



radisys.

LTE PGW Reference Application

Service Definition

1092419 1.11a

LTE PGW Reference Application

Service Definition

1092419 1.11a

Radisys Corporation

9450 Carroll Park Drive San Diego, CA 92121-2256

Phone: +1 (858) 882-8800

Fax: +1 (858) 777-3389

Web: <http://www.radisys.com>

LTE PGW Reference Application
Service Definition
1092419 1.11a

Continuous Computing, the Continuous Computing logo, Create | Deploy | Converge, Flex21, FlexChassis, FlexCompute, FlexCore, FlexDSP, FlexPacket, FlexStore, FlexSwitch, Network Service-Ready Platform, Quick!Start, TAPA, Trillium, Trillium+plus, Trillium Digital Systems, Trillium On Board, TAPA, and the Trillium logo are trademarks or registered trademarks of Continuous Computing Corporation. Other names and brands may be claimed as the property of others.

This document is confidential and proprietary to Continuous Computing Corporation. No part of this document may be reproduced, stored, or transmitted in any form by any means without the prior written permission of Continuous Computing Corporation.

Information furnished herein by Continuous Computing Corporation, is believed to be accurate and reliable. However, Continuous Computing Corporation assumes no liability for errors that may appear in this document, or for liability otherwise arising from the application or use of any such information or for any infringement of patents or other intellectual property rights owned by third parties, which may result from such application or use. The products, their specifications, and the information appearing in this document are subject to change without notice.

The information contained in this document is provided "as is" without any express representations or warranties. In addition, Continuous Computing Corporation disclaims all statutory or implied representations and warranties, including, without limitations, any warranty of merchantability, fitness for a particular purpose, or non-infringement of third-party intellectual property rights.

To the extent this document contains information related to software products you have not licensed from Continuous Computing Corporation, you may only apply or use such information to evaluate the future licensing of those products from Continuous Computing Corporation. You should determine whether or not the information contained herein relates to products licensed by you from Continuous Computing Corporation prior to any application or use.

Contributors: Continuous Computing Development Team, Naveen D'cruz.

Printed in U.S.A.

Copyright 1998-2011 by Continuous Computing Corporation. All rights reserved.

Contents

Figures	vii
----------------	------------

Tables	ix
---------------	-----------

Preface	xi
----------------	-----------

Objective	xi
Audience	xi
Document Organization	xi
Document Set	xii
Using Continuous Computing® Documentation	xiii
Notations	xiv
Release History	xiv

1 Introduction	1-1
-----------------------	------------

1.1 Product Interfaces	1-1
1.2 Product Entry Points	1-2
1.3 Acronyms	1-3

2 Layer Manager Interface	2-1
----------------------------------	------------

2.1 Primitive Listing	2-1
-----------------------------	-----

2.2	Product-specific Structures	2-2
2.2.1	avMngmt.....	2-2
2.2.2	Pst	2-3
2.3	Primitives and Procedures	2-4
2.3.1	Configuration Request.....	2-5
2.3.1.1	Configuration Data Structure	2-6
2.3.1.1.1	PGW General Configuration.....	2-6
2.3.1.1.2	PGW Lower SAP Configuration.....	2-8
2.3.1.1.3	Configuration Procedure Call Flow	2-9
2.3.2	Configuration Confirm	2-11
2.3.3	Unsolicited Status Indication	2-14
2.3.4	Control Request	2-17
2.3.5	Control Confirm	2-20
3	PGW Application Initialization and Shutdown	3-1
3.1	PGW Application Initialization	3-2
3.2	PGW Application Shutdown	3-4
	References	R-1
	Index	1-1

Figures

Figure 1-1	Trillium Advanced Portability Architecture	1-1
Figure 2-1	PGW Application Configuration Data Flow	2-10
Figure 2-2	Management – Unsolicited Status Procedure	2-16
Figure 2-3	Management – Control Request Procedure	2-20
Figure 3-1	PGW Application Initialization	3-2
Figure 3-2	Lower Layer Bind	3-3
Figure 3-3	Service Provider SAP Unbind	3-4
Figure 3-4	Shutdown Request	3-5

Tables

Table 1-1	LTE PGW Reference Application Interfaces	1-2
Table 1-2	Abbreviations	1-3
Table 2-1	Configuration primitives	2-1
Table 2-2	Unsolicited Status primitives	2-1
Table 2-3	Control primitives	2-2
Table 2-4	AvMngmt Data Structure	2-2
Table 2-5	Pst Structure fields	2-3
Table 2-6	Abbreviations used in flow diagrams	2-4
Table 2-7	Allowable Action Values of the PGW Application Layer	2-19

Preface

Objective

This document provides a detailed description of the services at the layer manager interface by the LTE PGW Reference Application software (p/n 1000419), designed by Continuous Computing Corporation. This product is referred to as PGW Application in the rest of the document.

Audience

Continuous Computing assumes that the readers of this document are familiar with telecommunication protocols, specifically LTE.

Document Organization

This document is organized into the following sections.

Section	Contents
1 Introduction	Describes the product and the interworking process.
2 Layer Manager Interface	Describes the layer manager interface primitives and procedures specific to PGW Application.
3 PGW Application Initialization and Shutdown	Defines the mechanisms to initialize the PGW Application and preparing it for execution. It also describes the procedures to shutdown.

Document Set

The suggested reading order of this document set is:

1. *LTE PGW Reference Application Functional Specification*

Describes the features and highlights the protocol and system characteristics of the software, including the memory characteristics and conformance details.

2. *LTE PGW Reference Application Service Definition*

Describes the procedures and the layer manager interface that are used to pass information between the software and the other software elements. The Interface Primitives section describes the software services. The Procedures section describes and shows the flow of primitives and messages across the interfaces.

3. *EGT Interface Service Definition*

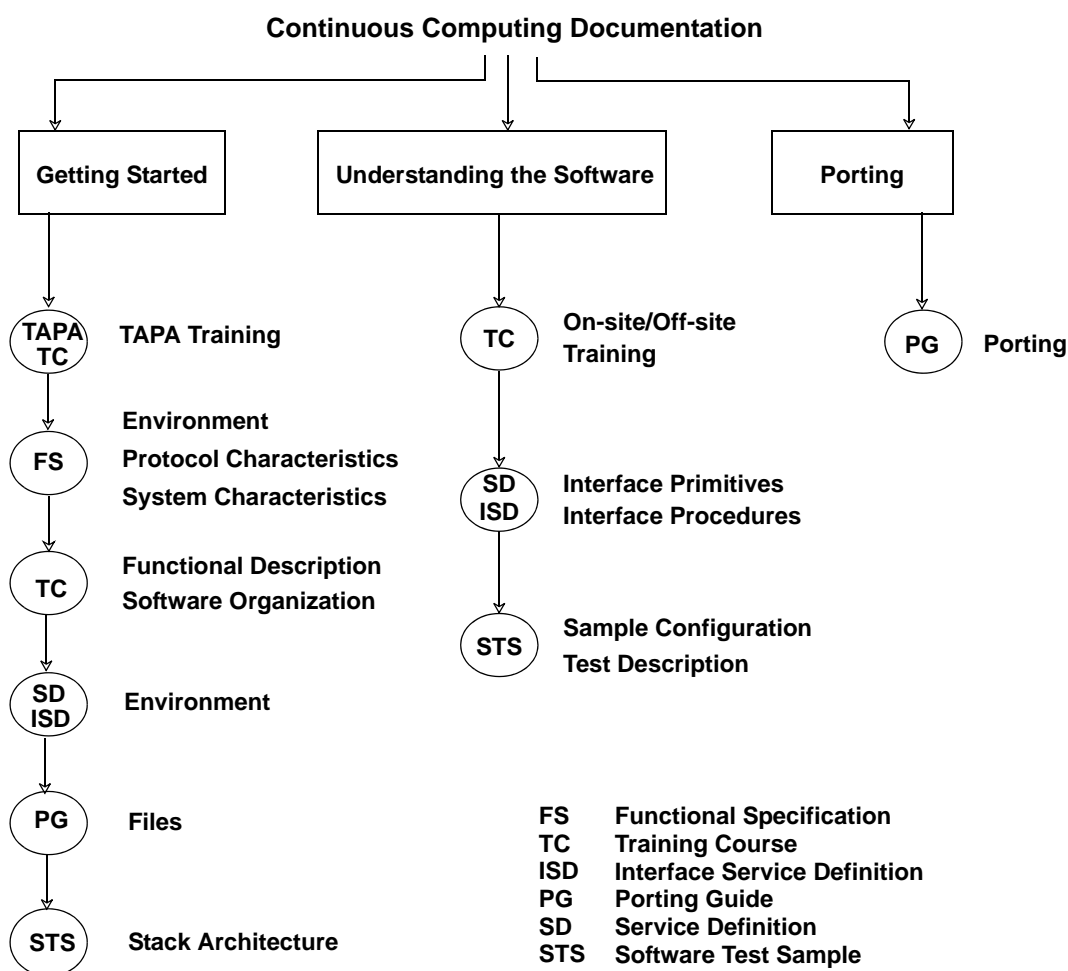
Provides details about the internal lower layer primitives for the EGT interface with Trillium LTE eGTP Product. The Interface Primitives section describes the software services. The Interface Service Definition describes the interface procedures defined for the service provider software.

4. *eGTP Service Definition*

Describes the procedures for eGTP at the layer manager interface. The Layer manager interface is used to configure, control, and collect the status and statistics information from the eGTP software.

Using Continuous Computing® Documentation

The following figure shows the various user approaches to using the software documentation. First time users must read the documents under the **Getting Started** column, where important sections and subsections are listed to the right of each document. For users familiar with the documentation, but who need to look up certain points concerning software use, **Understanding the Software** column is suggested. The **Porting** column is for users familiar with Trillium software and related telecommunications protocols and wish to install the software immediately onto their development environments.



Notations

This table displays the notations used in this document:

Notation	Explanation	Examples
Arial	Titles	1.1 Title
Book Antiqua	Body text	This is body text.
Bold	Highlights information	Loose coupling, tight coupling, upper layer interface
ALL CAPS	CONDITIONS, MESSAGES	AND, OR CONNECT ACK
<i>Italics</i>	<i>Document names, emphasis</i>	<i>PGW Application Service Definition.</i> This adds <i>emphasis</i> .
Courier New Bold	Code Filenames, pathnames	PUBLIC S16 AvMiLavCfgReq (pst, cfg) Pst *pst; AvMngmt *cfg;

Release History

This table lists the history of changes in successive revisions to this document:

Version	Date	Author (s)	Description
1.11a	October 20, 2011	Naveen Dcruz H	Addendum release for Radisys logo and template upgrade.
1.1	July 19, 2010	Rishi Raj	Initial release. Conforms to LTE PGW Reference Application software release, version 1.1.

Introduction

1.1 Product Interfaces

Figure 1-1 shows the PGW Application interfaces.

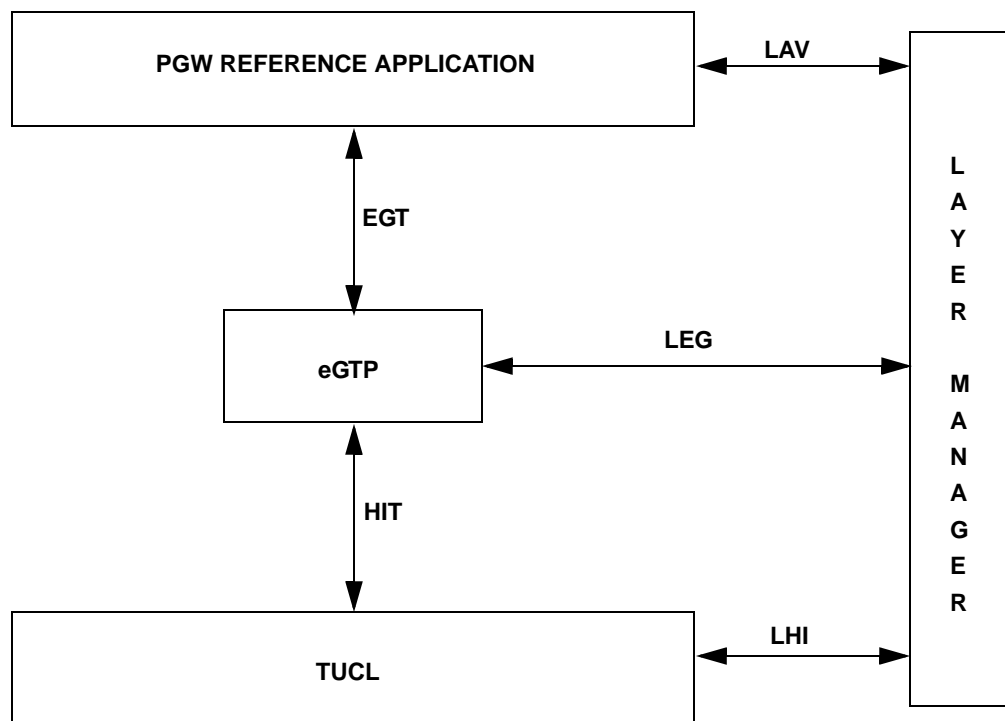


Figure 1-1: Trillium Advanced Portability Architecture

The following table lists the interfaces and describes their functions.

Table 1-1: LTE PGW Reference Application Interfaces

Interface	Description
System Services Interface (SSI)	Provides functions such as buffer management, timer management, date/time management, resource checking, initialization. Refer to the <i>System Services Interface Service Definition</i> for details.
Layer Manager Interface (LMI)	Provides the necessary functions to configure, control, and monitor the condition of each protocol layer. The interface is described later in this document.
Upper Interface (UI)	PGW Application is at top most layer in the protocol stack, it does not have any upper user.
Lower Interface (EGT)	PGW Application supports EGT interface with eGTP. The lower interface (EGT) provides the necessary primitives for PGW Application to configure eGTP layer parameters and data transfer. eGTP layer is used by the PGW Application layer to access its service provider's (lower layer) services. Refer to the <i>EGT Interface Service Definition</i> for a detailed description of this interface.

PGW Application interacts with the other layers and the layer manager by using the primitives and Service Access Points (SAPs) that are described later. PGW Application also interacts with system services by using a simple function interface.

1.2 Product Entry Points

Initialization Entry Point **AvPgwSndActvInit, AvPgwRcvActvInit** - Initializes a task for PGW Application.

Task Entry Point **AvPgwSndActvTsk, AvPgwRcvActvTsk** - Schedules a layer activation task for PGW Application.

1.3 Acronyms

The following acronyms are used in this document:

Table 1-2: Abbreviations

Abbreviation	Expansion
ANSI	American National Standards Institute
LI	Lower Interface
LMI	Layer Manager Interface
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
SAP	Service Access Point
SDU	Service Data Unit
SSI-SDK	System Services Interface - Software Development Kit
TAPA	Trillium Advanced Portability Architecture
UI	Upper Interface

For a list of commonly used terms, refer to the Engineering Glossary (part number PREN026) at <http://www.ccpu.com/search/glossary/>

Product-specific Glossary

Abbreviation	Expansion
CNE	Core Network Emulator
eGTP/e-GTP	Evolved GTP
ENB/eNB/ eNodeB	Evolved Node B
FDD	Frequency Division Duplex
MAC	Medium Access Control
MME	Mobile Management Entity
NAS	Non Access Stratum
P-GW/PGW	PDN Gateway
RB	Radio Bearer
RLC	Radio Link Control

Abbreviation	Expansion
RRC	Radio Resource Control
S1AP	S1 Application Protocol
SCTP	Stream Control Transmission Protocol
SRB	Signalling Radio Bearer
SG	Serving Gateway
SM	Stack Manager
LTE	Long Term Evolution
TDD	Time Division Duplex
TUCL	TCP/UDP Convergence Layer
UE	User Equipment

2

Layer Manager Interface

This section describes the layer manager interface primitives, data structures, and procedures specific to PGW Application. The layer manager interface for PGW Application is described in the following subsections.

2.1 Primitive Listing

The layer manager interface provides the following functions.

Configuration

This function configures the protocol layer resources.

Table 2-1: Configuration primitives

Name	Description	Reference
avMiLavCfgReq	Configuration request	Section 2.3.1
avMiLavCfgCfm	Configuration confirm	Section 2.3.2

Unsolicited Status

This function indicates a change in the status of the protocol layer.

Table 2-2: Unsolicited Status primitives

Name	Description	Implemented by
avMiLavStaInd	Status indication	Section 2.3.3

Control

This function activates and deactivates the protocol resources.

Table 2-3: Control primitives

Name	Description	Implemented by
avMiLavCntrlReq	Control request	Section 2.3.4
avMiLavCntrlCfm	Control confirm	Section 2.3.5

2.2 Product-specific Structures

Each management primitive has two common parameters like: **avMngmt** and **Pst**. These parameters are described here.

2.2.1 avMngmt

The management structure carries all the information required to configure and control the PGW Application layer.

```
typedef struct avMngmt
{
    Header      hdr;           /* Header */
    CmStatus    cfm;           /* Result of operation */
    union
    {
        AvCfg    cfg;           /* Configuration */
        AvCntrl   cntrl;         /* Control */
        AvUstaDgn usta;         /* Unsolicited status */
    }u;
} AvMngmt;
```

Table 2-4: AvMngmt Data Structure

Data Field	Description	Reference
hdr	Header	--
cfm	Specify whether operation is success or not.	--
cfg	Carry all the configuration parameters for the PGW Application layer.	Section 2.3.1.1
cntrl	Control structure	Section 2.3.4
usta	Unsolicited status	Section 2.3.3

This structure is used in every management primitive invoked.

2.2.2 Pst

In TAPA, a system consists of multiple TAPA entities or tasks the post structure:

- Is used to exchange and communicate primitives between various tasks.
- Is the first parameter of all UI, MI, and LI primitives
- Contains all the information required to identify the source and destination TAPA tasks. When the interface is loosely coupled between the source and the destination layers, the source layer provides all the information required by the system services to route the message buffer to the correct destination layer. In the destination layer, the post structure is only used to verify the identity of the source and to identify the specific primitive.
- Contains information that allows a message scheduler to schedule message efficiently. Each message can be assigned a priority. Messages can also be broadcast to all the tasks.

```
typedef struct    pst                /* Parameters for SPstTsk */
{
    ProcId        dstProcId;         /* Destination processor ID (U16) */
    ProcId        srcProcId;         /* Source processor ID (U16) */
    Ent           dstEnt;            /* Destination entity (U8) */
    Inst          dstInst;           /* Destination Instance (U8) */
    Ent           srcEnt;            /* Source entity (U8) */
    Inst          srcInst;           /* Source Instance (U8) */
    Prior         prior;             /* Priority (U8) */
    Route         route;             /* Route (U8) */
    Event         event;             /* Event (U8) */
    Region        region;            /* Region (U8) */
    Pool          pool;              /* Pool (U8) */
    Selector      selector;          /* Selector (U8) */
    CmIntfVer     intfVer;           /* Interface version (U16) */
} Pst;
```

The values used for the individual fields of the Post Structure at the management interface are described in Table 2-5. For primitives initiated by the stack manager to PGW Application, the Post Structure is initialized as shown below:

Table 2-5: Pst Structure fields

Parameter	Description and Allowable values
dstProcId	Processor ID of PGW Application. Allowable values: 0 to 255.
dstEnt	Entity ID of PGW Application. Allowable value: ENTEU
dstInst	Instance ID of PGW Application. Allowable values: 0 to 255
srcProcId	Processor ID of Stack manager. Allowable values: 0 to 255

Table 2-5: Pst Structure fields

Parameter	Description and Allowable values
srcEnt	Entity ID of Stack manager. Allowable value: ENTSM
srcInst	Instance ID of Stack manager. Allowable values: 0 to 255
prior	Priority value for the specific event.
route	Route information, if any.
event	The specific management request event.
region	Memory region information.
pool	Memory pool information.
selector	Specifies whether the stack manager is loosely coupled or tightly coupled with PGW Application. This field is used to decide how the primitive is routed from the layer manager to the PGW layer.

2.3 Primitives and Procedures

The following rules apply to each flow diagram in this section.

1. Time flows toward the bottom of the page.
2. The mnemonic above a line represents a function call or PGW Application primitive.
3. The mnemonic below a line represents a PGW Application message type.
4. A + indicates an OR condition (one path or another can be taken).
5. A o indicates an AND condition (both paths are taken in parallel).

The following table define the abbreviations above the flow diagrams.

Table 2-6: Abbreviations used in flow diagrams

Abbreviation	Definition
ss	System services
eg	eGTP Layer
av	PGW Application
lm	Layer manager

The PGW Application-specific procedures are described in this document.

Note: In the primitives listed below, the type refers to whether these parameters are:

- **Mandatory** (customer needs to fill in the value) denoted by **M**.
 - **Optional** (customer may or may not fill in the value) denoted by **O**.
 - **Not Required** (the value is filled in by the layer below) denoted by **NR**.
-

2.3.1 Configuration Request

Name	Configuration request. SmMiLavCfgReq is invoked in the layer manager. avMiLavCfgReq processes this request in PGW Application.
Direction	Layer manager to PGW Application.
Response	Configuration Confirm. AvMiLavCfgCfm is invoked in the PGW. SmMiLavCfgCfm processes this request in layer manager.
Location	avsm_avcfg.c and av_mi.c

Primitives

avMiLavCfgReq

SmMiLavCfgReq

Synopsis

```
PUBLIC S16 avMiLavCfgReq (pst, cfg)
Pst      *pst;
avMngmt  *cfg;
```

Parameters

pst

Pointer to the post structure. For the configuration request, the event field in the **Pst** structure is set to **EVTLAVCFGREQ**. Refer to Section 2.2.2.

cfg

This parameter represents the management structure described in Section 2.2.1, "AvMngmt." The parameters specific configuration request are described. This parameter is the pointer to the configuration structure. The configuration structure has the following format:

2.3.1.1 Configuration Data Structure

```
typedef struct avMngmt
{
    Header      hdr;      /* Header */
    CmStatus    cfm;      /* Result of operation */
    union
    {
        AvCfg    cfg;      /* Configuration */
        ....
        ....
    }u;
} AvMngmt;

typedef struct avCfg
{
    union
    {
        AvPgwGenCfg  pgwGenCfg;    /* PGW General config */
        AvPgwPeerCfg peerCfg;      /* All peer details for this
                                   PDN-GW */
        AvPgwLSapCfg pgwLSapCfg;   /* PGW Lower SAP config */
    }s;
} AvCfg;
```

Parameter	Description	Reference
AvPgwGenCfg	PGW general configuration	Section 2.3.1.1.1
AvPgwLSapCfg	PGW Lower SAP configuration	Section 2.3.1.1.2

2.3.1.1.1 PGW General Configuration

This structure provides the general configuration information for the layer. In the general configuration request, the **ElmntId** field of the Header structure must be set to STGEN:

```
hdr.elmId.elmnt = STGEN.
```

```
typedef struct
{
    Pst      lmPst;           /* Post structure for layer manager */
    Mem      mem;            /* Memory pool and region */
    U16      tmrRes          /* Timer resolution */
    U16      maxNmbEgtSaps;  /* Maximum number of lower (EGT) SAPs */
    U16      maxNmbTunls;   /* Maximum number of eGTP-u tunnels */
    U16      maxNmbBearers; /* Maximum number of bearers */
    U16      maxNmbUes;     /* Maximum number of UEs */
    AvPgwAddrCfg pgwAddrCfg; /* PDN-GW's self details */
} AvPgwGenCfg
```

The parameters are:

Field	Description
lmPst	<p>This is the post structure that the PGW uses to communicate with the stack manager. In this structure, the values of the following parameters must be set as indicated:</p> <p>dstEnt: Entity ID of the stack manager -ENTSM. dstInst: Instance ID of the stack manager. dstProcId: Processor ID of the stack manager. prior: Priority with which PGW must send any messages to the stack manager. route: Any routing information provided to system services, if the stack manager and PGW are loosely coupled. selector: The coupling used when messages are sent from PGW to the stack manager. region and pool: Memory region and memory pool from which message buffers are allocated for any messages sent from PGW to the stack manager. srcEnt, srcInst, and srcProcId must not be filled. PGW receives this information at initialization time from system services and fills in the appropriate values.</p>
mem	Memory pool and region.
tmrRes	Timer resolution.
maxNmbEgtSaps	Maximum number of lower (EGT) SAPs.

Field	Description
maxNmbTunls	Maximum number of eGTP-u tunnels.
manNmbBearers	Maximum number of bearer.
maxNmbUes	Maximum number of UEs.
AvPgwAddrCfg	PDN-GW's self details.

2.3.1.1.2 PGW Lower SAP Configuration

The PGW lower SAP configuration structure is given below.

```
typedef struct AvPgwLSapCfg
{
    SuId          suId;          /* S1AP SAP ID */
    SpId          spId;          /* Service Provider SAP ID */
    Selector      selector;      /* Coupling */
    MemoryId      mem;           /* Memory pool */
    Priority      prior;          /* Priority */
    Route         route;         /* Route */
    ProcId        dstProcId;     /* Destination processor ID */
    Ent           dstEntId;       /* Destination entity ID */
    Inst          dstInstId;      /* Destination instance ID */
    U8            maxBndRetry;    /* Maximum number of bind retries
                                allowed */
    TmrCfg        tBndTmr;       /* Bind timer for application */
    U32           nmbConn;        /* Number of bearers through this
                                SAP */
}AvPgwLSapCfg;
```

The parameters are:

Field	Description
suId	S1AP SAP ID or eGTP SAP ID
spId	Service provider SAP ID
selector	Lower interface selector
mem	Lower interface Memory pool and region
prior	Lower interface priority.
route	Lower interface route
dstPorcId	Destination processor ID
dstEntId	Destination entity ID
dstInstId	Destination instance ID

Field	Description
maxBndRetry	Maximum number of bind retries allowed
tBndTmr	Bind timer
nmbConn	Number of bearers through this SAP

2.3.1.1.3 Configuration Procedure Call Flow

The layer manager configures the various elements of PGW Application using the management – configuration procedure, which the layer manager initiates. The PGW Application configuration request primitive (**avMiLavCfgReq**) can be called more than once. **avMiLavCfgReq** primitives must be called before the bind primitives are called.

The following table lists the PGW Application configuration request primitive types.

Name	Description
General	Passes parameters that apply to the entire PGW Application software. It reserves the memory pool for its static memory requirements and register for timer service with SSI. It can be called once.
Lower SAP eGTP	PGW Application lower SAP configuration. The SAP is used to communicate with the provider eGTP at the EGT interface.

To operate properly, the configuration request primitive types must be called in the following order:

1. General
2. Lower SAP eGTP

The system services primitives are called during the management – configuration procedure

The **avMngmt.t.cfg** structure specifies the parameters that the configuration request primitive (**avMiLavCfgReq**) uses.

Figure 2-1 shows the PGW Application configuration data flow.

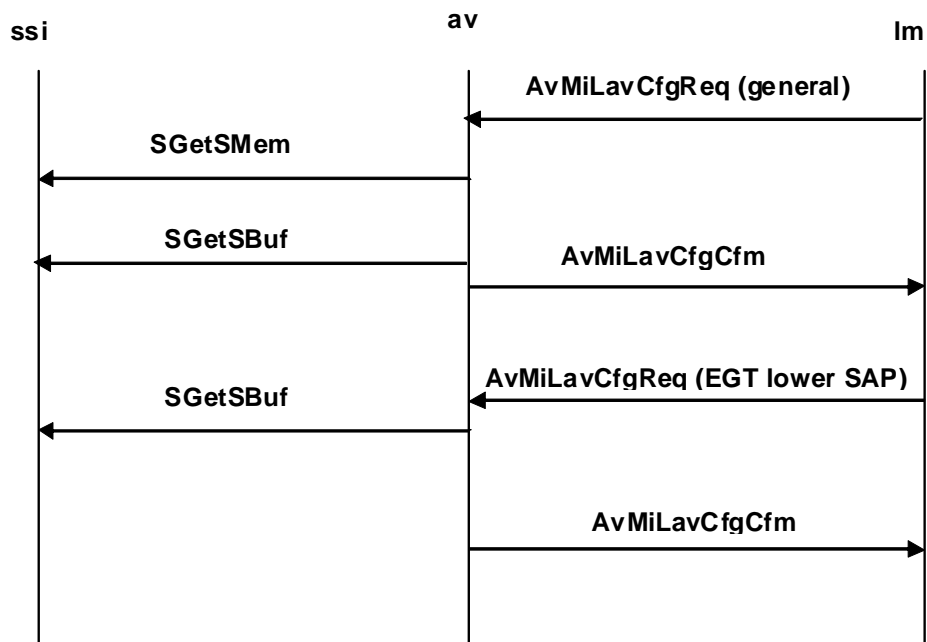


Figure 2-1: PGW Application Configuration Data Flow

2.3.2 Configuration Confirm

Name	Configuration confirm. avMiLavCfgCfm is invoked in PGW Application. SmMiLavCfgCfm processes this confirm primitive in the layer manager.
Direction	PGW Application to Layer Manager
Response	Not applicable
Location	av_mi.c and avsm_avcfg.c

Primitives

avMiLavCfgCfm

SmMiLavCfgCfm

Synopsis

```
PUBLIC S16 avMiLavCfgCfm(pst, cfm)
Pst      *pst;
avMngmt  *cfm;
```

Parameters

pst

Pointer to the post structure. For the configuration request, the event field in the Post structure is set to **EVTLAVCFGCFM**. Refer to Section 2.2.2

cfm

This parameter represents a pointer to the management structure described in Section 2.2.1. The configuration confirm specific parameters are described in the following sections.

```
typedef struct avMngmt
{
    Header      hdr;           /* Header */
    CmStatus    cfm;           /* Result of operation */
    union
    {
        ....
        ....
        ....
    }u;
} AvMngmt;
```

In the **Header** of the configuration confirm, the values of the relevant fields are set, as shown in the following table.

Field	Allowable Values
hdr.elmId.elmnt	Copied from the received configuration request primitive. Allowable values: <ul style="list-style-type: none"> • STAVGEN: For general configuration. • STAVEGTUSAP: For eGTP-u SAP configuration. • STAVEGTCSAP: For eGTP-c SAP configuration.
hdr.transId	Copied from the received configuration request to enable correlation of requests in the stack manager.

The status information is returned in the **CmStatus** data structure. For the configuration confirm primitive, the values of the **CmStatus** are:

status

Name	Description
LCM_PRIM_OK	Configuration request is processed to successful completion.
LCM_PRIM_NOK	The processing of the configuration request is aborted owing to an error. The error is qualified in the reason field.
LCM_PRIM_OK_NDONE	Configuration request is processed is not completed.

reason

The content of this field is set to **LCM_REASON_NOT_APPL** when **status** is set to **LCM_PRIM_OK**. When the **status** field is set to **LCM_PRIM_NOK**, the content of the reason field can be set to any of the following values, which indicates the reason for the failure of the primitive. The following table lists the values valid for general and Upper SAP configuration.

Value	Description
LCM_REASON_NOT_APPL	Reason not applicable. It is used with LCM_PRIM_OK.
LCM_REASON_INVALID_ENTITY	Invalid entity specified in PGW entity configuration request.
LCM_REASON_INVALID_INSTANCE	Invalid instance specified in PGW instance configuration request.
LCM_REASON_INVALID_MSGTYPE	Invalid message type.
LCM_REASON_MEM_NOAVAIL	Memory allocation failed. Either memory cannot be reserved using SGetSMem , or memory cannot be allocated to the required static structures using SGetSBuf .
LCM_REASON_INVALID_ELMNT	Value of the hdr.elmId.elmnt is invalid.
LCM_REASON_RECONFIG_FAIL	Reconfiguration on the specific element failed.
LCM_REASON_REGTMR_FAIL	Timer registration (SRegTmr) failed.
LCM_REASON_GENCFG_NOT_DONE	General configuration is not done.
LCM_REASON_INVALID_ACTION	Invalid action.
LCM_REASON_INVALID_SUBACTION	Invalid sub action.
LCM_REASON_INVALID_STATE	Invalid state.
LCM_REASON_INVALID_SAP	Invalid SAP ID in the SAP configuration request.
LCM_REASON_INVALID_PAR_VAL	Invalid parameter value.
LCM_REASON_QINIT_FAIL	Queue initialization failure.
LCM_REASON_NEG_CFM	Negative confirmation.
LCM_REASON_UPDTMR_EXPIRED	Update timer expired.
LCM_REASON_MISC_FAILURE	Miscellaneous failures.
LCM_REASON_EXCEED_CONF_VAL	Exceeds configured value.
LCM_REASON_HASHING_FAILED	Hashing failed.
LCM_REASON_PEERCFG_NOT_DONE	Peer configuration is not done.

2.3.3 Unsolicited Status Indication

Name	Unsolicited Status Indication. avMiLavStaInd is invoked in PGW Application, and SmMiLavStaInd processes confirm primitive in the layer manager.
Direction	PGW Application to Layer Manager
Response	Not applicable
Location	av_mi.c and avsm_avcfg.c

Primitives

avMiLavStaInd

SmMiLavStaInd

Synopsis

```
PUBLIC S16 avMiLavStaInd(pst, sta)
Pst      *pst;
avMngmt *usta;
```

Parameters

pst

Pointer to the post structure. For the configuration request, the event field in the **Pst** structure is set to EVTLAVUSTAIN. Refer to Section 2.2.2

usta

Unsolicited status indication. It is a pointer to the management structure. It has the following format.

```
typedef struct avMngmt
{
    Header      hdr;          /* Header */
    CmStatus    cfm;          /* Result of operation */
    union
    {
        ....
        ....
        AvUstaDgn usta;      /* Alarm */
    }u;
} AvMngmt;
```

```
typedef struct _avUstaDgn
{
    CmAlarm      alarm;
} AvUstaDgn;
```

hdr

Header structure.

cfm

Common status structure. It is not used for this primitive.

alarm

Alarm category and event. This structure provides information about the date and time, category, event, and cause for an alarm generation.

```
typedef struct cmAlarm
{
    DateTime      dt;           /* Date and time */
    U16           category;     /* Category of the alarm */
    U16           event;        /* Event */
    U16           cause;        /* Cause for event */
} CmAlarm;
```

Primitive	Description
dt	Date and time.
category	Specifies the category of the alarm.

The allowable values of **category** are:

Allowable Values	Description
LCM_CATEGORY_INTERFACE	Interface-related alarm.
LCM_CATEGORY_RESOURCE	System resource related alarm.
LCM_CATEGORY_PROTOCOL	Protocol-related alarm.
LCM_CATEGORY_INTERNAL	Internal state-related alarm.

evnt

This parameter specifies the event that caused the generation of a status indication to the layer manager, from the PGW software. Event codes are not unique and must be interpreted in conjunction with the category of the generated alarm. For each alarm category, the allowable values are:

Allowable Values	Description
LAV_EVENT_PGW_ROUTCFM	Route to external entity is done.

cause

This parameter specifies the cause for the alarm. The allowable values are:

Allowable Values	Description
LAV_CAUSE_ARP_SUCC	ARP request to external entity successful.

Description

The management – unsolicited status procedure provides unsolicited status information about PGW Application elements to the layer manager. PGW Application initiates this procedure. The PGW Application status indication primitive (**avMiLavStaInd**) can be called more than once and, if the unsolicited status is enabled, at any time after the configuration procedure. The PGW Application status indication primitive is not called if the unsolicited status is disabled. The unsolicited status can be enabled or disabled with the management – control procedure.

The **avMngmt.t.usta** structure specifies parameters that are used by the status indication (**avMiLavStaInd**) primitive.

Figure 2-2 shows the management – unsolicited status procedure.

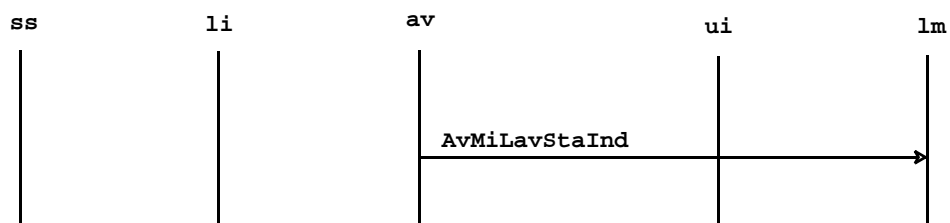


Figure 2-2: Management – Unsolicited Status Procedure

Note: The PGW Application uses this function to alert the layer manager to a significant change in the software state, or when an error is encountered while processing a primitive.

2.3.4 Control Request

Name	avMiLavCntrlReq
Direction	Layer Manager to PGW Application.
Response	Control Confirm. AvMiLavCntrlCfm is invoked in the PGW. SmMiLavCntrlCfm processes this request in layer manager.
Location	av_mi.c and avsm_avcfg.c

Primitives

avMiLavCntrlReq

SmMiLavCntrlReq

Synopsis

```
PUBLIC S16 avMiLavCntrlReq(pst, cntrl)
Pst      *pst;
avMngmt  *cntrl;
```

Parameters

pst

Pointer to the post structure. For the configuration request, the event field in the **Pst** structure is set to **EVTLAVCNTRLREQ**. Refer to Section 2.2.2

cntrl

Pointer to the control structure. Control structure has the following format:

```
typedef struct avMngmt
{
    Header          hdr;    /* Header */
    CmStatus        cfm;    /* Result of operation */

    union
```

```

    {
        ....
        AvCntrl    cntrl; /* Control */
        ....
    }u;
} AvMngmt;
typedef struct AvCntrl
{
    union
    {
        AvPgwCntrl    pgwCntrl;
    }u;
} AvCntrl;
typedef struct avPgwCntrl
{
    DateTime    dt;          /**< Date */
    U8          action;      /**< Action to be performed Bind,
                               Unbind or shutdown */
    U8          subAction;   /**< Subaction to be performed,
                               SADEGB and SAUSTA */

    union
    {
#ifdef DEBUGP
        AvDbgCntrl    dbg;
#endif /* DEBUGP */
        AvSapCntrl    sap;    /**< The layer manager can
                               selectively bind/unbind the
                               different SAPs */

    }u;
}AvPgwCntrl;
typedef struct _avDbgCntrl
{
    U32 dbgMask;
}AvDbgCntrl;
typedef struct avSapCntrl
{
    SpId id;
}AvSapCntrl;

```

The parameters are:

Parameter	Description
dt	This field is not used in this primitive. It is used in the avLavCntrlCfm primitive.
action	This field specifies the action that the PGW Application layer must take.
subAction	This field specifies the protocol element on which the PGW Application layer takes the specified action.

Parameter	Description
sap	Contains the information required for SAP control.
dbg dbgMask	<p>The layer manager can selectively enable/disable various levels of debug printing.</p> <p>The allowable values of the dbgMask are:</p> <p>LAV_DBGMASK_MEM- Memory debug mask</p> <p>LAV_DBGMASK_TRC - Trace debug mask</p> <p>LAV_DBGMASK_ERROR- Error debug mask</p> <p>LAV_DBGMASK_INFO - Information debug mask</p> <p>dbgMask can be set to a combination of the aforementioned allowable values. For example, if the debugging prints at the upper and lower interfaces of the layer are to be enabled/disabled, the dbgMask is set to DBGMASK_UI DBGMASK_LI.</p>

Table 2-7: Allowable Action Values of the PGW Application Layer

action	subaction	elmnt
AENA	SADBG	STAVGEN
ADISIMM	SAUSTA	STAVEGTSAP
ASHUTDOWN	--	--
ADEL	--	--
ABND	--	--
AUBND	--	--

Description

The layer manager uses the management – control procedure to control PGW Application elements. The layer manager initiates this procedure. The PGW Application control request primitive (**avMiLavCntrlReq**) can be called more than once and at any time after the management – configuration procedure.

The following PGW Application control request primitive types can be called:

The **avMngmt.t.cntrl.action** field specifies the control request primitive type.

The **avMngmt.t.cntrl.subaction** field specifies the element to be controlled.

The **avMngmt.t.cntrl** structure specifies the parameters that are used by the control request (**avMiLavCntrlReq**) primitive.

Figure 2-3 shows the management – control request procedure.

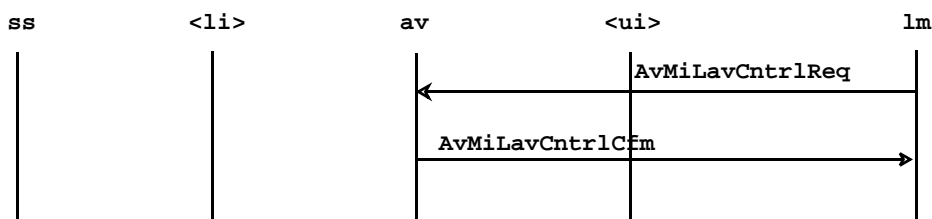


Figure 2-3: Management – Control Request Procedure

2.3.5 Control Confirm

Name	Control Confirm. avMiLavCntrlCfm is invoked in PGW Application, and SmMiLavCntrlCfm processes this confirm primitive in the layer manager.
Direction	PGW Application to Layer Manager
Response	Not applicable
Location	av_mi.c and avsm_avcfg.c

Primitives

avMiLavCntrlCfm

SmMiLavCntrlCfm

Synopsis

```
PUBLIC S16 avMiLavCntrlCfm(pst, cfm)
```

```
Pst      *pst;
```

```
avMngmt  *cfm;
```

Parameters

pst

Pointer to the post structure. For the configuration request, the event field in the **Pst** structure is set to **EVTLAVCNTRLCFM**. Refer to Section 2.2.2

cfm

This parameter represents a pointer to the management structure described in Section 2.2.1. The control confirm specific parameters are described in the following sections.

```
typedef struct avMngmt
{
    Header      hdr;          /* Header */
    CmStatus    cfm;          /* Result of operation */
    union
    {
        ....
        ....
        ....
    }u;
} AvMngmt;
```

In the **Header** parameter of the configuration confirm primitive, the values of the relevant fields are set as follows:

Field	Allowable Values
hdr.elmId.elmn	Copied from the received configuration request primitive. Allowable values: STAVGEN : For general configuration. STAVEGTUSAP : For eGTP SAP configuration. STAVEGTCSAP : for eGTP C SAP configuration.
hdr.tranId	Copied from the received control request to enable correlation of requests in the stack manager.

The status information is returned in the **CmStatus** data structure. For the control confirm primitive, the values of the **CmStatus** are set as follows:

status

Name	Description
LCM_PRIM_OK	Control request is processed to successful completion.
LCM_PRIM_OK_NDONE	The processing of the control request is deferred. This value is sent only if PGW cannot immediately service the control request.
LCM_PRIM_NOK	The processing of the control request is aborted owing to an error. The error is qualified in the reason field.

reason

The contents of this field are set to **LCM_REASON_NOT_APPL**, if the **status** field is set to **LCM_PRIM_OK**.

When the **status** field is set to the value **LCM_PRIM_NOK**, the contents of the reason field can be set to any of the following values, which indicates the reason for the failure of the primitive.

The following values are valid for all control operations:

Value	Description
LCM_REASON_NOT_APPL	Reason not applicable. It is used with LCM_PRIM_OK .
LCM_REASON_INVALID_ENTITY	Invalid entity specified in PGW entity configuration request.
LCM_REASON_INVALID_INSTANCE	Invalid instance specified in PGW instance configuration request.
LCM_REASON_INVALID_MSGTYPE	Invalid message type
LCM_REASON_MEM_NOAVAIL	Memory allocation failed. Either memory cannot be reserved using SGetSMem , or memory cannot be allocated to the required static structures using SGetSBuf .
LCM_REASON_INVALID_ELMNT	Value of the hdr.elmId.elmnt is invalid.
LCM_REASON_RECONFIG_FAIL	Reconfiguration on the specific element failed.
LCM_REASON_REGTMR_FAIL	Timer registration (SRegTmr) failed.
LCM_REASON_GENCFG_NOT_DONE	General configuration is not done.
LCM_REASON_INVALID_ACTION	Invalid action.
LCM_REASON_INVALID_SUBACTION	Invalid sub action.
LCM_REASON_INVALID_STATE	Invalid state.
LCM_REASON_INVALID_SAP	Invalid SAP ID in the SAP configuration request.
LCM_REASON_INVALID_PAR_VAL	Invalid parameter value.
LCM_REASON_QINIT_FAIL	Queue initialization failure.
LCM_REASON_NEG_CFM	Negative confirmation.
LCM_REASON_UPDTMR_EXPIRED	Update timer expired.
LCM_REASON_MISC_FAILURE	Miscellaneous failures.

Value	Description
LCM_REASON_EXCEED_CONF_VAL	Exceeds configured value.
LCM_REASON_HASHING_FAILED	Hashing failed.
LCM_REASON_PEERCFG_NOT_DONE	Peer configuration is not done.

3

PGW Application Initialization and Shutdown

The interface procedures define the mechanisms by which PGW Application software interacts, through primitives, with any adjacent software.

The procedures differ depending on whether a tightly or loosely coupled interface is used. A tightly coupled interface implies that the interface consists of direct function calls between the two layers. A loosely coupled interface implies that the interface consists of passing messages between the two layers through system services. Refer to the *System Services Interface Service Definition* for more details about interface coupling. The following description assumes a tightly coupled interface.

Note: Procedures specific to PGW Application Layer Manager Interface are described in this document.

3.1 PGW Application Initialization

For initialization, the interface procedures must be performed in the following order:

1. Initialization of PGW Application

Initializes the PGW Application software. The layer manager begins the procedure when it registers the initialization function for PGW Application using the **SRegTTsk** primitive. System services then calls the initialization function **AvPgwSndActvInit** and **avActvInit**. The initialization function must be called once before any other primitives or functions in PGW Application are called. It creates a system task by calling the **SCreateSTsk** primitive on SSI. Now, the TAPA task must be attached to this system task to make it scheduled by SSI.

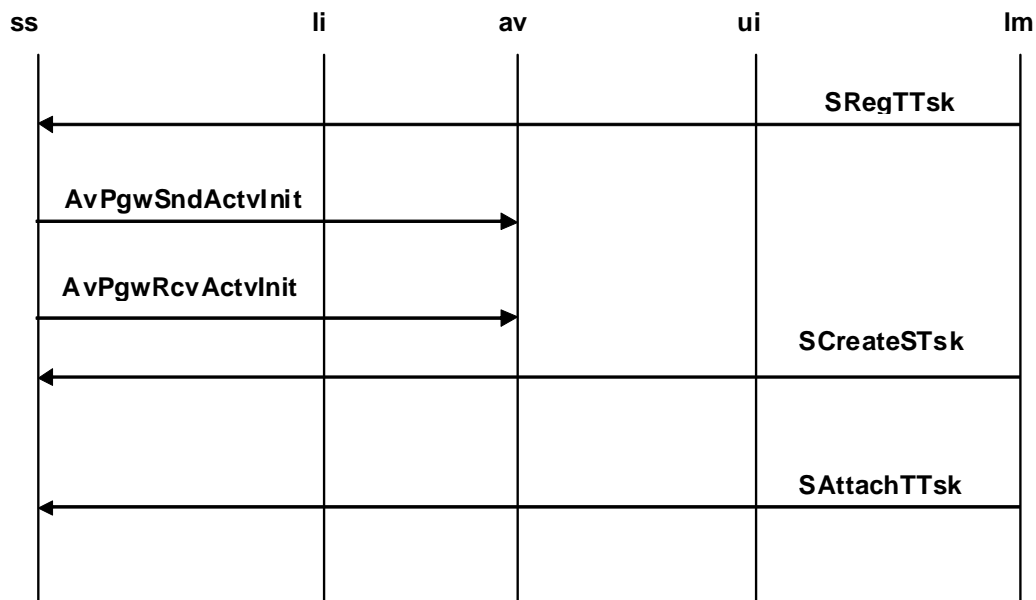


Figure 3-1: PGW Application Initialization

2. Management – Configuration

Configures the PGW Application software. For the configuration sequence, refer to Figure 2-1.

3. Lower layer – Bind

PGW Application software binds itself to its lower layer eGTP. The layer manager initiates the bind with the lower layer by sending the control request to the PGW Application with the **action** set to "Bind and Enable". The PGW Application then sends the bind request to the lower layer. After receiving the bind confirm from the lower layer, PGW Application sends an alarm to the layer manager, if alarms are enabled.

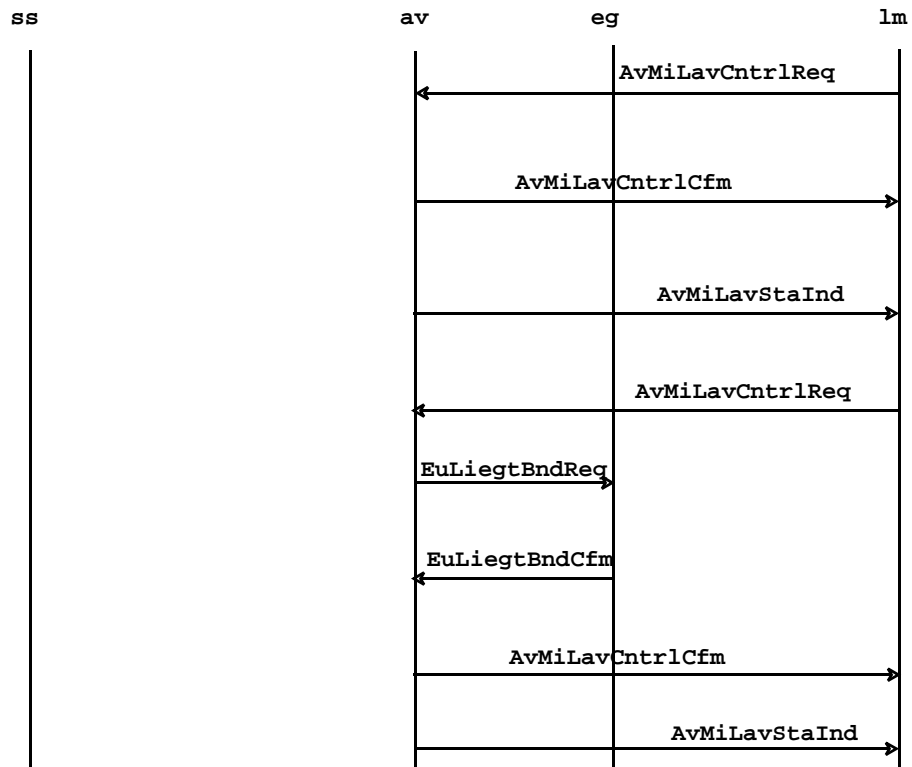


Figure 3-2: Lower Layer Bind

3.2 PGW Application Shutdown

For shutdown, the interface procedures must be performed in the following order:

1. Service provider SAP unbind

PGW Application software unbinds itself from its layer eGTP. The layer manager initiates the unbind with the lower layer by sending the control request to the PGW Application with the **action** set to "Unbind and Disable". The PGW Application then sends the unbind request to the lower layer after clearing the resources associated with this SAP.

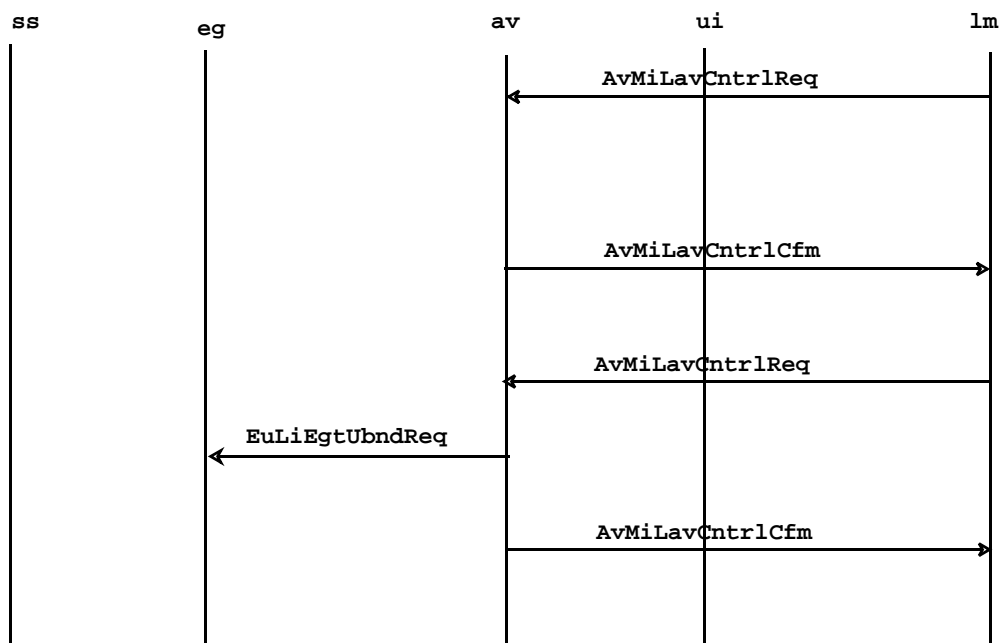


Figure 3-3: Service Provider SAP Unbind

2. Shutdown Request

The layer manager initiates the shutdown request to the PGW Application by sending the control request with the **action** set to "Shutdown". This operation brings back the PGW Application to its pre-configured state.

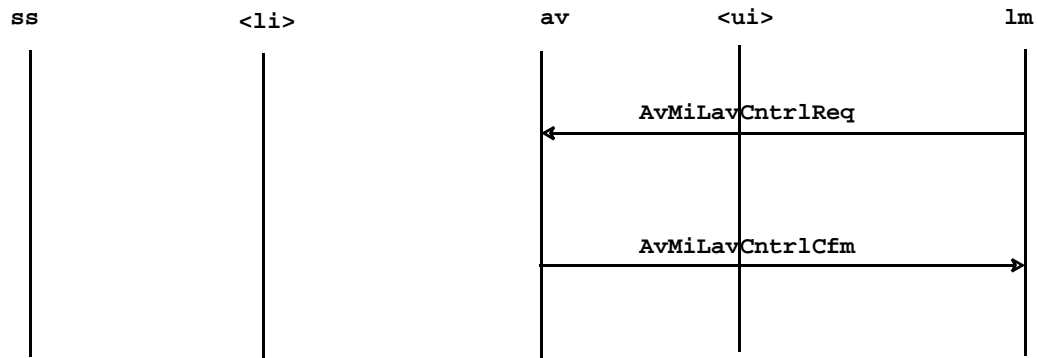


Figure 3-4: Shutdown Request

Now, the layer is deactivated and non-operational.

References

Refer to the following documents for more information:

LTE PGW Reference Application Functional Specification, Continuous Computing Corporation (p/n 1091419)

EGT Interface Service Definition, Continuous Computing Corporation (p/n 1100368).

TAPA Concepts and Guidelines Common Document, Continuous Computing Corporation (p/n 1111011).

System Services Interface Service Definition, Continuous Computing Corporation (p/n 1111001).

3G TS 29.274, version 1.3.0 (2008-10), Evolved GPRS Tunnelling Protocol for Control Plane.

3G TS 29.281, version 8.1.0 (2008-10), GPRS Tunnelling Protocol for User Plane.

Index

C

- configuration primitives 2-1
- configuration request 2-5
- control primitives 2-2
- control request 2-17
- coupling, tight or loose 3-1

D

- data flow diagrams, explained 2-4

H

- HiMiLhiCfgCfm 2-11
- HiMiLhiCfgReq 2-5
- HiMiLhiCntrlCfm 2-20
- HiMiLhiCntrlReq 2-17
- HiMiLhiStaInd 2-14

I

- interface procedures 3-1
- interfaces 1-1, 2-1
 - layer manager 1-2
 - lower layer 1-2
 - system services 1-2
 - upper layer 1-2

L

- layer interfaces 1-1
- layer manager 2-1
 - interaction 1-2
 - interface 1-2, 2-1
 - primitives 2-1
- loose coupling 3-1
- lower layer interface 1-2

M

- management structure 2-2, 2-3
- mandatory 2-5

N

- not required 2-5

O

- optional 2-5

P

- primitive type
 - mandatory 2-5
 - not required 2-5
 - optional 2-5

procedures 2-4, 3-1
product-specific structures 2-2

R

references R-1

S

Service Access Point (SAP) 1-2
SmMiLhiCfgCfm 2-11
SmMiLhiCfgReq 2-5
SmMiLhiCntrlCfm 2-20
SmMiLhiCntrlReq 2-17
SmMiLhiStaInd 2-14
system services
 interface 1-2

T

tight coupling 3-1

U

unsolicited status 2-14
 primitives 2-1
upper layer interface 1-2