN provided results in no or unreachable PGW address, then ePDG will not use local PGW resolution mechanism for establishing the call.

**Local configuration as preferred PGW selection mechanism**

The ePDG is further enhanced to support local configuration based PGW selection as the preferred method for PGW node selection.

The ePDG service should be configured indicating preferred method of PGW selection, whether local configuration or DNS/AAA server based PGW selection. Local Configuration based PGW selection as fallback mechanism is default configuration behavior.

This preferred PGW selection mechanism feature provides more control and flexibility to customer for routing/load balancing the sessions on desired PGW.

The feature shall be applicable only for initial attach and for Hand-Off calls ePDG shall use the PGW address provided by AAA server even if the feature is enabled as the PGW selected by local configuration may be different from one have the session on LTE.

**Non UICC Device Support Using Certificate Based Authentication**

ePDG is enhanced to support the non UICC devices connectivity to EPC via ePDG using certificate based UE authentication following authorization by AAA server.

ePDG already supports UICC devices connectivity using EAP-AKA based device authentication. However as non UICC devices cannot do EAP-AKA based authentication, alternate method of using certificates is used.
ePDG supports the X.509 certificate based authentication and also communicates with OCSP (Online Certificate Status Protocol) server for completing the authentication. Once the authentication is done ePDG communicates with AAA server for ensuring the authorization of the device.

As non UICC devices do not have IMSI, customized vIMSI in format similar to UICC IMSI uniquely identifying the non UICC device needs to be shared by the device. The device IMSI is shared as part of peer (device) certificate to ePDG. ePDG extracts serial number, issuing authority and OCSP responder address details from the certificate and communicates with OCSP responder. In case the OCSP responder detail is absent in the certificate the ePDG configuration is used for extracting the same. The OCSP client (ePDG) to the OCSP responder interaction will be over HTTP. A TCP socket connection will be established to the OCSP responder. OCSP responder communicates with the associated CA (certification authority) and gets the certificate revocation status which can be "good" or "revoked" or "unknown". The ePDG behavior in case of "unknown" is similar to "revoked". When the OCSP response reaches ePDG, it validates if the response is received from genuine entity and post validation checks the certificate status. If the certificate status is good then proceeds with device authorization.

ePDG expects the SUBJECT/CN field of UE certificate to contain the IMSI or NAI and detects that its NAI with presence of '@' else its IMSI. This extracted CN fields is accordingly verified with the IDi payload received from UE in IKE_AUTH_REQ message. The certificate identity is more reliable and also the IKE_AUTH_REQ identity does have significance is AUTH payload verification hence this functionality of comparison is in place. ePDG sends the NAI identity as received in the IKE_AUTH_REQ message to the AAA server and once AAA server sends back the authorization success then ePDG does PGW selection and communicates with PGW over S2b interface to establish the call.

IPsec subsystem does comply with RFC 2560 and uses open SSL 0.9.7 for certificate based authentication, therefore ePDG inherently complies with same.

ePDG supports both UICC and non-UICC devices simultaneously for same ePDG service. ePDG service does have single crypto template association with the service IP address and hence IPsec subsystem is enhanced for supporting the multiple authentication methods per crypto template. ePDG identifies whether certificate based authentication needs to be used or not by the presence of AUTH payload. If the AUTH parameter is absent in initial IKE_AUTH_REQ message it indicates that EAP-AKA based authentication is to be used. If the AUTH payload is present and the CERT payload is also present it indicates certificate based mechanism is to be used.

OCSP communication is optional and if not configured then ePDG validates based on the configured CA certificates.
Figure 7. NON UICC device Call flow


2. **ePDG → UE: IKEv2 SA_INIT RSP** The ePDG responds with an IKE_SA_INIT Response (SA, KE, Nr payloads, NAT-DETECTION Notify, [CERTREQ]).

3. **UE → ePDG: IKEv2 AUTH REQ** UE sends IKE_AUTH_REQ (IDi, AUTH, CERT, [CERTREQ], IDR, SA, CFG_REQUEST (INTERNAL_IP6_ADDRESS, [INTERNAL_IP6_DNS], [INTERNAL_IP6_PSCSF], TSi, TSr)). The UE does include AUTH and CERT payload to indicate that it will use the certificates (X.509) for authenticating itself. Presence of AUTH payload indicates EAP-AKA is not used. IDi contains the NAI and IDR does contain the APN name. Root NAI is of format X<IMSI>@nai.epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org so IMSI (virtual IMSI used for non UICC device IMSI) is required which should be of decimal digit UICC IMSI format. One proposed approach is to use <device prefix><MSISDN> where MSISDN is common for the associated non UICC and UICC devices but its operator decision and ePDG shall be able to handle it until its unique per non UICC device and is in UICC IMSI format. The certificate SUBJECT/CN field shall be containing the IMSI or NAI as it’s identifier. ePDG uses received public key as part of certificates for authenticating the UE. OCSP shall be used for checking the revocation status during the certificates based device authentication. OCSP communication is optional means if the OCSP responder is absent in operator infrastructure then the ePDG shall be authenticating the device using the configured Root CA certificate. **Note**: The device can share the certificates (X.509) or can communicate the URL to ePDG for downloading the device certificates. Both the mechanism are supported on ePDG.

4. **ePDG → OCSP responder : OCSP request** ePDG sends the OCSP request containing the certificate identifier.

5. **OCSP responder → ePDG :OCSP Response** OCSP responder checks and returns back the revocation status of the certificate. At this stage ePDG completes the authentication of the device.

6. **ePDG → AAA server :AAR** The ePDG sends the AA-Request (Session-Id, Auth-Application-Id, Origin-Host, Origin-Realm, Destination-Host, Destination-Realm, Auth-Request-Type (AUTHORIZE_AUTHENTICATE), User-Name (NAI)) message to the 3GPP AAA server. ePDG communicates the NAI for AAA to check UE identity and authorize the same.

7. **AAA server → HSS :SAR** The 3GPP AAA updates the HSS with the 3GPP AAA Server Address information for the user. The AAA sends Server-Assignment-Request (Session-Id, Auth-Session-State (NO_STATE_MAINTAINED), Origin-Host, Origin-Realm, Destination-Host, Destination-Realm, User-Name (IMSI-NAI), Server-Assignment-Type (REGISTRATION)). **Note**: As this call flow is not defined in 3GPP yet so the proposed message between AAA to HSS is to be decided by AAA and HSS vendors however based on existing SWx interface messages have proposed the usage of SAR.

---

**Error:** The diagram does not match the text content. The text appears to describe a different sequence of steps than what is depicted in the diagram. The text describes steps 1 through 7, while the diagram only shows steps 1 and 2. The diagram is likely intended to show the flow of events as described in the text, but it is not accurately reflecting the steps outlined in the text.
8. HSS → AAA server : SAA The HSS sends Server-Assignment-Answer (Session-Id, Result-Code, Experimental-Result (Vendor-Id, Experimental-Result-Code), Non-3GPP-User-Data (Subscription-ID (END_USER_E164, MSISDN), Non-3GPP-IP-Access (NON_3GPP_SUBSCRIPTION_ALLOWED), Non-3GPP-IP-Access-APN (Non_3GPP_APNS_ENABLE), APN-Configuration, ANID (WLAN), APN-OI-Replacement, APN-Configuration))

9. AAA server → ePDG: AA-Answer The 3GPP AAA Server responds with AAA (Session-Id, Auth-Application-Id, Auth-Request-Type, Origin-Host, Origin-Realm, Result-Code, User-Name, APN-Configuration, 3GPP-Charging-Characteristics, Subscription-ID)

10. ePDG → DNS server: DNS(NAPTR/AAAA) query ePDG sends DNS query to DNS server with APN/PGW FQDN for PGW resolution.

11. DNS server → ePDG: DNS query response DNS server returns the PGW address to ePDG as part of DNS AAAA/A response.

12. ePDG → PGW: S2b Create Session Req ePDG selects PGW based on DNS mechanism using APN/PGW FQDN. The ePDG sends Create Session Request (IMSI, [MSISDN], Serving Network, RAT Type (WLAN), Indication Flags, Sender F-TEID for C-plane, APN, Selection Mode, PAA, APN-AMBR, [APCO], Bearer Contexts(), [Recovery], [Private IE (P-CSCF)]). Selection Mode shall be set to "MS or network provided APN subscribed verified". Private IE is populated if the UE request P-CSCF addresses. The PGW performs the necessary interactions with 3GPP-AAA, PCRF and OCS/OFCS.

13. PGW → ePDG: Create Session Resp The PGW allocates the requested IP address session and responds back to the ePDG with a Create Session Response (Cause, PGW S2b F-TEID, PAA, [APN-AMBR],[APCO],Bearer Contexts Created (EPS Bearer ID, Cause, [TFT], S2b-U PGW F-TEID, Bearer Level QoS), [Recovery], [Private IE (P-CSCF)]) message.

14. ePDG → UE: IKEv2 AUTH_RESP ePDG sends IKE_AUTH_RESP (AUTH, IDr, [CERT (X509 CERTIFICATE SIGNATURE)], CP, SA, CFG_REPLY ([INTERNAL_IP4_ADDRESS], [INTERNAL_IP4_NETMASK], [INTERNAL_IP4_DNS], INTERNAL_IP6_ADDRESS, INTERNAL_IP6_SUBNET, INTERNAL_IP6_DNS, [P-CSCF]) TSi, TSr)

15. ePDG → UE: IPv6 RA The assumption is that the IP stack needs the RA to initialize the address.

EAP-MSCHAPv2/EAP-TLS/EAP-TTLS Based Support For NON UICC Devices

Currently 3GPP standard provides a mechanism for the UICC (SIM based) devices connectivity to the EPC via non-3GPP access enabling them for voice and video services over WiFi. However lot of non UICC devices such as iPads, Tablets, Laptops do not have defined 3GPP standard mechanism for connecting over WLAN to EPC via ePDG. These devices can use the same LTE subscription as for the UICC device do not have potential to utlize CSPs and monetize voice and video offering by extending the same to non UICC devices.

EAP-AKA is the mechanism defined in 3GPP standards for authenticating and authorizing the mobile devices using AAA server. The non UICC devices cannot support EAP-AKA.

For non UICC devices as IMSI is not present the IMSI mentioned in below flows is vIMSI which can be alphanumeric type (limit to 24 chars) or decimal digit IMSI and in such case when alphanumeric vIMSI is used its expected that AAA server shall be providing decimal digit IMSI to ePDG for S2b interface as part of mobile-node-identifier AVP.

Below is the list of different authentication mechanisms which can be used with ePDG acting as EAP pass-through mode for the non UICC device support:

- **EAP-MSCHAPv2**
  - Single phase
  - Use MSCHAPv2 inside EAP
  - Challenge/Response based mechanism
- EAP-TTLS (using MS-CHAPv2)
  - EAP method encapsulating TLS session
  - Two phases
    - Handshake phase (server authentication and key generation)
    - Data Phase (client authentication)
  - Handshake phase provides secure channel for data phase
  - Use MSCHAPv2 for authenticating client/device
  - Reference - RFC 5281

- EAP-TLS
  - Single phase
  - EAP method encapsulating TLS session
  - Use certificates between UE and AAA server for mutual authentication
  - Reference - RFC 5216

**EAP-MSCHAPv2 authentication mechanism call flow**

In this authentication mechanism the ePDG shall be acting in EAP pass-through mode and the AAA server shall be authenticating the device using EAP-MSCHAPv2. The authentication mechanism does have advantage of less lengthy call flow and is standard way. Additionally the operator does not require having certificate based infrastructure. The disadvantage is that MSK is 64 bytes but with 32 byte key and remaining 32 bytes as zeros as opposed to EAP-AKA where we have 64 byte non zero MSK. So effectively weaker authentication mechanism key. The Following diagram shows the call flow for the EAP-MSCHAPv2 based authentication:
1. UE → ePDG: IKEv2 SA_INIT UE (UICC based) sends IKE_SA_INIT Request (SA, KE, Ni, NAT-Detection Notify).
2. ePDG → UE: IKEv2 SA_INIT RSP The ePDG responds with an IKE_SA_INIT Response (SA, KE, Nr payloads, NAT-Detection Notify).
3. UE → ePDG: IKEv2 AUTH_REQ UE sends IKE_AUTH_REQ (Idi, [CERTREQ], IDR, SA, CP (CFQ_REQUEST (INTERNAL_IP6_ADDRESS, [INTERNAL_IP6_DNS], [INTERNAL_IP6_PCSF]), Tsi, Tsr)). The UE does not include AUTH payload to indicate that it will use the EAP-MSCHAPv2 method for authenticating itself to AAA. Idi contains the NAI in the form "A<IMSI>@nai.epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org". Per standards the prefix can be 0/1 indicating EAP-AKA/EAP-SIM now as we shall be indicating to AAA server that use different authentication method here EAP-MSCHAPv2 so can indicate using "A". ePDG shall be transparent to received prefix and shall send to AAA server so that operator is free to use any prefix except the defined ones.
4. ePDG → AAA server : DER The ePDG sends the DER (Session-Id, Auth-Application-Id, Origin-Host, Origin-Realm, Destination-Host, Destination-Realm, Auth-Request-Type (AUTHORIZE AUTHENTICATE), EAP-Payload, User-Name (NAI), RAT-Type (WLAN), MIPv6-Feature-Vector, Visited-Network-Identifier) message to the 3GPP AAA Server. The EAP-Payload shall contain the UE identity encoded by ePDG.

5. AAA server → ePDG: DEA The 3GPP AAA Server initiates the authentication challenge and responds with DEA (Session-Id, Base AVPs, Auth-Request-Type and EAP-Payload). The EAP-Payload shall contain the Challenge packet which is used to begin the EAP MS-CHAP-V2 protocol.

6. ePDG → UE: IKEv2 AUTH_RESP ePDG sends IKE_AUTH_RESP (IDr, [CERT (X509 CERTIFICATE SIGNATURE)], EAP Payload) The IDr is the identity of the ePDG and if the UE requests for certificates then CERT is included. The EAP message received from the 3GPP AAA Server (EAP-Request/Challenge) is included in order to start the EAP procedure over IKEv2.

7. UE → ePDG: IKEv2 AUTH_REQ The UE sends EAP message in IKE_AUTH Request (EAP) with user-name, MS-CHAP2- Response AVPs. The EAP message shall be of EAP-Type=EAP-MS-CHAP-V2(Response).

8. ePDG → AAA server : DER The ePDG sends the DER (Session-Id, Auth-Application-Id, Origin-Host, Origin-Realm, Destination-Host, Destination-Realm, Auth-Request-Type (AUTHORIZE AUTHENTICATE), EAP-Payload) message to the 3GPP AAA Server. The EAP-Payload shall contain the message as sent by UE.

9. AAA server → ePDG: DEA The 3GPP AAA Server on successful authentication responds with DEA (Session-Id, Base AVPs, Auth-Request-Type and EAP-Payload)

10. ePDG → UE: IKEv2 AUTH_RESP ePDG sends IKE_AUTH_RESP (EAP Payload) The EAP payload shall contain the EAP-MSCHAPv2 message as received from the AAA server.

11. UE → ePDG: IKEv2 AUTH_REQ The UE sends IKE_AUTH Request (EAP) with EAP-MSCHAPv2 "Success Response packet". UE successfully validates the EAP MS-CHAP-V2 Success Request packet sent by the AAA server, respond.

12. ePDG → AAA server : DER The ePDG sends the DER (Session-Id, Auth-Application-Id, Origin-Host, Origin-Realm, Destination-Host, Destination-Realm, Auth-Request-Type (AUTHORIZE AUTHENTICATE), EAP-Payload) message to the 3GPP AAA Server. The EAP-Payload shall contain the message as sent by UE.

13. AAA server → ePDG: DEA The 3GPP AAA Server sends an EAP success (Session-Id, Auth-Application-Id: 16777264, Result-Code, Origin-Host, Origin-Realm, Auth-Request-Type (AUTHORIZE AUTHENTICATE), EAP-Payload User-Name(0<~IMSI@<mnc<mnc val>_<mcc<mcc val>.pub.3gppnetwork.org), EAP-Master-Session-Key, APN-Configuration (Context-Identifier, PDN-Type: IPv4v6, Service-Selection (apn name), MIPv6-Agent-Info), Auth-Session-State: STATE_MAINTAINED, Origin-State-Id). At this point mutual authentication is done and device is authorized by AAA server. The MSK can be generated by AAA server using following logic however ePDG is transparent to MSK generation logic and till the devices and AAA server are in sync any other logic of MSK generation should also work. MSK = MasterReceiveKey + MasterSendKey + 32 bytes zeroes (padding) Note - Extensible Authentication Protocol Method for Microsoft CHAP derives two 16-byte keys, MasterSendKey and MasterReceiveKey (as specified in [RFC3079], section 3.3).

14. ePDG → UE: IKEv2 AUTH_RESP ePDG sends IKE_AUTH_RESP (EAP Payload) The EAP payload shall contain the EAP-MSCHAPv2 message as received from the AAA server.

15. UE → ePDG: IKEv2 AUTH_REQ UE sends IKE_AUTH request (AUTH) The UE takes its own copy of the MSK as input to generate the AUTH parameter to authenticate the first IKE_SA_INIT message.

16. ePDG → PGW: S2b Create Session Req ePDG sends Create Session Request (IMSI, [MSISDN], Serving Network, RAT Type (WLAN), Indication Flags, Sender F-TEID for C-plane, APN, Selection Mode, PAA, APN-AMBR, [APCO], Bearer Contexts(), [Recovery], [Private IE (P-CSFC)])). Selection Mode shall be set to "MS or network provided APN subscribed verified". The PGW performs the necessary interactions with 3GPP-AAA, PCRF and OCS/OFCS. ePDG shall set the HO in Indication flags IE and also the preserved IP address as received from UE in PAA IE.

17. PGW → ePDG: Create Session Resp The PGW allocates the requested IP address session and responses back to the ePDG with a Create Session Response (Cause, PGW S2b F-TEID, PAA, [APN-AMBR], APCO, Bearer Contexts Created (EPS Bearer ID, Cause, [TFT], S2b-U PGW F-TEID, Bearer Level QoS), [Recovery], [Private IE (P-CSFC)]) message.

18. ePDG → UE: IKEv2 AUTH_RESP ePDG sends IKE_AUTH_RESP (AUTH, CP, SA, CFG_REPLY ([INTERNAL_IP4_ADDRESS], [INTERNAL_IP4_NETMASK], [INTERNAL_IP4_DNS], INTERNAL_IP6_ADDRESS, INTERNAL_IP6_SUBNET, INTERNAL_IP6_DNS, [P-CSFC]) TSi, TSr) At
this stage the ePDG has completed the ipsec SA and tunnel setup and also GTP-U tunnel setup thus completing the data path. The IP address provided by PGW is communicated to UE.

19. ePDG → UE: IPv6 RA The assumption is that the IP stack needs the RA to initialize the address.

**EAP-TLS authentication mechanism Call Flow**

In this mechanism it’s assumed that the authenticator entity shall be AAA server supporting the certificate based authentication. The ePDG shall be acting in EAP pass-through mode thus communicating the EAP-TLS negotiation between device and AAA server. The AAA server once completing the authentication mechanism shall be sharing the MSK to ePDG for generating the AUTH parameters and completing the IKEv2 authentication. Following diagram shows the call flow for the EAP-TLS based authentication:

Figure 9. IPsec Based EAP-TLS Flow
1. UE ePDG: IKEv2 SA_INIT (UICC based) sends IKE_SA_INIT Request (SA, KE, Ni, NAT-Detection Notify).
2. ePDG → UE: IKEv2 SA_INIT_RSP The ePDG responds with an IKE_SA_INIT Response (SA, KE, Nr payloads, NAT-Detection Notify).
3. UE → ePDG: IKEv2 AUTH_REQ UE sends IKE_AUTH_REQ (Idi, [CERTREQ], IDR, SA, CP (CFQ_REQUEST (INTERNAL_IP6_ADDRESS, [INTERNAL_IP6_DNS], [INTERNAL_IP6_PSCCF]), TSi, TSr)). The UE does not include AUTH payload to indicate that it will use the EAP-TLS method for authenticating itself to AAA. Idi contains the NAI in the form "A<@<IMSI>@nai.epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org". Per standards the prefix can be 0/1 indicating EAP-AKA/EAP-SIM now as we shall be indicating to AAA server that use different authentication method here EAP-TLS so can indicate using "A". ePDG shall be transparent to received prefix and shall send to AAA server so that operator is free to use any prefix except the defined ones.
4. ePDG → AAA server :DER The ePDG sends the DER (Session-Id, Auth-Application-Id, Origin-Host, Origin-Realm, Destination-Host, Destination-Realm, Auth-Request-Type(AUTHORIZE_AUTHENTICATE), EAP-Payload, User-Name (NAI), RAT-Type(WLAN), MIP6-Feature-Vector, Visited-Network-Identifier) message to the 3GPP AAA Server. The EAP-Payload shall contain the UE identity encoded by ePDG.
5. AAA server → ePDG: DEA The 3GPP AAA Server initiates the authentication challenge and responds with DEA (Session-Id, Base AVPs, Auth-Request-Type and EAP-Payload). The EAP-Payload shall contain the EAP-TLS/Start, the Start 'S' bit is set with no data.
6. ePDG → UE: IKEv2 AUTH_RESP ePDG sends IKE_AUTH_RESP (Idr, [CERT (X509 CERTIFICATE SIGNATURE)], EAP Payload) The IDr is the identity of the ePDG and if the UE requests for certificates then CERT is included. The EAP message received from the 3GPP AAA Server (EAP-Request/Start) is included in order to start the EAP procedure over IKEv2.
7. UE → ePDG: IKEv2 AUTH_REQ UE sends IKE_AUTH_REQ (EAP payload) containing the TLS client hello handshake message.
8. ePDG → AAA server :DER The ePDG sends the DER (Session-Id, Auth-Application-Id, Origin-Host, Origin-Realm, Destination-Host, Destination-Realm, Auth-Request-Type(AUTHORIZE_AUTHENTICATE), EAP-Payload, User-Name (NAI), RAT-Type(WLAN), MIP6-Feature-Vector, Visited-Network-Identifier) message to the 3GPP AAA Server. The EAP-Payload shall contain the TLS client hello handshake message.
9. AAA server → ePDG: DEA The 3GPP AAA Server initiates the authentication challenge and responds with DEA (Session-Id, Base AVPs, Auth-Request-Type and EAP-Payload). The AAA server will then respond with an EAP-Request packet with EAP-Type=EAP-TLS. The data field of this packet will encapsulate one or more TLS records. These will contain a TLS server_hello handshake message, possibly followed by TLS certificate, server_key_exchange, certificate_request, server_hello_done and/or finished handshake messages, and/or a TLS change_cipher_spec message.
10. ePDG → UE: IKEv2 AUTH_RESP ePDG sends IKE_AUTH_RESP (EAP Payload) The EAP payload shall contain the TLS message as received from the AAA server.
11. UE → ePDG: IKEv2 AUTH_REQ The UE sends EAP message in IKE_AUTH Request (EAP). The data field of this packet MUST encapsulate one or more TLS records containing a TLS client_key_exchange, change_cipher_spec, and finished messages.
12. ePDG → AAA server :DER The ePDG sends the DER (Session-Id, Auth-Application-Id, Origin-Host, Origin-Realm, Destination-Host, Destination-Realm, Auth-Request-Type(AUTHORIZE_AUTHENTICATE), EAP-Payload) message to the 3GPP AAA Server. The EAP-Payload shall contain the message as sent by UE.
13. AAA server → ePDG: DEA The 3GPP AAA Server on successful authentication responds with DEA (Session-Id, Base AVPs, Auth-Request-Type and EAP-Payload) where EAP-Request/Start is included in the EAP-Payload.
14. ePDG → UE: IKEv2 AUTH_RESP ePDG sends IKE_AUTH_RESP (EAP Payload) The EAP payload shall contain the TLS message as received from the AAA server.
15. UE → ePDG: IKEv2 AUTH REQ The UE sends EAP message in IKE_AUTH Request (EAP) with no data.
16. ePDG → AAA server :DER The ePDG sends the DER (Session-Id, Auth-Application-Id, Origin-Host, Origin-Realm, Destination-Host, Destination-Realm, Auth-Request-Type (AUTHORIZE_AUTHENTICATE), EAP-Payload) message to the 3GPP AAA Server. The EAP-Payload shall contain the message as sent by UE.
17. AAA server → ePDG: DEA The 3GPP AAA Server sends an EAP success (Session-Id, Auth-Application-Id: 16777264, Result-Code, Origin-Host, Origin-Realm, Auth-Request-Type(AUTHORIZE_AUTHENTICATE),
EAP-Payload User-Name(0<IMSI>@mnc<mnc val>.mcc<mcc val>.pub.3gppnetwork.org), EAP-Master-Session-Key, APN-Configuration (Context-Identifier, PDN-Type: IPv4v6, Service-Selection (apn name), MIP6-Agent-Info), Auth-Session-State:STATE_MAINTAINED, Origin-State-Id). At this point device is authenticated and authorized by AAA server.

18. ePDG → UE: IKEv2 AUTH_RESP ePDG sends IKE_AUTH_RESP (EAP Payload) The EAP payload shall contain the TLS message as received from the AAA server.

19. UE → ePDG: IKEv2 AUTH_REQ UE sends IKE_AUTH request (AUTH) The UE takes its own copy of the MSK as input to generate the AUTH parameter to authenticate the first IKE_SA_INIT message.

20. ePDG → PGW: S2b Create Session Req ePDG sends Create Session Request (IMSI, [MSISDN], Serving Network, RAT Type (WLAN), Indication Flags, Sender F-TEID for C-plane, APN, Selection Mode, PAA, APN-AMBR, [APCO], Bearer Contexts(), [Recovery], [Private IE (P-CSCF)]). Selection Mode shall be set to "MS or network provided APN subscribed verified". The PGW performs the necessary interactions with 3GPP-AAA, PCRF and OCS/OFCS. ePDG shall set the HO in Indication flags IE and also the preserved IP address as received from UE in PAA IE.

21. PGW → ePDG: Create Session Resp The PGW allocates the requested IP address session and responds back to the ePDG with a Create Session Response (Cause, PGW S2b F-TEID, PAA, [APN-AMBR],APCO, Bearer Contexts Created (EPS Bearer ID, Cause, [TFT], S2b-U PGW F-TEID, Bearer Level QoS), [Recovery], [Private IE (P-CSCF)]) message.

22. ePDG → UE: IKEv2 AUTH_RESP ePDG sends IKE_AUTH_RESP (AUTH, CP, SA, CFG_REPLY([INTERNAL_IP4_ADDRESS], [INTERNAL_IP4_NETMASK], [INTERNAL_IP4_DNS], INTERNAL_IP6_ADDRESS, INTERNAL_IP6_SUBNET, INTERNAL_IP6_DNS, [P-CSCF]) TSi, TSr) At this stage the ePDG has completed the ipsec SA and tunnel setup and also GTP-U tunnel setup thus completing the data path. The IP address provided by PGW is communicated to UE.

23. ePDG → UE: IPv6 RA The assumption is that the IP stack needs the RA to initialize the address.

EAP-TTLS authentication mechanism Call Flow

The EAP-TTLS based approach is useful when there is no certificate based infrastructure present for the operator to configure certificate for each device. Unlike EAP-TLS it enables the device authentication without certificates using customized AVPs. Here we have defined MSCHAPv2 based authentication mechanism. Here the AAA server needs to provide the key similar to MSK to ePDG for validating/generating the AUTH payload during IKEv2 xchg. Following diagram shows the call flow for the EAP-TTLS based authentication:
1. UE → ePDG: IKEv2 SA_INIT UE (UICC based) sends IKE_SA_INIT Request (SA, KE, Ni, NAT-DETECTION Notify).

2. ePDG → UE: IKEv2 SA_INIT RSP The ePDG responds with an IKE_SA_INIT Response (SA, KE, Nr payloads, NAT-Detection Notify).

3. UE → ePDG: IKEv2 AUTH_REQ UE sends IKE_AUTH_REQ (IDi, [CERTREQ], IDr, SA, CP (CFQ_REQUEST (INTERNAL_IP6_ADDRESS, [INTERNAL_IP6_DNS], [INTERNAL_IP6_PSCSF]), TSi, TSr)). The UE does not include AUTH payload to indicate that it will use the EAP-TTLS method for authenticating itself to AAA. IDi contains the NAI in the form "A<~IMSI>@nai.epc.mnc<MNC>.mcc<MCC>.3gppnetwork.org". Per standards the prefix can be 0/1 indicating EAP-AKA/EAP-SIM now as we shall be indicating to AAA server that use different authentication method here.
EAP-TTLS so can indicate using "A". ePDG shall be transparent to received prefix and shall send to AAA server so that operator is free to use any prefix except the defined ones.

4. ePDG → AAA server : DER The ePDG sends the DER (Session-Id, Auth-Application-Id, Origin-Host, Origin-Realm, Destination-Host, Destination-Realm, Auth-Request-Type(AUTHORIZE_AUTHENTICATE), EAP-Payload, User-Name (NAI), RAT-Type(WLAN), MIP6-Feature-Vector, Visited-Network-Identifier) message to the 3GPP AAA Server. The EAP-Payload shall contain the UE identity encoded by ePDG.

5. AAA server → ePDG: DEA The 3GPP AAA Server initiates the authentication challenge and responds with DEA (Session-Id, Base AVPs, Auth-Request-Type and EAP-Payload). The EAP-Payload shall contain the EAP-TTLS/Start, the Start 'S' bit is set with no data.

6. ePDG → UE: IKEv2 AUTH_RESP ePDG sends IKE_AUTH_RESP (IDr, [CERT (X509 CERTIFICATE SIGNATURE)], EAP Payload) The Idr is the identity of the ePDG and if the UE requests for certificates then CERT is included. The EAP message received from the 3GPP AAA Server (EAP-Request/Start) is included in order to start the EAP procedure over IKEv2.

7. UE → ePDG: IKEv2 AUTH_REQ UE sends IKE_AUTH_REQ (EAP payload) containing the TLS client hello handshake message.

8. ePDG → AAA server : DER The ePDG sends the DER (Session-Id, Auth-Application-Id, Origin-Host, Origin-Realm, Destination-Host, Destination-Realm, Auth-Request-Type(AUTHORIZE_AUTHENTICATE), EAP-Payload, User-Name (NAI), RAT-Type(WLAN), MIP6-Feature-Vector, Visited-Network-Identifier) message to the 3GPP AAA Server. The EAP-Payload shall contain the TLS client hello handshake message.

9. AAA server → ePDG: DEA The 3GPP AAA Server initiates the authentication challenge and responds with DEA (Session-Id, Base AVPs, Auth-Request-Type and EAP-Payload). The AAA server will then respond with an EAP-Request packet with EAP-Type=EAP-TTLS. The data field of this packet will encapsulate one or more TLS records. These will contain a TLS server_hello handshake message, possibly followed by TLS certificate, server_key_exchange, server_hello_done and/or finished handshake messages, and/or a TLS change_cipher_spec message.

10. ePDG → UE: IKEv2 AUTH_RESP ePDG sends IKE_AUTH_RESP (EAP Payload) The EAP payload shall contain the TLS message as received from the AAA server.

11. UE → ePDG: IKEv2 AUTH_REQ The UE sends EAP message in IKE_AUTH Request (EAP). The data field of this packet MUST encapsulate one or more TLS records containing a TLS client_key_exchange, change_cipher_spec, and finished messages.

12. ePDG → AAA server : DER The ePDG sends the DER (Session-Id, Auth-Application-Id, Origin-Host, Origin-Realm, Destination-Host, Destination-Realm, Auth-Request-Type(AUTHORIZE_AUTHENTICATE), EAP-Payload) message to the 3GPP AAA Server. The EAP-Payload shall contain the message as sent by UE.

13. AAA server → ePDG: DEA The 3GPP AAA Server on successful authentication responds with DEA (Session-Id, Base AVPs, Auth-Request-Type and EAP-Payload) where EAP-Payload does contain the TLS finished message.

14. ePDG → UE: IKEv2 AUTH_RESP ePDG sends IKE_AUTH_RESP (EAP Payload) The EAP payload shall contain the TLS message as received from the AAA server. This stage the first phase of TTLS is done completing the TLS handshake and AAA server is authenticated by device and keys are generated to secure subsequent message handling.

15. UE → ePDG: IKEv2 AUTH_REQ The UE sends EAP message in IKE_AUTH Request (EAP) with user-name, MS-CHAP2-Response, MS-CHAP Challenge AVPs.

16. ePDG → AAA server : DER The ePDG sends the DER (Session-Id, Auth-Application-Id, Origin-Host, Origin-Realm, Destination-Host, Destination-Realm, Auth-Request-Type(AUTHORIZE_AUTHENTICATE), EAP-Payload) message to the 3GPP AAA Server. The EAP-Payload shall contain the message as sent by UE.

17. AAA server → ePDG: DEA The 3GPP AAA Server on successful authentication responds with DEA (Session-Id, Base AVPs, Auth-Request-Type and EAP-Payload), Upon receipt of these AVPs from the UE, the AAA server MUST verify that the value of the MS-CHAP-Challenge AVP and the value of the Ident in the client's MS-CHAP2-Response AVP are equal to the values generated as challenge material. If either item does not match exactly, the AAA server MUST reject the UE. In success case, AAA shall encode the MS-CHAP2-Success attribute.

18. ePDG → UE: IKEv2 AUTH_RESP ePDG sends IKE_AUTH_RESP (EAP Payload) The EAP payload shall contain the EAP-TTLS message as received from the AAA server.
19. **UE → ePDG: IKEv2 AUTH_REQ** The UE sends IKE_AUTH Request (EAP) with no data. Upon receipt of the MS-CHAP2-Success AVP, the UE is able to authenticate the AAA. If the authentication succeeds, the UE sends an EAP-TTLS packet to the TTLS server containing no data (that is, with a zero-length Data field). Upon receipt of the empty EAP-TTLS packet from the client, the TTLS server considers the MS-CHAP-V2 authentication to have succeeded.

20. **ePDG → AAA server: DER** The ePDG sends the DER (Session-Id, Auth-Application-Id, Origin-Host, Origin-Realm, Destination-Host, Destination-Realm, Auth-Request-Type(AUTHORIZE_AUTHORIZE), EAP-Payload) message to the 3GPP AAA Server. The EAP-Payload shall contain the message as sent by UE.

21. **AAA server → ePDG: DEA** The 3GPP AAA Server sends an EAP success (Session-Id, Auth-Application-Id: 16777264, Result-Code, Origin-Host, Origin-Realm, Auth-Request-Type(AUTHORIZE_AUTHORIZE), EAP-Payload User-Name(0<IMSIP>@mnc<mnc val>.mcc<mcc val>.pub.3gppnetwork.org), EAP-Master-Session-Key, APN-Configuration (Context-Identifier, PDN-Type: IPv4v6, Service-Selection (apn name), MIP6-Agent-Info), Auth-Session-State: STATE_MAINTAINED, Origin-State-Id). At this point mutual authentication is done and device is authorized by AAA server.

22. **ePDG → UE: IKEv2 AUTH_RESP** ePDG sends IKE_AUTH_RESP (EAP Payload) The EAP payload shall contain the TLS message as received from the AAA server.

23. **UE → ePDG: IKEv2 AUTH_REQ** UE sends IKE_AUTH request (AUTH) The UE takes its own copy of the MSK as input to generate the AUTH parameter to authenticate the first IKE_SA_INIT message.

24. **ePDG → PGW: S2b Create Session Req** ePDG sends Create Session Request (IMSI, [MSISDN], Serving Network, RAT Type (WLAN), Indication Flags, Sender F-TEID for C-plane, APN, Selection Mode, PAA, APN-AMBR, [APCO], Bearer Contexts, [Recovery], [Private IE (P-CSCF)]). Selection Mode shall be set to "MS or network provided APN subscribed verified". The PGW performs the necessary interactions with 3GPP-AAA, PCRF and OCS/OFCS. ePDG shall set the HO in Indication flags IE and also the preserved IP address as received from UE in PAA IE.

25. **PGW → ePDG: Create Session Resp** The PGW allocates the requested IP address session and responds back to the ePDG with a Create Session Response (Cause, PGW S2b F-TEID, PAA, [APN-AMBR], [APCO], Bearer Contexts Created (EPS Bearer ID, Cause, [TFT], S2b-U PGW F-TEID, Bearer Level QoS), [Recovery], [Private IE (P-CSCF)]) message.

26. **ePDG → UE: IKEv2 AUTH_RESP** ePDG sends IKE_AUTH_RESP (AUTH, CP, SA, CFG_REPLY([INTERNAL_IP4_ADDRESS], [INTERNAL_IP4_NETMASK], [INTERNAL_IP4_DNS], INTERNAL_IP6_ADDRESS, INTERNAL_IP6_SUBNET, INTERNAL_IP6_DNS, [P-CSCF]) TSi, TSr) At this stage the ePDG has completed the ipsec SA and tunnel setup and also GTP-U tunnel setup thus completing the data path. The IP address provided by PGW is communicated to UE.

27. **ePDG → UE: IPv6 RA** The assumption is that the IP stack needs the RA to initialize the address.

### Emergency APN Support on ePDG

To support VoWiFi calls on ePDG should support emergency APN session. For areas where the LTE coverage is less or absent then the user shall utilize the WiFi to be able to perform the emergency session via ePDG.

In release 17.1 ePDG shall be handling the emergency sessions for authenticated UICC UEs. The un-authenticated and non UICC UEs is out of scope of this release 17.1. A new ePDG-APN bulkstats schema is added for capturing the APN level ePDG service statistics which shall be useful for emergency APN related statistics capture..

### Emergency APN Support Use Cases

<table>
<thead>
<tr>
<th>S.No</th>
<th>Use Case</th>
<th>Expected Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>ePDG receives the emergency session with UE indicating the emergency APN connectivity request for UE whose profile is present at AAA/HSS.</td>
<td>Call should be successfully established.</td>
</tr>
</tbody>
</table>
Evolved Packet Data Gateway Overview

Features and Functionality

<table>
<thead>
<tr>
<th>S.No</th>
<th>Use Case</th>
<th>Expected Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</td>
<td>ePDG receives the emergency session with UE indicating the emergency APN connectivity request for UE whose authentication fails at AAA.</td>
<td>ePDG shall be rejecting the call.</td>
</tr>
<tr>
<td>3.</td>
<td>Local PGW configured within the Emergency APN support and dynamic PGW selection fails as DNS server does not respond.</td>
<td>ePDG shall be utilizing the APN profile configuration and establish call with local configured PGW.</td>
</tr>
<tr>
<td>4.</td>
<td>Local PGW configured within the Emergency APN support and PGW obtained from dynamic PGW selection fails does not responds.</td>
<td>ePDG shall be utilizing the APN profile configuration and establish call with local configured PGW.</td>
</tr>
<tr>
<td>5.</td>
<td>Local configuration based PGW selection is configured as preferred way of PGW selection corresponding to emergency APN profile.</td>
<td>ePDG shall be utilizing the APN profile configuration and establish call with local configured PGW.</td>
</tr>
</tbody>
</table>

Passing on UE Tunnel Endpoint Address over SWm Support

Mobile operators would like to be able to block Vowifi calls from users while roaming. It is required that the tunnel endpoint (WLC or AP) IP address to be passed on from ePDG. This is very important to the operator as it generates a huge amount of revenue from roaming calls and would like to minimise the revenue leakage from users making Vowifi calls while roaming.

How Passing on UE tunnel Endpoint Address over SWm works

The provisioning of UE Tunnel Endpoint-IP (IKEv2 tunnel endpoint incase of NAT) to AAA server will help the operator in identifying the UE's location at AAA server. The operator uses this information to control the access or to decide the UE connections. For example, Operator can lookup the GeoIP database (GeoDB) against the UE tunnel endpoint IP to identify the country from where the UE is connecting from. Based on this information operator can allow the call or reject it(using auth-failure) according to the policy configured. Lets say the policy dictates that the VoWiFi calls are allowed only for UEs connecting from home country but not allowed while roaming outside the country, they can save the revenue leakage using this information.

The value will be sent in UE-Local-IP-Address AVP(IPv4/IPv6) in all the DER messages to AAA server in SWm interface. The AVP is sent as part of standard SWm dictionary (aaa-custom16). In case of AAA server rejects the call based on the tunnel endpoint IP, ePDG will send AUTHENTICATION_FAILED/24 as NOTIFY error message in IKEv2 message to communicate the same to UE.

This feature is supported for EAP based authentication mechanism and not for non UICC deployment using certificate based device authentication.

Custom SWm to SWu error code mapping

ePDG does supports mapping of SWm to SWu error codes so that device can identify whether its temporary failure or permanent and can accordingly try connecting to the ePDG.

The communication service providers (CSP) would like the ability to take different actions depending on the severity of the error received from the AAA (SWm interface). If there is a temporary congestion in the network, a retry is appropriate.
In compliance with RFC 5996 2.21.2 ePDG sends AUTHENTICATION_FAILED/24 as Notify Error message type in IKE_AUTH_RESP message on SWu interface for all the SWm interface error codes.

The ePDG needs mapping of SWm to SWu error codes for communicating different error codes to device, enabling device to identify whether its temporary failure or permanent and can accordingly try connecting to the ePDG.

The ePDG continues to release the call while notifying the UE about the SWm error, however the UE based on error code shall take decision when to try connecting again.

For the mapping ePDG uses Notify Error Message type between 31 to 8191 from the range reserved for IANA or from the private range 8192 to 16383.

ePDG Bearer Duration KPIs

ePDG supports QCI based bearer duration information display at more granular level to enable customers to Monitor VoWiFi dedicated bearers.

For more information on show subscriber statistics and for show session duration commands refer CLI Reference Guide.

Data Buffering Support for DL Packets Before Session Establishment

To establish ePDG call once the PGW sends the create session response message to ePDG the call setup is complete at PGW and Downlink traffic may come. However on ePDG processing of create session response and setting up of IPsec tunnel may take small duration, so it is required that before bearer establishment and IPsec tunnel establishment is completed ePDG should have capability to buffer the data. In case of handover especially when the LTE bearer is torn down after sending create session response the downlink traffic shall be sent over the WLAN so this becomes even more important to buffer data on ePDG avoiding any traffic loss.

3GPP standards section 8.6.2 "Handover from 3GPP access to untrusted Non-3GPP IP Access with GTP on S2b" indicates that traffic can come from PGW to ePDG before even the session setup is done at ePDG (during the processing of create session response at ePDG).

Downlink DSCP Marking(SWu)

The ESP IP header of the downlink packet in SWu interface sent out of ePDG has the TOS value copied from the inner IP payload of the ESP packet. But as per the customer requirement the TOS value should be taken from the configuration or GTPU IP header received on S2B side.

Functional description

The ePDG marks the DSCP value in the ESP IP header while sending out in the SWu interface(both IPv4 and IPv6) based on the following order of priority:

1. DSCP Configuration per QCI
   - Use command `qci num downlink encaps-header dscp-marking dscp-marking-value` to configure marking of specific DSCP in downlink direction per QCI.

2. From GTPU header received from PGW.

Download DSCP marking feature is backward compatible, where the Inner-GTP IP packet(S2B) DSCP value should be copied to the outer ESP IP packet(SWu). Use command `qci num downlink encaps-header copy-inner dscp-marking-value` to enable copying of DSCP value from inner-gtp-ip packet header(S2B) to the outer-esp header(SWu)
DSCP marking is supported in different platforms like ASR5000, ASR5500 and VPC-Si/Di.

**ePDG Fast Re-Auth Support**

The UEs accessing through ePDG can perform multiple reattach due to movement across/within WLAN Network and can also access multiple PDN at the same time. In these cases, the UE authentication is performed frequently with AAA-server involving HSS node interaction for EAP-AKA algorithm.

The operator providing the untrusted WLAN access solution through ePDG can enable fast-reauthentication in AAA-server and UE in order to perform faster authentication and reduce the load in HSS. This is because the fast-reauthentication uses the keys derived in the previous full-authentication. Also fast-reauthentication helps the operator to enable local-policy in UE node to authenticate itself to AAA server periodically for enhanced security.

**Reattach with fast-reauth-id Call flow**

Below call flow describes Reattach with fast-reauth-id.
1. Call-1 established for APN-1 and AAA-Server has provided fast-reauth-id during this authentication process. ePDG will store mapping between IMSI and fast-reauth-id.

2. UE starts Call-2 for fast-reauthentication by sending the IKE-SA-INIT message to ePDG. IKE-SA established between UE and ePDG with IKE-SA_INIT message exchange.

3. The UE sends the fast-reauth-id in NAI format (fast-reauth-id@realm) in the IDi payload and the APN-1 (in the IDr payload) in this first message of the IKE_AUTH phase, and begins negotiation of child security associations. The UE omits the AUTH parameter in order to indicate to the ePDG that it wants to use EAP over IKEv2. The UE includes the configuration payload (CFG_REQUEST) within the IKE_AUTH request message to give indication that it needs to reconfigure the IP address.

4. ePDG Identifies the previous session based on the received fast-reauth-id as it already created mapping. ePDG will handle this request as new session. This is because the presence of CP-Payload indicates that the call
should be established till PGW without retaining the IP address(S2B interface). The ePDG sends the Diameter-EAP-Request message to the 3GPP AAA Server, containing the fast-reauth-id and APN.

**Important:** Please note that ePDG uses the new diameter-session-id here as it is creating a new session.

5. The 3GPP AAA Server shall validate the fast-reauthentication-id and initiates the fast re-authentication request.

**Important:** Please note that there is no communication with HSS/HLR at this stage since fast-reauthentication-id is used. This makes the procedure faster and reduce load in HSS.

6. The ePDG responds with its identity, a certificate, and sends the AUTH parameter to protect the previous message it sent to the UE (in the IKE_SA_INIT exchange). The EAP message received from the 3GPP AAA Server (EAP-Request/Fast-Reauthentication) is included in order to start the EAP procedure over IKEv2.
7. The ePDG checks the authentication parameters and responds to the fast-reauthentication. The only payload (apart from the header) in the IKEv2 message is the EAP message.
8. The ePDG forwards the EAP-Response/Fast-Reauthentication message to the 3GPP AAA Server.
9. The AAA checks, if the Fast-Reauthentication response is correct. When all checks are successful, the 3GPP AAA Server sends the final Diameter-EAP-Answer(with a result code indicating success) including the users IMSI, relevant service authorization information, an EAP success and the key material to the ePDG. This key material shall consist of the MSK generated during the fast-reauthentication process. AAA-Server shall include the SN-Fast-Reauth-Username AVP with the fast-reauthentication-id value given to UE in step 5. ePDG creates mapping between IMSI and Fast-reauthentication-ID at this point.
10. The EAP Success message is forwarded to the UE over IKEv2.
11. The UE shall take its own copy of the MSK as input to generate the AUTH parameter to authenticate the first IKE_SA_INIT message. The AUTH parameter is sent to the ePDG in IKE_AUTH message.
12. The ePDG checks the correctness of the AUTH received from the UE. At this point the UE is authenticated. In S2b interface, ePDG initiates GTPv2 signaling between ePDG and PDN GW for creating the default-bearer for APN by sending Create-Session-Request to PGW with UE/APN details and request for IP-address allocation.
13. PGW responds with the Create-Session-Response message containing the allocation IP address, QoS details for this default-bearer connection.
14. The ePDG calculates the AUTH parameter which authenticates the second IKE_SA_INIT message. The ePDG sends the assigned Remote IP address in the configuration payload (CFG_REPLY) in the IKE_AUTH_RESPONSE message to UE. Fast-reauthentication is completed and Call-2 is connected now.
15. ePDG initiates the IKE-SA INFO_DELETE message for Call-1 to UE to delete the IKE-SA as part of call deletion.
16. UE responds with IKE-SA INFO_DELETE to delete the IKE-SA.
17. Call-1 is disconnected at ePDG.
Multi-pdn with fast-reauth-id Call flow

Figure 12. Multi-pdn with fast-reauth-id (fast-reauth-id Per UE case with CP Payload)

1. Call-1 established for APN-1 and receive Fast-reauth-id-1.
2. IKE_SA_INIT Request and Response done.
3. Call-2: IKE_AUTH Request for APN-2 Idi Payload
   Type: RFC822_ADDR
   Value: Fast-reauth-id-1 (NAI format) CP Payload
4. DER: [Call-2]
   [EAP_PAYLOAD AVP with RESPONSE]
5. AAA support per-UE fast-reauth. So fast-reauth-id-1 is authenticated for APN-2.
6. DEA: [EAP_PAYLOAD AVP with REQUEST]
7. IKE_AUTH Response
   EAP Payload
8. IKE_AUTH Request
   EAP Payload
9. DER:
   [EAP_PAYLOAD AVP with RESPONSE]
10. DEA:
    [EAP_PAYLOAD AVP with SUCCESS]
    [SN-Fast-Reauth-Username AVP with FastReauthId] 
    [MobileNodeIdentifier AVP with NAI having IMSI]
11. IKE_AUTH Response
12. IKE_AUTH Request
13. CSRequest [Call2]
14. CSResponse
15. IKE_AUTH Response
16. Call-2 comes up for APN-2. Call-1 also exists for APN-1.

1. Call-1 established for APN-1 and AAA-Server has provided fast-reauth-id-1 during this authentication process. ePDG will store mapping between imsi and fast-reauth-id-1
2. UE starts Call-2 to connect to APN-2 using the fast-reauth-id-1. IKE-SA established between UE and ePDG with IKE_SA_INIT message exchange.
3. The UE sends the fast-auth-id-1 in NAI format(fast-reauth-id-1@realm) in the IDi payload and the APN-2 (in the IDR payload) in this first message of the IKE_AUTH phase, and begins negotiation of child security associations. The UE omits the AUTH parameter in order to indicate to the ePDG that it wants to use EAP over IKEv2. The UE includes the configuration payload (CFG_REQUEST) within the IKE_AUTH request message to give indication that it needs to configure the IP address.
4. ePDG Identifies the previous session based on the received Fast-reauth-id as it already created mapping. ePDG will handle this request as new session. This is because the request is for new APN and also the presence of CP-Payload indicates that the call should be established till PGW without retaining the IP address(S2B interface). The ePDG sends the Diameter-EAP-Request message to the 3GPP AAA Server, containing the fast-reauth-id-1 and APN-2. Please note that ePDG uses the new diameter-session-id here as it is creating a new session.

5. AAA server supports Fast-Reauthentication on per-UE basis. Hence it accepts fast-reauth-id-1 for APN-2.

6. The 3GPP AAA Server validates the fast-reauth-id-1 and initiates the fast re-authentication request.

**Important:** Please note that there is no communication with HSS/HLR at this stage since fast-reauth-id-1 is used. This makes the procedure faster and reduce load in HSS.

7. The ePDG responds with its identity, a certificate, and sends the AUTH parameter to protect the previous message it sent to the UE (in the IKE_SA_INIT exchange). The EAP message received from the 3GPP AAA Server (EAP-Request/Fast-Reauthentication) is included in order to start the EAP procedure over IKEv2.

8. The UE checks the authentication parameters and responds to the fast-reauthentication. The only payload (apart from the header) in the IKEv2 message is the EAP message.

9. The ePDG forwards the EAP-Response/Fast-Reauthentication message to the 3GPP AAA Server.

10. The AAA checks if the Fast-reauthentication response is correct. When all checks are successful, the 3GPP AAA Server sends the final Diameter-EAP-Answer(with a result code indicating success) including the users IMSI, relevant service authorization information, an EAP success and the key material to the ePDG. This key material consists of the MSK generated during the fast-reauthentication process. AAA-Server includes the SN-Fast-Reauth-Username AVP with the fast-reauthentication-id value given to UE in step 5. ePDG creates mapping between IMSI and Fast-reauthentication-ID at this point.

11. The EAP Success message is forwarded to the UE over IKEv2.

12. The UE shall take its own copy of the MSK as input to generate the AUTH parameter to authenticate the first IKE_SA_INIT message. The AUTH parameter is sent to the ePDG in IKE_AUTH message.

13. The ePDG checks the correctness of the AUTH received from the UE. At this point the UE is authenticated. In S2b interface, ePDG initiates GTPv2 signaling between ePDG and PDN GW for creating the default-bearer for APN by sending Create-Session-Request to PGW with UE/APN details and request for IP-address allocation.

14. PGW responds with the Create-Session-Response message containing the allocation IP address, QoS details for this default-bearer connection.

15. The ePDG calculates the AUTH parameter which authenticates the second IKE_SA_INIT message. The ePDG shall send the assigned Remote IP address in the configuration payload (CFG_REPLY) in the IKE_AUTH_RESPONSE message to UE.

16. Call-2 is connected now for APN-2 and Call-1 already exists for APN-1.

**ePDG Offline charging**

Offline charging is a process where charging information is collected concurrently with that resource usage. The charging information is then passed through a chain of logical charging functions. At the end of this process, CDR files are generated by the network, which are then transferred to the network operator's Billing Domain(BD). Charging information like amount of data transmitted in uplink and downlink direction are collected as part of ePDG-CDR are used to inter-operator settlements.

**ePDG Offline charging Architecture**

The ePDG Offline charging involves the following functionalists for WLAN 3GPP IP Access:

- Charging Trigger Function
The Charging Trigger Function (CTF) which is an integrated component generates charging events and forwards them to the Charging Data Function (CDF). The CDF, in turn generates ePDG-CDRs which are then transferred to the CGF. Finally, the CGF create ePDG-CDR files and forwards them to the Billing Domain. The CTF and CDF are integrated in ePDG, however, the CGF may exist as a physically separate entity or integrated to ePDG. If the CGF is external to the ePDG, then the CDF forwards the CDRs to the CGF across the Ga interface (using GTPP protocol defined in TS 32.295). ePDG-CDR format is as defined in TS 32.298 v12.6.0.
**Evolved Packet Data Gateway Overview**

**Features and Functionality**

**Figure 14. ePDG Offline Charging Callflow**

1. UE established call with ePDG as per standard procedure. Charging-Char assigned by AAA-server during authentication process for billing purpose.

2. Accounting context selected based on the configuration to identify the GTPP group to be used. This group contains details about dictionary, external charging-agent address and options to enable fields in the ePDG-CDR.
   - Accounting profile selected based on the configuration and charging-char to identify various triggers for charging information addition/closure.
   - Network resource usage collected in the form of number of packets/bytes sent/received etc. for record-addition triggers for the subscriber charging-char.
   - ePDG CDRs generated on the record-closure triggers

3. ePDG-CDRs sent using GTPP protocol

**ePDG Offline Charging**

ePDG supports CDRs to bill the UEs for network resource usage as defined in 3GPP specification TS32.298.

Apart from the standard ePDG-CDR fields ePDG Offline Charging feature populates the following additional fields:
- IKEv2 tunnel endpoint IP address (UE Side tunnel endpoint address)
- Source Port number used in IKEv2 tunnel
- ePDG SWu interface IP address (ePDG side tunnel endpoint address)
- Destination Port number used in IKEv2 tunnel
- AP-MAC address used by UE to connect in WLAN network

Custom24 is the GTPP dictionary for standard ePDG-CDR as per specifications and custom38 is the GTPP dictionary for CDRs with above additional fields.
Supported Triggers for ePDG-CDRs Charging Information Addition

The “List of Traffic Volumes” attribute of the ePDG-CDR consists of a set of containers, on encountering the following trigger conditions, the charging information will be added to the container:

**QoS Change:** A change in the QoS will result to open the “List of Traffic Data Volumes” container being closed and added to the CDR and a new bearer specific container is opened. Also when there is a Change in QoS, the trigger will be sent to accounting module for CDR information addition using the API "sessmgr_acct_api_event(handle, params)" with the params --> qos_change set from the sessmgr.

**Tariff time:** On reaching the Tariff Time Change open “List of Traffic Data Volumes” containers is closed and added to the CDR. Tariff-time change is to add charging information to CDR during a particular tariff-time of day.

**Record Closure:** Open “List of Traffic Data Volumes” containers is closed and added to the ePDG-CDR.

Supported Triggers for ePDG-CDR closure

The following events trigger closure and sending of ePDG-CDR:

- **Time Limit:** CDRs are generated after every x seconds where x is the configured time limit.
- **Max container Triggers:** Maximum number of charging condition changes (QoS/tariff time change). CDRs are generated when the max bucket limit is reached. By default its 4.
- **Volume limit trigger:** CDRs are generated whenever the uplink or downlink data volume for the session crosses the configured uplink, downlink or total limit.
- **Management Intervention:** CDRs can be generated by management intervention such as clear command issued by the operator to cleanup a session.

Assumptions and Limitations

- The AP-MAC address will be populated in ePDG-CDR only when it is supplied by UE during initial IKEv2 exchange in IDi payload as expected by ePDG. Please see the ePDG admin guide to understand the format of IDi payload with AP-MAC address encoded in it.
- The CDF functionality is integrated within ePDG. RF interface is not support.

UE Local IP Address IE in the S2B Interface over GTPv2

This chapter describes UE Local IP Address IE in the S2B Interface over GTPv2 feature, below are the links to main sections of the document:

The location of the UE initiating a VoWifi call via ePDG will be identified based on the UE local IP address reported on S2B interface. This location information can be used for multiple purposes like billing and lawful interception etc.

Below is the specifications of the “UE Local IP Address IE in the S2B Interface over GTPv2” feature:

- ePDG saves the UE local IP address (SRC address of the IKE messages received from UE) and the port (SRC port of the IKE message received from UE) and send them to PGW over S2B interface.
- The port information must be sent on S2B interface only when a NAT is detected between UE and ePDG (UE is behind a NAT).
- CLI configuration is supported to control the inclusion of the UE local IP Address and port on the S2B interface.
- The above functionality needs to be supported only for GTPv2 based S2B interface.

How It Works

This section describes signaling flow during an ePDG session setup procedure.
Figure 15. ePDG Session Setup Procedure

The above figure shows the signaling flow during an ePDG session setup procedure:

1. The IKEv2 procedure starts with the IKE_INIT (step 2) message received at ePDG from UE. The SRC address and port of the IKE_INIT message is recorded at ePDG and NAT detection is done as defined in RFC 5996.
2. The IKE_INIT message triggers the IKEv2 tunnel setup and after the IKE_INIT_RESP in step 3, the UE sends IKE_AUTH message (step 4).
3. This IKE_AUTH_REQ from UE triggers the multi round authentication with AAA server on SWm interface.
4. ePDG sends IKE_AUTH_RESP in step 11 to complete the EAP authentication.
5. The next IKE_AUTH_REQ from UE triggers the session setup towards the PGW over s2b interface and ePDG should include the "UE Local IP Address" and "UE UDP Port" (only if NAT detected) AVPs in the Create Session Request message.
**Important:** UE Local IP Address IE in the S2B Interface over GTPv2" supports only for GTPv2 based s2b interface.

**Detailed Description**

Following table summarizes the expected behavior for UE Local IP Address IE in the S2B Interface over GTPv2 feature.

<table>
<thead>
<tr>
<th>SR.No</th>
<th>AVP Inclusion via configuration</th>
<th>NAT detected on SWu Interface</th>
<th>Expected Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Enabled</td>
<td>Yes</td>
<td>Both “UE Local IP Address” and “UE UDP Port” AVPs are sent in Create Session Request message to PGW.</td>
</tr>
<tr>
<td>2.</td>
<td>Enabled</td>
<td>No</td>
<td>Only “UE Local IP Address” AVP is sent in Create Session Request message to PGW.</td>
</tr>
<tr>
<td>3.</td>
<td>Disabled</td>
<td>Yes</td>
<td>Both “UE Local IP Address” and “UE UDP Port” AVPs are NOT sent in Create Session Request message to PGW.</td>
</tr>
<tr>
<td>4.</td>
<td>Disabled</td>
<td>No</td>
<td>Both “UE Local IP Address” and “UE UDP Port” AVPs are NOT sent in Create Session Request message to PGW.</td>
</tr>
</tbody>
</table>

**External Interfaces**

This feature impacts the GTPv2 based s2b interface towards PGW. The following two AVPs as defined in 3GPP 29.274 are included in the Create Session Requested message as per the conditions mentioned in the above table.

Table 10. GTPv2 IE Definition for UE Local IP Address and UE UDP Port

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Condition</th>
<th>Description</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>UE Local IP Address</td>
<td>CO</td>
<td>The ePDG should include this IE on S2b interface based on local policy for Fixed Broadband access network interworking see 3GPP in TS 23.139 [51].</td>
<td>IP Address</td>
</tr>
<tr>
<td>UE UDP Port</td>
<td>CO</td>
<td>The ePDG shall include this IE on S2b interface if NAT is detected and UE Local IP Address is present for Fixed Broadband access network interworking see 3GPP in TS 23.139 [51].</td>
<td>Port Number</td>
</tr>
</tbody>
</table>

**Note:** Even though the 3GPP specification mentions the usage of the AVPs in the context of Fixed Broadband access network, they are being used for untrusted WiFi access in this case.
How the ePDG Works

This section describes the ePDG during session establishment and disconnection.

ePDG Session Establishment

The figure below shows an ePDG session establishment flow. The table that follows the figure describes each step in the flow.
Figure 16. ePDG Session Establishment

Table 11. ePDG Session Establishment

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>IKEv2 SA_INIT Request</td>
</tr>
<tr>
<td>2.</td>
<td>IKEv2 SA_INIT Response</td>
</tr>
<tr>
<td>3.</td>
<td>IKEv2 AUTH Request</td>
</tr>
<tr>
<td>4.</td>
<td>DER (EAP-Payload, User-Name (NAI))</td>
</tr>
<tr>
<td>5.</td>
<td>DEA (EAP Request/AAA Challenge)</td>
</tr>
<tr>
<td>6a.</td>
<td>UE runs AKA algorithms, verifiesAUTN, and generates RES and MSK</td>
</tr>
<tr>
<td>7.</td>
<td>IKEv2 AUTH Request</td>
</tr>
<tr>
<td>8.</td>
<td>DER (EAP Response/AAA Challenge)</td>
</tr>
<tr>
<td>8a.</td>
<td>3GPP AAA server verifies ifAT_RES = XRES</td>
</tr>
<tr>
<td>9.</td>
<td>DEA (EAP Success, Key Material, IMSI)</td>
</tr>
<tr>
<td>10.</td>
<td>AUTH payload is computed using the keying material</td>
</tr>
<tr>
<td>11.</td>
<td>IKEv2 AUTH Request</td>
</tr>
<tr>
<td>12.</td>
<td>IKEv2 AUTH Request</td>
</tr>
<tr>
<td>12a.</td>
<td>Checks AUTH correctness</td>
</tr>
<tr>
<td>13.</td>
<td>PBU</td>
</tr>
<tr>
<td>14.</td>
<td>PBA</td>
</tr>
<tr>
<td>15.</td>
<td>Calculate AUTH</td>
</tr>
<tr>
<td>16.</td>
<td>IKEv2 AUTH Response</td>
</tr>
<tr>
<td>17.</td>
<td>IPv6 Router Advertisement</td>
</tr>
<tr>
<td>18.</td>
<td>Delete old IKE SA if it already exists for this APN</td>
</tr>
<tr>
<td>19.</td>
<td>User data</td>
</tr>
</tbody>
</table>
### How the ePDG Works

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The WLAN UE initiates an IKEv2 exchange with the ePDG by issuing an IKEv2 SA_INIT Request message to negotiate cryptographic algorithms, exchange nonces, and perform a Diffie-Hellman exchange with the ePDG.</td>
</tr>
<tr>
<td>2.</td>
<td>The ePDG returns an IKEv2 SA_INIT Response message.</td>
</tr>
<tr>
<td>3.</td>
<td>The UE sends the user identity in the Idi payload and the APN information in the IDR payload in the first message of the IKE_AUTH phase and begins negotiation of Child SAs. The UE omits the AUTH parameter in order to indicate to the ePDG that it wants to use EAP over IKEv2. The user identity is compliant with the NAI (Network Access Identifier) format specified in TS 23.003 and contains the IMSI as defined for EAP-AKA in RFC 4187. The UE sends the CP payload (CFG_REQUEST) within the IKE_AUTH Request message to obtain an IPv4 and/or IPv6 home IP address and/or a home agent address. The root NAI is in the format “0&lt;IMSI&gt;@nai.epc.mnc&lt;MNC&gt;.mcc&lt;MCC&gt;.3gppnetwork.org”.</td>
</tr>
<tr>
<td>4.</td>
<td>The ePDG sends a DER (Diameter EAP Request) message containing the user identity and APN to the 3GPP AAA server.</td>
</tr>
<tr>
<td>5.</td>
<td>The 3GPP AAA server fetches the user profile and authentication vectors from the HSS/HLR if these parameters are not available on the 3GPP AAA server. The 3GPP AAA server looks up the IMSI of the authenticated user based on the received user identity (root NAI) and includes EAP-AKA as the requested authentication method in the request sent to the HSS. The HSS generates the authentication vectors with the AMF separation bit = 0 and sends them back to the 3GPP AAA server. The 3GPP AAA server checks the user’s subscription information to verify that the user is authorized for non-3GPP access. The 3GPP AAA server increments the counter for IKEv2 SAs. If the maximum number of IKE SAs for the associated APN is exceeded, the 3GPP AAA server sends an indication to the ePDG that established the oldest active IKEv2 SA (it could be the same ePDG or a different one) to delete the oldest IKEv2 SA. The 3GPP AAA server updates its total active IKEv2 SAs for the APN. The 3GPP AAA server initiates the authentication challenge and responds with a DEA (Diameter EAP Answer). The user identity is not requested again.</td>
</tr>
<tr>
<td>6.</td>
<td>The ePDG responds with its identity (a certificate) and sends the AUTH parameter to protect the previous message it sent to the UE in the IKEv2 SA_INIT exchange. It completes the negotiation of the Child SAs, if any. The EAP Request/AKA Challenge message received from the 3GPP AAA server is included in order to start the EAP procedure over IKEv2.</td>
</tr>
<tr>
<td>6a.</td>
<td>The UE checks the authentication parameters.</td>
</tr>
<tr>
<td>7.</td>
<td>The UE responds to the authentication challenge with an IKEv2 AUTH Request message. The only payload apart from the header in the IKEv2 message is the EAP Response/AKA Challenge message.</td>
</tr>
<tr>
<td>8.</td>
<td>The ePDG forwards the EAP Response/AKA Challenge message to the 3GPP AAA server in a DER message.</td>
</tr>
<tr>
<td>8a.</td>
<td>The 3GPP AAA server checks if the authentication response is correct.</td>
</tr>
<tr>
<td>9.</td>
<td>When all checks are successful, the 3GPP AAA server sends the final DEA (with a result code indicating EAP success) that includes the relevant service authorization information and key material to the ePDG. The key material consists of the MSK generated during the authentication process. The MSK is encapsulated in the EAP-Master-Session-Key-AVP as defined in RFC 4072.</td>
</tr>
<tr>
<td>10.</td>
<td>The MSK is used by the ePDG to generate the AUTH parameters in order to authenticate the IKEv2 SA_INIT messages as specified for IKEv2 in RFC 4306. These first two messages had not been authenticated earlier as there was no key material available yet. Per RFC 4304, the shared secret generated in an EAP exchange (the MSK) when used over IKEv2 must be used to generate the AUTH parameters.</td>
</tr>
<tr>
<td>11.</td>
<td>The EAP Success/Failure message is forwarded to the UE over IKEv2.</td>
</tr>
<tr>
<td>12.</td>
<td>The UE takes its own copy of the MSK as input to generate the AUTH parameter to authenticate the first IKEv2 SA_INIT message. The AUTH parameter is sent to the ePDG.</td>
</tr>
<tr>
<td>12a.</td>
<td>The ePDG checks the correctness of the AUTH parameter received from the UE. At this point the UE is authenticated.</td>
</tr>
<tr>
<td>Step</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>13.</td>
<td>On successful authentication, the ePDG establishes the PMIP tunnel towards the P-GW by sending a PBU (Proxy-MIP Binding Update), which includes the NAI and APN and the Home Network Prefix or IPv4 Home Address option.</td>
</tr>
<tr>
<td>14.</td>
<td>The P-GW allocates the requested IP address (IPv4/IPv6 or both) session and responds back to the ePDG with a PBA (Proxy-MIP Binding Acknowledgement).</td>
</tr>
<tr>
<td>15.</td>
<td>The ePDG calculates the AUTH parameter that authenticates the second IKEv2 SA_INIT message.</td>
</tr>
<tr>
<td>16.</td>
<td>The ePDG sends the AUTH parameter, the assigned remote IP address in the CP payload, the SAs, and the rest of the IKEv2 parameters to the UE, and IKEv2 negotiation is complete.</td>
</tr>
<tr>
<td>17.</td>
<td>The ePDG sends an IPv6 Router Advertisement to the UE to ensure that the IPv6 stack is fully initialized.</td>
</tr>
<tr>
<td>18.</td>
<td>If the ePDG detects that an old IKEv2 SA for the APN already exists, it deletes the IKEv2 SA and sends an INFORMATIONAL exchange with a DELETE payload to the UE to delete the old IKEv2 SA in the UE as specified in RFC 4306.</td>
</tr>
<tr>
<td>19.</td>
<td>The ePDG session/IPSec SA is fully established and ready for data transfer.</td>
</tr>
</tbody>
</table>

**UE-initiated Session Disconnection**

The figure below shows the message flow during a UE-initiated session disconnection. The table that follows the figure describes each step in the message flow.

**Figure 17.** UE-initiated Session Disconnection

![Diagram](image)
Table 12. UE-initiated Session Disconnection

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The UE sends an INFORMATIONAL Request. The Encrypted Payload has a single Delete Payload which contains the SPI of the IKEv2 SA corresponding to the WLAN UE session to be disconnected.</td>
</tr>
<tr>
<td>2.</td>
<td>On receiving the IKEv2 INFORMATIONAL Request with Delete from the UE, the ePDG begins the disconnection of the WLAN UE session. It begins the tear down the session by sending PBU for deregistration to P-GW to disconnect the session.</td>
</tr>
<tr>
<td>3.</td>
<td>P-GW sends back the PBA message acknowledging the session deletion.</td>
</tr>
<tr>
<td>4.</td>
<td>The ePDG responds back to the UE's IKEv2 INFORMATION request with a IKEv2 INFORMATIONAL RSP.</td>
</tr>
<tr>
<td>6.</td>
<td>3GPP AAA clears the SWn sessions and responds back to the ePDG with a Session-Terminate-Ack (STA).</td>
</tr>
</tbody>
</table>

Figure 18. UE initiated Session Disconnection - GTPv2

Table 13. UE-initiated Session Disconnection GTPv2

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The UE sends an INFORMATIONAL Request. The Encrypted Payload has a single Delete Payload which contains the SPI of the IKEv2 SA corresponding to the WLAN UE session to be disconnected.</td>
</tr>
</tbody>
</table>
Table 14. ePDG-initiated Session Disconnection

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>An Admin/AAA trigger causes the ePDG to start disconnecting the WLAN UE session by sending an IKEv2 INFORMATIONAL (DELETE) Request message. The encrypted payload has a single DELETE payload that contains the SPI of the IKEv2 SA corresponding to the WLAN UE session being disconnected.</td>
</tr>
</tbody>
</table>
### How the ePDG Works

**ePDG Administration Guide, StarOS Release 18**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</td>
<td>The ePDG also begins to tear down the S2b PMIP session by sending a PBU (Proxy-MIP Binding Update) De-registration message to the P-GW.</td>
</tr>
<tr>
<td>3.</td>
<td>The ePDG responds to the UE’s IKEv2 INFORMATIONAL (DELETE) Request message with an IKEv2 INFORMATIONAL (DELETE) Response message.</td>
</tr>
<tr>
<td>4.</td>
<td>On receiving the PBU (Proxy-MIP Binding Update) De-registration message, the P-GW disconnects the UE session and releases local resources. The P-GW completes the disconnection of the WLAN UE session and responds to the ePDG with a PBA De-registration message.</td>
</tr>
<tr>
<td>5.</td>
<td>The ePDG disconnects the SWu session by sending an STR (Session Terminate Request) message to the 3GPP AAA/HSS.</td>
</tr>
<tr>
<td>6.</td>
<td>The 3GPP AAA clears the SWu sessions and responds to the ePDG with an STA (Session Terminate Acknowledgment) message.</td>
</tr>
</tbody>
</table>

### P-GW-initiated Session Disconnection

The figure below shows the message flow during a P-GW-initiated session disconnection. The table that follows the figure describes each step in the message flow.

**Figure 20.** P-GW-initiated Session Disconnection

Table 15. P-GW-initiated Session Disconnection

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
### WiFi-to-WiFi Re-Attach with same ePDG

The figure below shows the message flow if the UE loses connection to the ePDG and then reconnects using the same ePDG. The table that follows the figure describes each step in the message flow.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The PGW sends BRI (Binding revocation indication) to ePDG for disconnecting the session.</td>
</tr>
<tr>
<td>2.</td>
<td>The ePDG sends IKEv2 Informational Delete Request () to UE to disconnect the session.</td>
</tr>
<tr>
<td>3.</td>
<td>The ePDG sends BRA (Binding revocation acknowledgement) to PGW acknowledging the session disconnect.</td>
</tr>
<tr>
<td>4.</td>
<td>The UE sends IKEv2 Informational Delete Response ().</td>
</tr>
<tr>
<td>5.</td>
<td>ePDG sends STR (Session ID, Base AVPs, Termination Cause) to the 3GPP AAA.</td>
</tr>
<tr>
<td>6.</td>
<td>3GPP AAA clears the SWn sessions and responds back to the ePDG with a STA (Session ID, Base AVPs).</td>
</tr>
</tbody>
</table>
How the ePDG Works

Table 16. WiFi-to-WiFi Re-Attach

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The UE is authenticated and a PDN connection is established. This scenario addresses a case where the UE has ungracefully disconnected from the network and is reattaching to the network again.</td>
</tr>
<tr>
<td>Step</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>2.</td>
<td>The session is still active in the ePDG and P-GW along with AAA, PCRF and AAA.</td>
</tr>
<tr>
<td>3.</td>
<td>The step 2 through 12 are identical to the UE initial attach scenario defined in section 3.2.1. It is assumed that the UE will not populate the IP Addresses in the IKE Config Request.</td>
</tr>
<tr>
<td>4.</td>
<td>The ePDG shall be detecting the duplicate session and clearing the previous established session at its ends. Further ePDG shall be establishing new session on P-GW following below steps</td>
</tr>
<tr>
<td>15.</td>
<td>ePDG → UE: IKE_AUTH - The ePDG sends IKE_AUTH (AUTH, CP, SA, CFG_REPLY ([INTERNAL_IP4_ADDRESS], [INTERNAL_IP4_NETMASK], [INTERNAL_IP4_DNS], INTERNAL_IP6_ADDRESS, INTERNAL_IP6_SUBNET, INTERNAL_IP6_DNS, P-CSCF) TSi, TSr). The ePDG calculates the AUTH parameter, which authenticates the second IKE_SA_INIT message. The ePDG sends the assigned IP address in the configuration payload (CFG_REPLY). The AUTH parameter is sent to the UE together with the configuration payload, security associations and the rest of the IKEv2 parameters and the IKEv2 negotiation terminates.</td>
</tr>
<tr>
<td>16.</td>
<td>ePDG → P-GW: PBU (Proxy-MIP Binding Update) - The ePDG selects the P-GW based on DNS response from the APN-FQDN. The ePDG sends PBU (IMSI, [MSIDSN], Serving Network, RAT Type (WLAN), Indication Flags, Sender F-TEID for C-plane, APN, Selection Mode, PAA, APN-AMBR, Bearer Contexts), [Recovery], [Charging Characteristics], Private IE (P-CSCF). The F-TEID shall be set to zero so that P-GW shall handle the same as create-on-create case.</td>
</tr>
<tr>
<td>19.</td>
<td>P-GW → ePDG: PBA (Proxy-MIP Binding Acknowledgement) - The P-GW terminates the previous session by handling it as create on create case and establishes a new session. The P-GW allocates the requested IP address session and responds back to the ePDG with a PBA (Cause, P-GW S2b Address C-plane, PAA, [Recovery], APN-AMBR, Additional Protocol Configuration Option (APCO) Bearer Contexts Created, Private IE (P-CSCF)) message.</td>
</tr>
<tr>
<td>21.</td>
<td>ePDG → UE: Router Advertisement - The ePDG sends Router Advertisement to ensure IP Stack is fully initialized.</td>
</tr>
</tbody>
</table>
How the ePDG Works

Description:

The UE is authenticated and a PDN connection is established. This scenario addresses a case where the UE has ungracefully disconnected from the network and is reattaching to the network again.

The session is still active in the ePDG and P-GW along with AAA, PCRF and AAA.

The step 2 through 12 are identical to the UE initial attach scenario defined in section 3.2.1. It is assumed that the UE will not populate the IP Addresses in the IKE Config Request.
The ePDG detects the duplicate session and clears the previous established session at its ends. Then the ePDG establishes a new session on the P-GW using the following steps:

### Table 17. WiFi-to-WiFi Re-Attach - GTPv2

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.</td>
<td><strong>ePDG -&gt; P-GW: Create Session Request</strong> - The ePDG selects the P-GW based on DNS response from the APN-FQDN. The ePDG sends Create Session Request (IMSI, [MSIDSN], Serving Network, RAT Type (WLAN), Indication Flags, Sender F-TEID for C-plane, APN, Selection Mode, PAA, APN-AMBR, Bearer Contexts), [Recovery], [Charging Characteristics], Private IE (P-CSCF). The TEID shall be set to zero so that P-GW shall handle the same as create-on-create case.</td>
</tr>
<tr>
<td>14.</td>
<td><strong>P-GW -&gt; ePDG: Create Session Response</strong> - The P-GW terminates the previous session by handling it as create on create case and establishes a new session. The P-GW allocates the requested IP address session and responds back to the ePDG with a Create Session Response (Cause, P-GW S2b Address C-plane, PAA, [Recovery], APN-AMBR, Additional Protocol Configuration Option (APCO) Bearer Contexts Created, Private IE (P-CSCF)) message.</td>
</tr>
<tr>
<td>29.</td>
<td><strong>ePDG -&gt; UE: IKE_AUTH</strong> - The ePDG sends IKE_AUTH (AUTH, CP, SA, CFG_REPLY ([INTERNAL_IP4_ADDRESS], [INTERNAL_IP4_NETMASK], [INTERNAL_IP4_DNS], INTERNAL_IP6_ADDRESS, INTERNAL_IP6_SUBNET, INTERNAL_IP6_DNS, P-CSCF) TSi, TSr). The ePDG calculates the AUTH parameter, which authenticates the second IKE_SA_INIT message. The ePDG sends the assigned IP address in the configuration payload (CFG_REPLY). The AUTH parameter is sent to the UE together with the configuration payload, security associations and the rest of the IKEv2 parameters and the IKEv2 negotiation terminates.</td>
</tr>
<tr>
<td>30.</td>
<td><strong>ePDG -&gt; UE: Router Advertisement</strong> - ePDG sends Router Advertisement to ensure IP Stack is fully initialized.</td>
</tr>
</tbody>
</table>

### WiFi to LTE Handoff with Dedicated Bearer (UE initiated)

When a VoLTE call is ongoing, the P-GW will install the bearers on the LTE network using piggyback procedure.
Figure 23. WiFi to LTE Handoff with Dedicated Bearer - Part 1
The UE which was previously having a WiFi call attaches to the LTE.
Table 18. WiFi to LTE Handoff with Dedicated Bearer

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>26.</td>
<td>P-GW -&gt; ePDG: Delete Bearer Request - The P-GW sends Delete Bearer Request (EPS Bearer ID / LBI, Cause) to ePDG to disconnect the session. If releasing all the bearers LBI shall be set to the identity of the default bearer associated with the PDN connection. Cause shall be set to “Access changed from Non-3GPP to 3GPP”.</td>
</tr>
<tr>
<td>27.</td>
<td>ePDG -&gt; UE: IKEv2 Informational Delete Request - The ePDG sends IKEv2 Informational Delete Request () to UE to disconnect the session.</td>
</tr>
<tr>
<td>28.</td>
<td>ePDG -&gt; P-GW: Delete Bearer Response - The ePDG sends Delete Bearer Response (Cause, Linked EPS Bearer Identity, Bearer Context, [Recovery]) to P-GW.</td>
</tr>
<tr>
<td>29.</td>
<td>UE -&gt; ePDG: IKEv2 Informational Delete Response - UE responds with IKEv2 Information Delete Response () and initiates air interface resource release. Step is conditional and UE may not send this response.</td>
</tr>
<tr>
<td>30.</td>
<td>ePDG -&gt; AAA: Session Termination Request - The ePDG sends STR (Session ID, User-Name (IMSI-NAI), Termination-Cause) to the 3GPP AAA.</td>
</tr>
<tr>
<td>31.</td>
<td>AAA -&gt; ePDG: Session Termination Answer - The AAA sends STA (Session ID, Result-Code) to the ePDG.</td>
</tr>
</tbody>
</table>

LTE to WiFi Hand Off - With Dedicated bearer (UE initiated)

In this call flow we use the IMS PDN with an ongoing VoLTE call with the associated dedicated bearers. The UE detects suitable WiFi access point and connects to AP as per node selection.
Table 19. LTE to WiFi Hand Off - With Dedicated Bearer

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</td>
<td>UE -&gt; ePDG: The UE sends IKE_SA_INIT Message.</td>
</tr>
<tr>
<td>3.</td>
<td>ePDG -&gt; UE: The ePDG responds with IKE_SA_INIT_RSP Message.</td>
</tr>
</tbody>
</table>
### How the ePDG Works

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.</td>
<td>The UE sends the user identity (in the IDi payload) and the APN information (in the IDr payload) in this first message of the IKE_AUTH phase, and begins negotiation of child security associations. The UE omits the AUTH parameter in order to indicate to the ePDG that it wants to use EAP over IKEv2. The user identity shall be compliant with Network Access Identifier (NAI) format specified in TS 23.003 containing the IMSI, as defined for EAP-AKA in RFC 4187. The UE shall send the configuration payload (CFG_REQUEST) within the IKE_AUTH request message with the preserved IP address(es) from the LTE session so that ePDG knows its handoff case and communicates same IP address to P-GW. When the MAC ULI feature is enabled the root NAI used will be of the form “0&lt;IMSI&gt;@AP_MAC_ADDR:nai.epc.mnc&lt;MNC&gt;.mcc&lt;MCC&gt;.3gppnetwork.org”.</td>
</tr>
<tr>
<td>5.</td>
<td>The ePDG sends the Authentication and Authorization Request message to the 3GPP AAA Server, containing the user identity and APN.</td>
</tr>
<tr>
<td>6.</td>
<td>The 3GPP AAA Server shall fetch the user profile and authentication vectors from HSS/HLR (if these parameters are not available in the 3GPP AAA Server). The 3GPP AAA Server shall lookup the IMSI of the authenticated user based on the received user identity (root NAI) and include the EAP-AKA as requested authentication method in the request sent to the HSS. The HSS shall then generate authentication vectors with AMF separation bit = 0 and send them back to the 3GPP AAA server. The 3GPP AAA Server checks in user's subscription if he/she is authorized for non-3GPP access. The counter of IKE SAs for that APN is stepped up. If the maximum number of IKE SAs for that APN is exceeded, the 3GPP AAA Server shall send an indication to the ePDG that established the oldest active IKE SA (it could be the same ePDG or a different one) to delete the oldest established IKE SA. The 3GPP AAA Server shall update accordingly the information of IKE SAs active for the APN. The 3GPP AAA Server initiates the authentication challenge. The user identity is not requested again.</td>
</tr>
<tr>
<td>7.</td>
<td>The ePDG responds with its identity, a certificate, and sends the AUTH parameter to protect the previous message it sent to the UE (in the IKE_SA_INIT exchange). It completes the negotiation of the child security associations if any. The EAP message received from the 3GPP AAA Server (EAP-Request/AKA-Challenge) is included in order to start the EAP procedure over IKEv2.</td>
</tr>
<tr>
<td>8.</td>
<td>The UE checks the authentication parameters and responds to the authentication challenge. The only payload (apart from the header) in the IKEv2 message is the EAP message.</td>
</tr>
<tr>
<td>9.</td>
<td>The ePDG forwards the EAP-Response/AKA-Challenge message to the 3GPP AAA Server.</td>
</tr>
<tr>
<td>9a.</td>
<td>The AAA checks, if the authentication response is correct.</td>
</tr>
<tr>
<td>9b.</td>
<td>When all checks are successful, the 3GPP AAA Server sends the final Authentication and Authorization Answer (with a result code indicating success) including the relevant service authorization information, an EAP success and the key material to the ePDG. This key material shall consist of the MSK generated during the authentication process. When the SWm and Swd interfaces between ePDG and 3GPP AAA Server are implemented using Diameter, the MSK shall be encapsulated in the EAP-Master-Session-Key-AVP, as defined in RFC 4072.</td>
</tr>
<tr>
<td>10.</td>
<td>The MSK shall be used by the ePDG to generate the AUTH parameters in order to authenticate the IKE_SA_INIT phase messages, as specified for IKEv2 in RFC 4306. These two first messages had not been authenticated before as there was no key material available yet. According to RFC 4306 [3], the shared secret generated in an EAP exchange (the MSK), when used over IKEv2, shall be used to generated the AUTH parameters.</td>
</tr>
<tr>
<td>11.</td>
<td>The EAP Success/Failure message is forwarded to the UE over IKEv2.</td>
</tr>
<tr>
<td>12.</td>
<td>UE -&gt; ePDG: IKEv2 AUTH_REQUEST - The UE sends Auth_Request (IDi, [CERT]</td>
</tr>
</tbody>
</table>
### Evolved Packet Data Gateway Overview

#### How the ePDG Works

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 13. | **ePDG -> P-GW: Create Session Request** - The ePDG sends Create Session Request (IMSI, Serving Network, RAT Type (WLAN), Indication Flags (handover=1, DAB=IPv4v6), Sender F-TEID for C-plane, APN, Selection Mode, PAA, APN-AMBR, Bearer Contexts) to the P-GW. Selection Mode shall be set to “MS or network provided APN, subscribed verified”.
| 16a. | **P-GW -> ePDG: Create Session Response** - The P-GW allocates the requested IP address session and responds back to the ePDG with a Create Session Response (Cause, P-GW S2b Address C-plane, PAA, Bearer Contexts Created, APN-AMBR, Recovery, Additional Protocol Configuration Options (APCO), Private Extension) message.
| 16b. | **P-GW -> ePDG: Create Bearer Request** - If there are PCC rules that require a dedicated bearer, the P-GW sends Create Bearer Request (LBI, Bearer Contexts (EPS Bearer ID, TFT, S2b-U PGW F-TEID, Bearer Level QoS)) to the ePDG. Note that Charging ID is not sent on S2b.
| 16c. | The ePDG sends Create Bearer Response (Cause, Bearer Context (EPS Bearer ID, Cause, S2b-U ePDG F-TEID, S2b-U PGW F-TEID), [Recovery]) message.
| 17. | **ePDG -> UE: IKE_AUTH** - The ePDG calculates the AUTH parameter, which authenticates the second IKE_SA_INIT message. The ePDG sends the assigned IP address in the configuration payload (CFG_REPLY). The AUTH parameter is sent to the UE together with the configuration payload, security associations and the rest of the IKEv2 parameters and the IKEv2 negotiation terminates. |

---

**ePDG Administration Guide, StarOS Release 18**
Supported Standards

The ePDG service complies with the following standards:

- **3GPP References**
- **IETF References**

### 3GPP References

- 3GPP TS 23.234-b.0.0: “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3GPP system to Wireless Local Area Network (WLAN) interworking; System description (Release 11)”.
- 3GPP TS 24.301-b.7.0: “3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS)”.
- 3GPP TS 23.402-b.7.0: “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Architecture enhancements for non-3GPP accesses (Release 9)”.
- 3GPP TS 24.302-b.7.0: “3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Access to the 3GPP Evolved Packet Core (EPC) via non-3GPP access networks; Stage 3 (Release 8)”.
- 3GPP TS 29.273-b.6.0: “3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Evolved Packet System (EPS); 3GPP EPS AAA interfaces (Release 9)”.
- 3GPP TS 29.274-b.7.0: “3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; 3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3 (Release 11) (b.7.0 (June 2013))”.
- 3GPP TS 29.275-a.2.0: “3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Proxy Mobile IPv6 (PMIPv6) based Mobility and Tunnelling protocols; Stage 3 (Release 8)”.
- 3GPP TS 29.303-b.2.0 Generation Partnership Project; Technical Specification Group Core Network and Terminals; Domain Name System Procedures; Stage 3(Release 11).
- 3GPP TS 33.234-b.4.0: “3rd Generation Partnership Project; Technical Specification Group Service and System Aspects; 3G Security; Wireless Local Area Network (WLAN) Interworking Security; (Release 6)”.
- 3GPP TS 33.402-b.4.0: “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3GPP System Architecture Evolution (SAE); Security aspects of non-3GPP accesses; (Release 8).”

### IETF References

- RFC 3588 (September 2003): “Diameter Base Protocol”.
- RFC 3602 (September 2003): The AES-CBC Cipher Algorithm and Its Use with IPsec”.

---

*ePDG Administration Guide, StarOS Release 18*
- RFC 3715 (March 2004): “IPsec-Network Address Translation (NAT) Compatibility Requirements”.
- RFC 3748 (June 2004): “Extensible Authentication Protocol (EAP)”.
- RFC 3775 (June 2004): “Mobility Support in IPv6”.
- RFC 3948 (January 2005): “UDP Encapsulation of IPsec ESP Packets”.
- RFC 4303 (December 2005): “IP Encapsulating Security Payload (ESP)”.
- RFC 4306 (December 2005): “Internet Key Exchange (IKEv2) Protocol”.
- RFC 4739 (November 2006): “Multiple Authentication Exchanges in the Internet Key Exchange (IKEv2) Protocol”.
- RFC 5845 (June 2010): “Generic Routing Encapsulation (GRE) Key Option for Proxy Mobile IPv6”.
- RFC 5846 (June 2010): “Binding Revocation for IPv6 Mobility”.
- RFC 5996 (September 2010): “Internet Key Exchange Protocol Version 2 (IKEv2)”.
This chapter provides configuration instructions for the ePDG (evolved Packet Data Gateway).

**Important:** Information about the commands in this chapter can be found in the *eHRPD/LTE 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.

The following section is included in this chapter:

- Configuring the System to Perform as an Evolved Packet Data Gateway
Configuring the System to Perform as an Evolved Packet Data Gateway

This section provides a high-level series of steps and the associated configuration file examples for configuring the system to perform as an ePDG in a test environment. For a configuration example without instructions, see “Sample Evolved Packet Data Gateway Configuration File”.

Information provided in this section includes the following:

- Required Information
- Evolved Packet Data Gateway Configuration

Required Information

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

Required Local Context Configuration Information

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

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management Interface Configuration</td>
<td></td>
</tr>
<tr>
<td>Interface name(s)</td>
<td>The name(s) of the management interface(s), which can be from 1 to 79 alpha and/or numeric characters. Multiple names are needed if multiple interfaces will be configured.</td>
</tr>
<tr>
<td>IP address(es) and subnet mask(s)</td>
<td>The IPv4 address(es) and subnet mask(s) assigned to the interface(s). Multiple addresses and subnet masks are needed if multiple interfaces will be configured.</td>
</tr>
<tr>
<td>Remote access type(s)</td>
<td>The type(s) of remote access that will be used to access the system, such as ftpd, sshd, and/or telnetd.</td>
</tr>
<tr>
<td>Security administrator name(s)</td>
<td>The name(s) of the security administrator(s) with full rights to the system.</td>
</tr>
<tr>
<td>Security administrator password(s)</td>
<td>Open or encrypted passwords can be used.</td>
</tr>
<tr>
<td>Gateway IP address(es)</td>
<td>Used when configuring static IP routes from the management interface(s) to a specific network.</td>
</tr>
</tbody>
</table>
Required Information for ePDG Context and Service Configuration

The following table lists the information that is required to configure the ePDG context and service on the ePDG.

Table 21. Required Information for ePDG Context and Service Configuration

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ePDG Context Configuration</td>
<td></td>
</tr>
<tr>
<td>ePDG context name</td>
<td>The name of the ePDG context, which can be from 1 to 79 alpha and/or numeric characters.</td>
</tr>
<tr>
<td>EAP profile name(s)</td>
<td>The name(s) of the EAP profile(s) to be used for UE authentication via the EAP authentication method.</td>
</tr>
<tr>
<td>IPSec transform set name(s)</td>
<td>The name(s) of the IPSec transform set(s) to be used by the ePDG service.</td>
</tr>
<tr>
<td>IKEv2 transform set name(s)</td>
<td>The name(s) of the IKEv2 transform set(s) to be used by the ePDG service.</td>
</tr>
<tr>
<td>Crypto template name(s)</td>
<td>The name(s) of the IKEv2 crypto template(s) to be used by the ePDG service.</td>
</tr>
<tr>
<td>Configuration for the SWu, SWm, and DNS Interfaces, and the SWu and SWm Loopback Interfaces</td>
<td></td>
</tr>
<tr>
<td>SWu interface name</td>
<td>The name of the SWu interface, which can be from 1 to 79 alpha and/or numeric characters. This is the interface that carries the IPSec tunnels between the WLAN UEs and the ePDG.</td>
</tr>
<tr>
<td>SWm interface name</td>
<td>The name of the SWm interface, which can be from 1 to 79 alpha and/or numeric characters. This is the interface between the ePDG and the external 3GPP AAA server.</td>
</tr>
<tr>
<td>DNS interface name</td>
<td>The name of the DNS interface, which can be from 1 to 79 alpha and/or numeric characters. This is the interface between the ePDG and the external DNS.</td>
</tr>
<tr>
<td>SWu loopback interface name</td>
<td>The name of the SWu loopback interface, which can be from 1 to 79 alpha and/or numeric characters.</td>
</tr>
<tr>
<td>SWm loopback interface name</td>
<td>The name of the SWm loopback interface, which can be from 1 to 79 alpha and/or numeric characters.</td>
</tr>
<tr>
<td>IP addresses and subnet masks</td>
<td>The IP addresses assigned to the SWu (IPv4), SWm (either IPv4 or IPv6), and DNS interfaces (either IPv4 or IPv6), and to the SWu (IPv4) and SWm (either IPv4 or IPv6) loopback interfaces.</td>
</tr>
<tr>
<td>Physical Ethernet port numbers</td>
<td>The physical Ethernet ports to which the SWu, DNS, and SWm interfaces will be bound. Ports are identified by the chassis slot number where the line card resides, followed by the number of the physical connectors on the card. For example, port 19/1 identifies connector number 1 on the card in slot 19. A single physical port can facilitate multiple interfaces.</td>
</tr>
</tbody>
</table>
### Required Information

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AAA Group Configuration</strong></td>
</tr>
<tr>
<td>The name of the Diameter dictionary used for authentication.</td>
</tr>
<tr>
<td><strong>Diameter endpoint name</strong></td>
</tr>
<tr>
<td>The name of the Diameter endpoint, which can be from 1 to 63 alpha and/or</td>
</tr>
<tr>
<td>numeric characters. This is the name of the external 3GPP AAA server using</td>
</tr>
<tr>
<td>the SWm interface.</td>
</tr>
<tr>
<td><strong>ePDG Service Configuration</strong></td>
</tr>
<tr>
<td>The name of the ePDG service, which can be from 1 to 63 alpha and/or numeric</td>
</tr>
<tr>
<td>characters.</td>
</tr>
<tr>
<td><strong>PLMN ID (Public Land Mobile Network Identifier)</strong></td>
</tr>
<tr>
<td>The MCC (Mobile Country Code) and MNC (Mobile Network Code) for the ePDG.</td>
</tr>
<tr>
<td><strong>Egress context name</strong></td>
</tr>
<tr>
<td>The name of the Egress context, which can be from 1 to 79 alpha and/or numeric characters.</td>
</tr>
<tr>
<td><strong>MAG service name</strong></td>
</tr>
<tr>
<td>The name of the MAG (Mobile Access Gateway) service on the ePDG, which can be from 1 to 63 alpha and/or numeric characters.</td>
</tr>
<tr>
<td><strong>EGTP service name</strong></td>
</tr>
<tr>
<td>The name of the EGTP service associated with ePDG, which can be from 1 to 63 alpha and/or numeric characters.</td>
</tr>
<tr>
<td><strong>ePDG FQDN</strong></td>
</tr>
<tr>
<td>The ePDG FQDN (Fully Qualified Domain Name), used for longest suffix matching during P-GW dynamic allocation. The ePDG FQDN can be from 1 to 256 alpha and/or numeric characters.</td>
</tr>
<tr>
<td><strong>Diameter endpoint name</strong></td>
</tr>
<tr>
<td>The name of the Diameter endpoint, which can be from 1 to 63 alpha and/or numeric characters. This is the name of the external 3GPP AAA server using the SWm interface.</td>
</tr>
<tr>
<td><strong>Origin host</strong></td>
</tr>
<tr>
<td>The name of the Diameter origin host, which can be from 1 to 255 alpha and/or numeric characters.</td>
</tr>
<tr>
<td><strong>Origin host address</strong></td>
</tr>
<tr>
<td>The IPv6 address of the Diameter origin host.</td>
</tr>
<tr>
<td><strong>Peer name</strong></td>
</tr>
<tr>
<td>The name of the Diameter endpoint, which can be from 1 to 63 alpha and/or numeric characters. This is the name of the external 3GPP AAA server using the SWm interface.</td>
</tr>
<tr>
<td><strong>Peer realm name</strong></td>
</tr>
<tr>
<td>The name of the peer realm, which can be from 1 to 127 alpha and/or numeric 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.</td>
</tr>
<tr>
<td><strong>Peer address</strong></td>
</tr>
<tr>
<td>The IPv4 or IPv6 address of the Diameter endpoint.</td>
</tr>
<tr>
<td><strong>DNS client name</strong></td>
</tr>
<tr>
<td>The name of the DNS client on the ePDG, which can be from 1 to 63 alpha and/or numeric characters.</td>
</tr>
<tr>
<td><strong>DNS address</strong></td>
</tr>
<tr>
<td>The IPv4 or IPv6 address of the local DNS client.</td>
</tr>
</tbody>
</table>

### Required Information for Egress Context and MAG Service Configuration

The following table lists the information that is required to configure the Egress context and MAG (Mobile Access Gateway) service on the ePDG.
Table 22. Required Information for Egress Context and MAG Service Configuration

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Egress context name</td>
<td>The name of the Egress context, which can be from 1 to 79 alpha and/or numeric characters.</td>
</tr>
<tr>
<td></td>
<td><strong>S2b Interface Configuration</strong></td>
</tr>
<tr>
<td>S2b interface name</td>
<td>The name of the S2b interface, which can be from 1 to 79 alpha and/or numeric characters. This is the interface that carries the PMIPv6 signaling between the MAG (Mobile Access Gateway) function on the ePDG and the LMA (Local Mobility Anchor) function on the P-GW.</td>
</tr>
<tr>
<td>MIPv6 address and subnet mask</td>
<td>The MIPv6 address and subnet mask assigned to the S2b interface.</td>
</tr>
<tr>
<td>S2b loopback interface name</td>
<td>The name of the S2b loopback interface, which can be from 1 to 79 alpha and/or numeric characters.</td>
</tr>
<tr>
<td>MIPv6 address and subnet mask</td>
<td>The MIPv6 address and subnet mask assigned to the S2b loopback interface.</td>
</tr>
<tr>
<td>Gateway IPv6 address</td>
<td>The gateway IP address for configuring the IPv6 route from the S2b interface to the P-GW.</td>
</tr>
<tr>
<td></td>
<td><strong>MAG Service Configuration</strong></td>
</tr>
<tr>
<td>MAG service name</td>
<td>The name of the MAG (Mobile Access Gateway) service, which can be from 1 to 63 alpha and/or numeric characters.</td>
</tr>
<tr>
<td>Physical Ethernet port numbers</td>
<td>The physical Ethernet ports to which the SWu, DNS, SWm, and S2b interfaces will be bound. Ports are identified by the chassis slot number where the line card resides, followed by the number of the physical connectors on the card. For example, port 24/1 identifies connector number 1 on the card in slot 24. A single physical port can facilitate multiple interfaces.</td>
</tr>
</tbody>
</table>

**Required Information for Egress Context and EGTP Service Configuration**

The following table lists the information that is required to configure the Egress context and EGTP (Evolved GPRS Tunneling Protocol) service on the ePDG.

Table 23. Required Information for Egress Context and EGTP Service Configuration

<table>
<thead>
<tr>
<th>Required Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Egress context name</td>
<td>The name of the Egress context, which can be from 1 to 79 alpha and/or numeric characters.</td>
</tr>
<tr>
<td></td>
<td><strong>S2b Interface Configuration</strong></td>
</tr>
<tr>
<td>S2b interface name</td>
<td>The name of the S2b interface, which can be from 1 to 79 alpha and/or numeric characters. This is the interface that carries the GTPv2 Signaling and data messages between ePDG and PGW.</td>
</tr>
</tbody>
</table>
### Required Information

<table>
<thead>
<tr>
<th>Description</th>
<th>S2b loopback interface name</th>
<th>Gateway IPv6 address</th>
</tr>
</thead>
<tbody>
<tr>
<td>The name of the S2b loopback interface, which can be from 1 to 79 alpha and/or numeric characters.</td>
<td></td>
<td>The gateway IP address for configuring from the S2b interface to the P-GW.</td>
</tr>
</tbody>
</table>

### eGTP Service Configuration

<table>
<thead>
<tr>
<th>Description</th>
<th>GTPU service name</th>
<th>egtp-service name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use GTPU service name to allow configuration of GTPU Service. Use the bind configuration to bind the s2b loopback address. This will be used for data plane of GTPv2.</td>
<td></td>
<td>Use EGTP service name to allow configuration of eGTP service. Use the bind configuration to bind the s2b loopback address for gtpc and also use the association cli to associate the gtpu-service name.</td>
</tr>
</tbody>
</table>

### Evolved Packet Data Gateway Configuration

The figure below shows the contexts in which ePDG configuration occurs. The steps that follow the figure explain the high-level ePDG configuration steps.

#### Step 1
Set system configuration parameters such as activating PSC2s, enabling Diameter Proxy mode, and enabling session recovery by following the configuration examples in the System Administration Guide.

#### Step 2
Set initial configuration parameters in the local context by following the configuration example in the section Initial Configuration.
Step 3 Configure the ePDG context, the EAP profile, the IPSec and IKEv2 transform sets, the crypto template, the SWu, SWm, and DNS interfaces, the SWu and SWm loopback interfaces, and the AAA group for Diameter authentication by following the configuration example in the section ePDG Context and Service Configuration.

Step 4 Configure the Egress context and MAG service or Egress context and EGTP by following the configuration example in the section Egress Context and MAG Service Configuration. or Required Information for Egress Context and EGTP Service Configuration

Step 5 Enable ePDG bulk statistics by following the configuration example in the section Bulk Statistics Configuration.

Step 6 Enable system logging activity by following the configuration example in the section Logging Configuration.

Step 7 Save the configuration file.

Initial Configuration

Set local system management parameters by following the configuration example in the section Modifying the Local Context.

Modifying the Local Context

Use the following configuration example to create a management interface, configure remote access capability, and set the default subscriber in the local context:

```
configure
  context local
    interface <mgmt_interface_name>
      ip address <ip_address> <subnet_mask>
    exit
  server ftpd
  ssh key <data> length <octets>
  ssh key <data> length <octets>
  ssh key <data> length <octets>
  server sshd
    subsystem sftpd
    exit
  server telnetd
    exit
  subscriber default
```
The `server` command configures remote server access protocols for the current context. The system automatically creates a default subscriber, a default AAA group, and a default GTTP group whenever a context is created. The `ip route` command in this example creates a default route for the management interface.

### ePDG Context and Service Configuration

**Step 1**  Create the context in which the ePDG service will reside by following the configuration example in the section Creating the ePDG Context.

**Step 2**  Create the ePDG service by following the configuration example in the section Creating the ePDG Service.

### Creating the ePDG Context

Use the following configuration example to create the ePDG context, the EAP profile, the IPSec and IKEv2 transform sets, the crypto template, the SWu, SWm, and DNS interfaces, the SWm and IPSec loopback interfaces, and the AAA group for Diameter authentication:

```
configure
  context <epdg_context_name>
    eap-profile <eap_profile_name>
      mode authenticator-pass-through
    exit
    ipsec transform-set <ipsec_tset_name>
```

hmac aes-xcbc-96
exit
ikev2-ikesa transform-set <ikev2_ikesa_tset_name>
hmac aes-xcbc-96
prf aes-scbc-128
exit
crypto template <crypto_template_name> ikev2-dynamic
  authentication remote eap-profile <eap_profile_name>
  exit
ikev2-ikesa retransmission-timeout <milliseconds>
ikev2-ikesa transform-set list <ikev2_ikesa_tset_name>
ikev2-ikesa rekey
  payload <payload_name> match childsa match any
  ipsec transform-set list <ipsec_tset_name>
    lifetime <seconds>
    rekey keepalive
  exit
ikev2-ikesa keepalive-user-activity
ikev2-ikesa policy error-notification
ikev2-ikesa policy use-rfc5996-notification
exit
ip routing maximum-paths <max_num>
interface <swu_interface_name>
  ip address <ip_address> <subnet_mask>
  exit
interface <swm_interface_name>
  ip address <ip_address> <subnet_mask>
  exit
interface <epdg_dns_interface_name>
ip address <ip_address> <subnet_mask>
exit
interface <swu_loopback_interface_name> loopback
ip address <ip_address> <subnet_mask>
exit
interface <swm_ipsec_loopback_interface_name> loopback
ip address <ip_address> <subnet_mask>
exit
subscriber default
aaa group <group_name>
ip context-name <epdg_context_name>
exit
aaa group default
exit
aaa group <group_name>
diameter authentication dictionary <aaa_custom_dictionary>
diameter authentication endpoint <endpoint_name>
diameter authentication max-retries <max_retries>
diameter authentication max-transmissions <max_transmissions>
diameter authentication request-timeout <request_timeout_duration>
diameter authentication failure-handling eap-request request-timeout action terminate
diameter authentication failure-handling eap-request result-code <start_result_code_1> to <end_result_code_1> action retry-and-terminate
diameter authentication failure-handling eap-request result-code <start_result_code_2> to <end_result_code_2> action terminate
diameter authentication server <host_name> priority <priority>
exit
gtp group default
exit
In this example, the EAP method is used for UE authentication. The `eap-profile` command creates the EAP profile to be used in the crypto template for the ePDG service. The `mode authenticator-pass-through` command specifies that the ePDG functions as an authenticator passthrough device, enabling an external EAP server to perform UE authentication.

The `crypto template` command and associated commands are used to define the cryptographic policy for the ePDG. You must create one crypto template per ePDG service. The `ikev2-dynamic` keyword in the `crypto template` command specifies that IKEv2 protocol is used. The `authentication remote` command specifies the EAP profile to use for authenticating the remote peer.

The `rekey keepalive` command enables Child SA (Security Association) rekeying so that a session will be rekeyed even when there has been no data exchanged since the last rekeying operation. The `ikev2-ikesa keepalive-user-activity` command resets the user inactivity timer when keepalive messages are received from the peer. The `ikev2-ikesa policy error-notification` command enables the ePDG to generate Error Notify messages for Invalid IKEv2 Exchange Message ID and Invalid IKEv2 Exchange Syntax for the IKE_SA_INIT exchange.

The `ip routing maximum-paths` command enables ECMP (Equal Cost Multiple Path) routing support and specifies the maximum number of ECMP paths that can be submitted by a routing protocol in the current context. The `interface` command creates each of the logical interfaces, and the associated `ip address` command specifies the IP address and subnet mask of each interface.

The `aaa group` command configures the AAA server group in the ePDG context and the `diameter authentication` commands specify the associated Diameter authentication settings.

The `ikev2-ikesa policy use-rfc5996-notification` command enables processing for new notification payloads added in RFC 5996, and is disabled by default.

### Creating the ePDG Service

Use the following configuration example to do the following:

- Create the ePDG service.
- Specify the context in which the MAG/EGTP service will reside.
- Specify the ePDG FQDN (Fully Qualified Domain Name) used for longest suffix matching during P-GW dynamic allocation.
- Bind the crypto template to the ePDG service.
- Specify the Diameter origin endpoint and associated settings.
- Specify the name of the DNS client for DNS queries and bind the IP address.

```
configure

custom <epdg_context_name>

epgd-service <epdg_service_name>

plmn id mcc <code> mnc <code>
```

**Important:** When GTPv2 is used instead of mobile-access-gateway configuration, ePDG shall use associate `egtp-service egtp_service_name`.
mobile-access-gateway context <egress_context_name> mag-service <mag_service_name>
setup-timeout <seconds>
fqdn <domain_name>
bind address <ip_address> crypto-template <crypto_template_name>
pgw-selection agent-info error-terminate
dns-pgw selection topology weight
exit
ip route <ip_address/subnet mask> <ip_address/subnet mask> <gateway_ip_address> <mgmt_interface_name>
ip domain-lookup
ip name-servers <ip_address>
diameter endpoint <endpoint_name>
use-proxy
origin host <host_name> address <ip_address> port <port_number>
response-timeout <seconds>
connection timeout <seconds>
cea-timeout <seconds>
dpa-timeout <seconds>
connection retry-timeout <seconds>
peer <peer_name> realm <realm_name> address <ip_address>
route-entry peer <peer_id> weight <priority>
exit
dns-client <dns_client_name>
bind address <ip_address>
exit
end

The ePDG context defaults to a MAG service configured in the same context unless the mobile-access-gateway command is used to specify the context where the MAG service will reside as shown above. The fqdn command configures the ePDG FQDN (Fully Qualified Domain Name) for longest suffix match during P-GW dynamic allocation. The IP address that you to the ePDG service above is used as the connection point for establishing the IKEv2 sessions.
between the WLAN UEs and the ePDG. The `pgw-selection agent-info error-terminate` command specifies the action to be taken during P-GW selection when the MIP6-agent-info parameter is expected but not received from the AAA server/HSS, which is to terminate P-GW selection and reject the call. The `dns-pgw selection topology weight` command enables P-GW load balancing based on both topology, in which the nearest P-GW to the subscriber is selected first, and weight, in which the P-GW is select based on a weighted average.

The `ip route` command in this example creates a route for the SWu interface between the WLAN UEs and the ePDG and specifies the destination IP addresses that will use this route. The `ip domain-lookup` command enables domain name lookup via DNS for the current context. The `ip name-servers` command specifies the IP address of the DNS that the ePDG context will use for logical host name resolution. The `diameter endpoint` command specifies the Diameter origin endpoint.

The `origin host` command specifies the origin host for the Diameter endpoint. The `peer` command specifies a peer address for the Diameter endpoint. The `route-entry` command creates an entry in the route table for the Diameter peer.

The `dns-client` command specifies the DNS client used during P-GW FQDN discovery.

### Egress Context and MAG Service Configuration

Create the Egress context and the MAG (Mobile Access Gateway) service by following the configuration example in the section Configuring the Egress Context and MAG Service.

### Configuring the Egress Context and MAG Service

Use the following configuration example to configure the Egress context, the MAG (Mobile Access Gateway) service, the S2b interface and S2b loopback interface to the P-GW, and bind all of the logical interfaces to the physical Ethernet ports.

```
configure

context <egress_context_name>

interface <s2b_interface_name>
  ipv6 address <ipv6_address>
  exit

interface <s2b_loopback_interface_name>
  ipv6 address <ipv6_address>
  exit

subscriber default
  exit

aaa group default
  exit

gtpp group default
```
exit
mag-service <mag_service_name>
    reg-lifetime <seconds>
    bind address <ipv6_address>
exit
ipv6 route <ipv6_address/prefix_length> next-hop <ipv6_address> interface <s2b_interface_name>
exit
port ethernet <slot_number/port_number>
    no shutdown
    vlan <tag>
    bind interface <swu_interface_name> <epdg_context_name>
exit
port ethernet <slot_number/port_number>
    no shutdown
    vlan <tag>
    bind interface <epdg_dns_interface_name> <epdg_context_name>
exit
port ethernet <slot_number/port_number>
    no shutdown
    vlan <tag>
    bind interface <swm_interface_name> <epdg_context_name>
exit
port ethernet <slot_number/port_number>
    no shutdown
    vlan <tag>
    bind interface <s2b_interface_name> <egress_context_name>
exit
end
The `mag-service` command creates the MAG (Mobile Access Gateway) service that communicates with the LMA (Local Mobility Anchor) service on the P-GW to provide network-based mobility management. The `ipv6 route` command configures a static IPv6 route to the next-hop router. In this configuration, it configures a static route from the ePDG to the P-GW over the S2b interface. The `bind interface` command binds each logical interface to a physical Ethernet port.

**Egress Context and EGTP Service Configuration**

Create the Egress context and the EGTP (Evolved GPRS Tunnel Protocol) service by following the configuration example in the section Configuring the Egress Context and EGTP Service.

**Configuring the Egress Context and EGTP Service**

Use the following configuration example to configure the egress context, the EGTP (Evolved GPRS Tunnel Protocol) service, the S2b interface and S2b loopback interface to the P-GW, and bind all of the logical interfaces to the physical Ethernet ports.

```
configure

context <egress_context_name>

interface <s2b_interface_name>

ipv4/ipv6 address <ipv6_address>

exit

interface <s2b_loopback_interface_name>

ipv4/ipv6 address <ipv6_address>

exit

subscriber default

exit

aaa group default

exit

gtppp group default

exit

gtpu-service <gtpu-service-name>

reg-lifetime <seconds>

bind ipv4/ipv6-address <s2bloopbackipv4/ipv6_address>

exit
```
egtp-service egtp-epdg-egress
interface-type interface-epdg-egress
associate gtpu-service gtpu-epdg-egress
exit
ipv4/ipv6 route <ipv4/ipv6_address/prefix_length> next-hop <ip4/ipv6_address>
interface <s2b_interface_name>
exit
port ethernet <slot_number/port_number>
no shutdown
vlan <tag>
bind interface <swu_interface_name> <epdg_context_name>
exit
port ethernet <slot_number/port_number>
no shutdown
vlan <tag>
bind interface <epdg_dns_interface_name> <epdg_context_name>
exit
port ethernet <slot_number/port_number>
no shutdown
vlan <tag>
bind interface <swm_interface_name> <epdg_context_name>
exit
port ethernet <slot_number/port_number>
no shutdown
vlan <tag>
bind interface <s2b_interface_name> <egress_context_name>
exit
end

The `egtp-service` command creates the eGTP (evolved GPRS Tunneling Protocol) service that communicates with the LMA (Local Mobility Anchor) service on the P-GW to provide network-based mobility management. The `ipv6`
route command configures a static IPv6 route to the next-hop router. In this configuration, it configures a static route from the ePDG to the P-GW over the S2b interface. The bind interface command binds each logical interface to a physical Ethernet port.

**Bulk Statistics Configuration**

Use the following configuration example to enable ePDG bulk statistics:

```
configure

bulkstats collection
bulkstats mode

sample-interval <time_interval>
transfer-interval <xmit_time_interval>

file <number>

receiver <ip_address> primary mechanism ftp login <username> password <pwd>
receiver <ip_address> secondary mechanism ftp login <username> password <pwd>

epdg schema <file_name> format " txbytes : %txbytes% txpkts : %txpkts% rxbytes : %rxbytes% rxpkts : %rxpkts% sess-txbytes : %sess-txbytes% sess-rxbytes : %sess-rxbytes% sess-txpackets : %sess-txpackets% sess-rxpackets : %sess-rxpackets% eap-rttlsrvrpassthru : %eap-rttlsrvrpassthru% eap-rxsuccessrvrpassthru : %eap-rxsuccessrvrpassthru% num-gtp-bearermodified : %num-gtp-bearermodified% num-gtp-db-active : %num-gtp-db-active% num-gtp-db-released : %num-gtp-db-released% curses-gtp-ipv4 : %curses-gtp-ipv4% curses-gtp-ipv6 : %curses-gtp-ipv6% curses-gtp-ipv4v6 : %curses-gtp-ipv4v6% "

end
```

The `bulkstats collection` command in this example enables bulk statistics, and the system begins collecting predefined bulk statistical information.

The `bulkstats mode` command enters Bulk Statistics Configuration Mode, where you define the statistics to collect.

The `sample-interval` command specifies the time interval, in minutes, to collect the defined statistics. The `<time_interval>` can be in the range of 1 to 1440 minutes. The default value is 15 minutes.

The `transfer-interval` command specifies the time interval, in minutes, to transfer the collected statistics to the receiver (the collection server). The `<xmit_time_interval>` can be in the range of 1 to 999999 minutes. The default value is 480 minutes.

The `file` command specifies a file in which to collect the bulk statistics. A bulk statistics file is used to group bulk statistics schema, delivery options, and receiver configuration. The `<number>` can be in the range of 1 to 4.

The `receiver` command in this example specifies a primary and secondary collection server, the transfer mechanism (in this example, ftp), and a login name and password.

The `epdg schema` command specifies that the epdg schema is used to gather statistics. The `<file_name>` is an arbitrary name (in the range of 1 to 31 characters) to use as a label for the collected statistics defined by the `format` option. The `format` option defines within quotation marks the list of variables in the epdg schema to collect. The format string can be in the range of 1 to 3599.
For descriptions of the epdg schema variables, see “ePDG Schema Statistics” in the Statistics and Counters Reference. For more information on configuring bulk statistics, see the System Administration Guide.

Logging Configuration

Use the following configuration example to enable logging on the ePDG:

```configure
logging filter active facility sessmgr level <critical/error>
logging filter active facility ipsec level <critical/error>
logging filter active facility ikev2 level <critical/error>
logging filter active facility epdg level <critical/error>
logging filter active facility aaamgr level <critical/error>
logging filter active facility diameter level <critical/error>
logging filter active facility egtpc level <critical/error>
logging filter active facility egtpmgr level <critical/error>
logging filter active facility gtpumgr level <critical/error>
logging filter active facility diameter-auth level <critical/error>
logging active
end```

Non UICC device support for certificate and multi authentication configuration

List of authentication methods are defined and associated in Crypto Template. The basic sample configuration required for OCSP and Certificate based authentication is as follows. For backward compatibility, the configuration for auth method inside Crypto Template will be working.

The following are the configuration considerations:

1. At max three sets of authentication methods in list can be associated.
2. Each set has only one local and one remote authentication method configuration.
3. The existing configuration inside the Crypto Template takes precedence over the new auth-method-set defined in case same auth method is configured at both places.

```configure
#CA Certificate for device certificate authentication:
   ca-certificate name <ca-name> pem url file: <ca certificate path>

#ePDG Certificate:
ca-certificate name <epdg-name> pem url file: <epdg certificate path> private-key pem url file:<epdg private key path>
eap-profile <profile name>
    mode authenticator-pass-through
#exit
ikev2-ikesa auth-method-set <list-name-1>
    authentication remote certificate
    authentication local certificate
#exit
ikev2-ikesa auth-method-set <list-name-2>
    authentication eap-profile eap1
#exit
crypto template boston ikev2-subscriber
    ikev2-ikesa auth-method-set list <list-name-2> <list-name-2> certificate <epdg-name>
    ca-certificate list ca-cert-name <ca-name>
#exit

Saving the Configuration

Save the ePDG configuration file 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, see the *System Administration Guide* and the *eHRPD/LTE Command Line Interface Reference*. 
This chapter provides information for monitoring the status and performance of the ePDG (evolved Packet Data Gateway) using the `show` commands found in the CLI (Command Line Interface). These command have many related keywords that allow them to provide useful information on all aspects of the system ranging from current software configuration through call activity and status.

The selection of `show` commands listed in this chapter is intended to provided the most useful and in-depth information for monitoring the system. For additional information on these and other `show` commands and keywords, refer to the `eHRPD/LTE Command Line Interface Reference`.

The system also supports the sending of SNMP (Simple Network Management Protocol) traps that indicate status and alarm conditions. See the `SNMP MIB Reference` for a detailed listing of these traps.
Monitoring ePDG Status and Performance

The following table contains the CLI commands used to monitor the status of the ePDG features and functions. Output descriptions for most of the commands are located in the Statistics and Counters Reference.

Table 24. ePDG Status and Performance Monitoring Commands

<table>
<thead>
<tr>
<th>To do this:</th>
<th>Enter this command:</th>
</tr>
</thead>
<tbody>
<tr>
<td>View ePDG Service Information and Statistics</td>
<td>show epdg-service { all [ counters ]</td>
</tr>
<tr>
<td>View ePDG service information and statistics.</td>
<td>show epdg-service session { all</td>
</tr>
<tr>
<td>View additional session statistics.</td>
<td>show session { disconnect-reasons</td>
</tr>
<tr>
<td>View ePDG bulk statistics.</td>
<td>show bulkstats variables epdg</td>
</tr>
<tr>
<td>View bulk statistics for the system.</td>
<td>show bulkstats data</td>
</tr>
<tr>
<td>View IPSec and IKEv2 Information</td>
<td>show crypto ipsec security-associations [ summary</td>
</tr>
<tr>
<td>View IPSec transform sets.</td>
<td>show crypto ipsec transform-set</td>
</tr>
<tr>
<td>View IKEv2 security associations.</td>
<td>show crypto ikev2-ikesa security-associations [ peer ipv4/ipv6_address</td>
</tr>
<tr>
<td>View IKEv2 transform sets.</td>
<td>show crypto ikev2-ikesa transform-set</td>
</tr>
<tr>
<td>View IKEv2 statistics.</td>
<td>show crypto statistics [ ikev2 ]</td>
</tr>
<tr>
<td>View crypto manager statistics.</td>
<td>show crypto managers [ crypto-map crypto_map_name</td>
</tr>
<tr>
<td>View Diameter AAA Server Information</td>
<td>show diameter aaa-statistics all</td>
</tr>
<tr>
<td>View Diameter AAA server statistics.</td>
<td>show diameter message-queue counters { inbound</td>
</tr>
<tr>
<td>View Diameter message queue counters.</td>
<td>show diameter statistics</td>
</tr>
<tr>
<td>View Congestion Control Information</td>
<td></td>
</tr>
</tbody>
</table>
To do this: | Enter this command:
---|---
View congestion control statistics. | `show congestion-control statistics ipsecmgr`

View Subscriber Information

View Subscriber Configuration Information

View locally configured subscriber profile settings (must be in the context where the subscriber resides). | `show subscribers configuration username subscriber_name`

View remotely configured subscriber profile settings. | `show subscribers aaa-configuration username subscriber_name`

View subscriber information based on IPv6 address. | `show subscribers ipv6-address ipv6_address`

View subscriber information based on IPv6 address prefix. | `show subscribers ipv6-prefix prefix`

View subscriber information based on caller ID. | `show subscribers callid call_id`

View subscriber information based on username. | `show subscribers username name`

View information for troubleshooting subscriber sessions. | `show subscribers debug-info`

View a summary of subscriber information. | `show subscribers summary`

View Subscribers Currently Accessing the System

View a list of subscribers currently accessing the system. | `show subscribers all`
To do this: | Enter this command:
--- | ---
View a list of ePDG subscribers currently accessing the system. | `show subscribers epdg-only [ [ all ] | [ callid call_id ] | [ card-num card_num ] | [ configured-idle-timeout { 0..4294967295 | < idle_timeout | > idle_timeout | greater-than idle_timeout | less-than idle_timeout } ] | [ connected-time { 0..4294967295 | < connected_time | > connected_time | greater-than connected_time | less-than connected_time } ] | [ counters ] | [ data-rate ] | [ full ] | [ gtp-version version ] | [ gtpu-bind-address ip_address ] | [ gtpu-service service_name ] | [ idle-time { 0..4294967295 | < idle_time | > idle_time | greater-than idle_time | less-than idle_time } ] | [ ip-address { < ipv4_address | > ipv4_address | IPv4 | greater-than ipv4_address | less-than ipv4_address } ] | [ ipv6-prefix ipv6_address/len_format ] | [ long-duration-time-left { 0..4294967295 | < long_dur_time | > long_dur_time | greater-than long_dur_time | less-than long_dur_time } ] | [ network-type { gre | ipip | ipsec | ipv4 | ipv4-pmipv6 | ipv4v6 | ipv4v6-pmipv6 | ipv6 | ipv6-pmipv6 | l2tp | mobile-ip | proxy-mobile-ip } ] | [ qci qci ] | [ rx-data { 0..18446744073709551615 | < rx_bytes | > rx_bytes | greater-than rx_bytes | less-than rx_bytes } ] | [ session-time-left { 0..4294967295 | < sess_time_left | > sess_time_left | greater-than sess_time_left | less-than sess_time_left } ] | [ smgr-instance smgr_instance ] | [ summary ] | [ tft ] | [ tx-data { 0..18446744073709551615 | < tx_bytes | > tx_bytes | greater-than tx_bytes | less-than tx_bytes } ] | [ username ] | [ [ grep grep_options | more ] ] |
### To do this:

<table>
<thead>
<tr>
<th>View a list of ePDG subscribers currently accessing the system per ePDG service.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enter this command:</strong></td>
</tr>
<tr>
<td>`show subscribers  epdg-service service_name [ [ all ]</td>
</tr>
<tr>
<td><strong>Enter this command:</strong></td>
</tr>
<tr>
<td><code>show subscribers full username subscriber_name</code></td>
</tr>
<tr>
<td>View statistics for subscribers using a MAG service on the system.</td>
</tr>
<tr>
<td><strong>Enter this command:</strong></td>
</tr>
<tr>
<td>`show subscribers mag-only [ all</td>
</tr>
<tr>
<td>View statistics for subscribers using a MAG service per MAG service.</td>
</tr>
<tr>
<td><strong>Enter this command:</strong></td>
</tr>
<tr>
<td><code>show subscribers mag-service service_name</code></td>
</tr>
</tbody>
</table>

**View Session Subsystem and Task Information**

**View Session Subsystem Statistics**

**Important:** Refer to the System Administration Guide for additional information on the Session subsystem and its various manager tasks.

<p>| View AAA Manager statistics. |
| <strong>Enter this command:</strong> |
| <code>show session subsystem facility aaamgr all</code> |
| View AAA Proxy statistics. |
| <strong>Enter this command:</strong> |
| <code>show session subsystem facility aaaproxy all</code> |</p>
<table>
<thead>
<tr>
<th>To do this:</th>
<th>Enter this command:</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Session Manager statistics.</td>
<td><code>show session subsystem facility sessmgr all</code></td>
</tr>
<tr>
<td>View MAG Manager statistics.</td>
<td><code>show session subsystem facility magmgr all</code></td>
</tr>
<tr>
<td>View session progress information for the ePDG service.</td>
<td><code>show session progress epdg-service service_name</code></td>
</tr>
<tr>
<td>View session duration information for the ePDG service.</td>
<td><code>show session duration epdg-service service_name</code></td>
</tr>
<tr>
<td>View Task Statistics</td>
<td></td>
</tr>
<tr>
<td>View resource allocation and usage information for Session Manager.</td>
<td><code>show task resources facility sessmgr all</code></td>
</tr>
<tr>
<td>View resource allocation and usage information for IPSec Manager.</td>
<td><code>show task resources facility ipsecmgr all</code></td>
</tr>
<tr>
<td>View Session Resource Status</td>
<td></td>
</tr>
<tr>
<td>View session resource status.</td>
<td><code>show resources session</code></td>
</tr>
<tr>
<td>View Session Recovery Status</td>
<td></td>
</tr>
<tr>
<td>View session recovery status.</td>
<td><code>show session recovery status [ verbose ]</code></td>
</tr>
<tr>
<td>View Session Disconnect Reasons</td>
<td></td>
</tr>
<tr>
<td>View session disconnect reasons.</td>
<td><code>show session disconnect-reasons</code></td>
</tr>
<tr>
<td>View GTPU Tunnels Information</td>
<td></td>
</tr>
<tr>
<td>View GTPU tunnels information</td>
<td><code>show gtpu statistics</code></td>
</tr>
<tr>
<td>View GTP Session Information Like Control Plane TEIDs</td>
<td><code>show egtp sessions</code></td>
</tr>
<tr>
<td>View Subscriber TFT</td>
<td></td>
</tr>
<tr>
<td>View subscriber TFT</td>
<td><code>show subscriber tft</code></td>
</tr>
<tr>
<td>View GTP Messages Information</td>
<td></td>
</tr>
<tr>
<td>View GTP messages information</td>
<td><code>show egtpc statistics</code></td>
</tr>
<tr>
<td>Chassis ICSR Status and monitoring</td>
<td></td>
</tr>
<tr>
<td>To do this:</td>
<td>Enter this command:</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>View SRP Information</td>
<td><code>show srp info</code></td>
</tr>
<tr>
<td>View SRP checkpoint</td>
<td><code>show srp checkpoint statistics</code></td>
</tr>
<tr>
<td>Statistics</td>
<td></td>
</tr>
</tbody>
</table>
Clearing Statistics and Counters

It may be necessary to periodically clear statistics and counters in order to gather new information. The system provides the ability to clear statistics and counters based on their grouping.

Statistics and counters can be cleared using the CLI `clear` command. You can also use specific command options such as `clear epdg-service statistics dns-stats`. Refer to the eHRPD/LTE Command Line Interface Reference for detailed information on using this command.
Chapter 4
Evolved Packet Data Gateway Engineering Rules

This appendix provides ePDG (evolved Packet Data Gateway) engineering rules or guidelines that must be considered prior to configuring the system for your network deployment.

The following rules are covered in this appendix:
IKEv2/IPSec Restrictions

The following is a list of known restrictions for IKEv2 and IPSec:

- IKEv2 as per RFC 5996 is supported. IKEv1 is not supported.
- MOBIKE is not supported.
- Only one Child SA is supported.
- Each ePDG service must specify one crypto template.
- IKEv2 multiple authentication and fast re-authentication are not supported.
- Per RFC 4306 and RFC 4718, the following known restrictions apply with respect to the payload and its order. Violations result in INVALID_SYNTAX being returned which is being enabled or disabled through a configuration CLI.
  - While RFC 4306 Section 2.19 specifies that the “CP payload MUST be inserted before the SA payload,” the ePDG does not force strict ordering of this. The ePDG processes these payloads as long as the UE sends a CP payload anywhere inside the encryption data.
  - While RFC 4306 Section 2.23 specifies “The location of the payloads (Notify payloads of type NAT_DETECTION_SOURCE_IP and NAT_DETECTION_DESTINATION_IP) in the IKE_SA_INIT packets are just after the Ni and Nr payloads (before the optional CERTREQ payload),” the ePDG does not force strict ordering of this and still can process these NOTIFY payloads.
- ePDG egress processing will ensure that payloads are in order.
- As described above, when the ePDG receives IKEv2 messages, the ePDG does not enforce the payloads to be in order. However, when the ePDG sends the response or generates any IKEv2 messages, the ePDG will ensure that payloads are ordered according to RFC 4306.
- Traffic selector payloads from the UE support only traffic selectors by IP address range. In other words, the IP protocol ID must be 0. The start port must be 0 and the end port must be 65535. IP address range specification in the TSr payload is not supported.
- Only IKE and ESP protocol IDs are supported. AH is not supported.
- The IKE Protocol ID specification may not use the NONE algorithm for authentication or the ENCR_NULL algorithm for encryption as specified in Section 5 (Security Considerations) of RFC 4306.
- In ESP, ENCR_NULL encryption and NONE authentication cannot be simultaneously used.
- No more than 16 transform types may be present in a single IKE_SA_INIT or IKE_AUTH Request message. If a deviation from this format is used in the proposal format, the ePDG returns an error of INVALID_SYNTAX.
X.509 Certificate (CERT) Restrictions

The following are known restrictions for the creation and use of X.509 CERT:

- The maximum size of a CERT configuration is 4096 bytes.
- The ePDG includes the CERT payload only in the first IKE_AUTH Response for the first authentication.
- If the ePDG receives the CERT-REQ payload when it is not configured to use certificate authentication and if the CRITICAL bit is set in the IKE_AUTH request, the ePDG will reject the exchange. If the ePDG receives the CERT-REQ payload when it is not configured to use certificate authentication and if the CRITICAL bit is not set, the ePDG ignores the payload and proceeds with the exchange to be authenticated using EAP.
- Only a single CERT payload is supported. While RFC 4306 mandates the support of up to four certificates, the ePDG service will support only one X.509 certificate per context. This is due to the size of an X.509 certificate. Inclusion of multiple certificates in a single IKE_AUTH may result in the IKE_AUTH message not being properly transmitted.
GTPv2 Restrictions

The following are known restrictions for the creation and use of GTPv2:

- The ePDG should not send the Modify Bearer Command towards PGW for the modification of QoS or APN AMBR for the default bearer when triggered by HSS to ePDG and then to PGW.
- The ePDG should not send Delete PDN connection set request message per 23.007 for the FQ-CSID failure.
- The ePDG does not support allowing the UE to have more than one PDN connection with one APN.
- The ePDG should not allow the UE to have more than one PDN connection with one APN.
- The ePDG should not handle the delete PDN connection set request received from PGW, basically terminating all the sessions corresponding to the PGW FQ-CSID present in "delete PDN connection set request" message.
- The ePDG should not be allowed to send "Trace Activation/Deactivation" message to PGW for subscriber tracing when same is notified to ePDG on the SWm interface with presence of "Trace Information" AVP.
- The ePDG should not do any policy (QoS) enforcement, ePDG should only be doing the UL traffic QCI to DSCP mapping and marking. Downlink traffic marking shall be done at PGW and ePDG should not handle DSCP for same including the pass through mode marking. ePDG should be be communicating the static QoS profile received from the AAA to the PGW.
- The ePDG does not have CAC/Admission control functionality.
- The ePDG does not supports handling the piggy backed message as specified by 3GPP. ePDG does expects the separate create bearer request message post handling of create session request and response for the creation of dedicated bearer.
S2b Interface Rules

This section describes the engineering rules for the S2b interface for communications between the MAG (Mobility Access Gateway) service residing on the ePDG and the LMA (Local Mobility Anchor) service residing on the P-GW.

MAG-to-LMA Rules

The following engineering rules apply to the S2b interface from the MAG service residing on the ePDG to the LMA service residing on the P-GW:

- An S2b interface is created once the IP address of a logical interface is bound to a MAG service.
- The logical interface(s) that will be used to facilitate the S2b interface(s) must be configured within the Egress context.
- MAG services must be configured within the Egress context.
- MAG services must be associated with an ePDG service.
- Depending on the services offered to the subscriber, the number of sessions facilitated by the S2b interface can be limited.
- Only the IPv6 transport mechanism is supported between the MAG service and the LMA service.
- Node alive is not supported between the MAG service and the LMA service.

EGTP Service Rules

The following engineering rules apply to the S2b interface from the EGTP service residing on the ePDG:

- First GTPU service is defined and then eGTP service is defined with association of previously defined GTPU service and later on the eGTP service is associated with the ePDG service residing in same or different context.
- An S2b interface is created once the IP address of a logical interface is bound to a eGTP and GTPU service.
- The eGTP and GTPU services must be configured within same egress context.
- The eGTP service must be associated with an ePDG service.
- **no gtpc path-failure detection-policy** CLI must be configured under eGTP service to avoid path failure detection action. When this configuration is used the ePDG does not cleans up session if the retransmission timeout has happened for the echo request sent by ePDG.
ePDG Service Rules

The following engineering rule applies to services configured within the system:

- A maximum of 256 services (regardless of type) can be configured per system.
ePDG Subscriber Rules

The following engineering rule applies to subscribers configured within the system:

- Default subscriber templates must be configured per ePDG service.
Chapter 5
IKEv2 Error Codes and Notifications

This appendix lists the IKEv2 error codes and notifications supported by the ePDG (evolved Packet Data Gateway). The following table lists the IKEv2 error codes generated by the ePDG.

Table 25. IKEv2 Error Codes Generated by the ePDG

<table>
<thead>
<tr>
<th>Value</th>
<th>Error Code</th>
<th>ePDG Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UNSUPPORTED_CRITICAL_PAYLOAD</td>
<td>The ePDG sends this code if the Critical Bit exists in the received message and the Payload Type is unrecognized.</td>
</tr>
<tr>
<td>4</td>
<td>INVALID_IKE_SPI</td>
<td>The ePDG does not send this code. The ePDG ignores messages with an unrecognized SPI in order to minimize the impact of DoS attacks.</td>
</tr>
<tr>
<td>5</td>
<td>INVALID_MAJOR_VERSION</td>
<td>The ePDG sends this code in response to messages with an invalid Major Version. The ePDG supports a CLI command to suppress sending this error notification in response to IKE_SA_INIT Request messages. This is done in order to avoid DoS attacks.</td>
</tr>
<tr>
<td>7</td>
<td>INVALID_SYNTAX</td>
<td>The ePDG sends this code upon receiving messages with an inappropriate format, or when necessary payloads are missing. The ePDG does not send this code during IKE_SA_INIT exchanges for an unknown IKE SA. The ePDG sends this code for non-IKEv2 INIT exchanges only (such as IKE_AUTH, CREATE_CHILD_SA, or INFORMATIONAL exchanges). The ePDG also supports a CLI command to suppress sending this error notification. This is done in order to avoid DoS attacks.</td>
</tr>
<tr>
<td>9</td>
<td>INVALID_MESSAGE_ID</td>
<td>The ePDG sends this code in INFORMATIONAL Request messages only. The ePDG also supports a CLI command to suppress sending this error notification in response to IKE_SA_INIT Request messages. This is done in order to avoid DoS attacks.</td>
</tr>
<tr>
<td>11</td>
<td>INVALID_SPI</td>
<td>The ePDG does not send this code. The ePDG ignores ESP packets with an unrecognized SPI in order to minimize the impact by DoS attacks.</td>
</tr>
<tr>
<td>14</td>
<td>NO_PROPOSAL_CHOSEN</td>
<td>The ePDG sends this code when it cannot not choose a proposal from the UE. The ePDG supports a CLI command to suppress sending this code.</td>
</tr>
<tr>
<td>17</td>
<td>INVALID_IKE_PAYLOAD</td>
<td>The ePDG sends this code when the IKE payload from the UE is invalid.</td>
</tr>
<tr>
<td>24</td>
<td>AUTHENTICATION_FAILED</td>
<td>The ePDG sends this code during the EAP authentication when EAP authentication fails.</td>
</tr>
<tr>
<td>35</td>
<td>NO_ADDITIONAL_SAS</td>
<td>The ePDG sends this code when a CREATE_CHILD_SA Request message is unacceptable because the ePDG is unwilling to accept any more CHILD SAs on the IKE_SA.</td>
</tr>
</tbody>
</table>
The following table lists the IKEv2 error codes expected by the ePDG from the WLAN UEs.

Table 26. IKEv2 Error Codes Expected by the ePDG

<table>
<thead>
<tr>
<th>Value</th>
<th>Error Code</th>
<th>ePDG Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>INTERNAL_ADDRESS_FAILURE</td>
<td>The ePDG sends this code when the ePDG experiences a failure in address assignment.</td>
</tr>
<tr>
<td>37</td>
<td>FAILED_CP_REQUIRED</td>
<td>The ePDG sends this code when the CP payload (CFG_REQUEST) was expected but not received.</td>
</tr>
<tr>
<td>38</td>
<td>TS_UNACCEPTABLE</td>
<td>The ePDG sends this code when the TSi and/or TSr parameters contain IP protocol values other than 0.</td>
</tr>
<tr>
<td>39</td>
<td>INVALID_SELECTORS</td>
<td>The ePDG does not send this code because the selector range is not checked and ingress filtering is applied instead.</td>
</tr>
<tr>
<td>40</td>
<td>TEMPORARY_FAILURE</td>
<td>When it is under collision scenarios as specified in RFC 5996.</td>
</tr>
<tr>
<td>41</td>
<td>CHILD_SA_NOT_FOUND</td>
<td>When it is under collision scenarios as specified in RFC 5996.</td>
</tr>
</tbody>
</table>

The following table lists the notify status types defined in RFCs 4306 and 4739 that are supported by the ePDG.

Table 27. Notify Status Types Supported by the ePDG

<table>
<thead>
<tr>
<th>Value</th>
<th>Notify Status Type</th>
</tr>
</thead>
</table>

The ePDG deletes the session information when a notification is received.
<table>
<thead>
<tr>
<th>Value</th>
<th>Notify Status Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>16388</td>
<td>NAT_DETECTION_SOURCE_IP</td>
</tr>
<tr>
<td>16389</td>
<td>NAT_DETECTION_DESTINATION_IP</td>
</tr>
<tr>
<td>16390</td>
<td>COOKIE</td>
</tr>
<tr>
<td>16393</td>
<td>REKEY_SA</td>
</tr>
</tbody>
</table>