**EPC User’s Guide**

|  |  |
| --- | --- |
| Editor: | EURECOM |
| Deliverable nature: | Public |
| Due date: | July, 2015 |
| Delivery date: | July, 2015 |
| Version: | 0.4 |
| Total number of pages: | 33 |
| Reviewed by: |  |
| Keywords: | EPC, LTE, MME, S-GW, HSS, S1AP, DIAMETER |

**Abstract**

The deliverable presents the EPC developed by EURECOM.

The document presents the deployment scenarios of the EPC, its configuration, installation and running.

**List of authors**

|  |  |
| --- | --- |
| **Company** | **Authors** |
| **EURECOM** | Christian BONNET, Lionel GAUTHIER, Rohit GUPTA, Florian KALTENBERGER, Raymond KNOPP, Navid NIKAIEN, Cedric ROUX. |
| **OPEN CELLS** | Laurent THOMAS. |
|  |  |
|  |  |

History

|  |  |  |  |
| --- | --- | --- | --- |
| **Modified by** | **Date** | **Version** | **Comments** |
| Lionel GAUTHIER | 09/06/2015 | 0.1 | Initial Draft |
| Lionel GAUTHIER | 11/06/2015 | 0.2 | Minor corrections |
| Lionel GAUTHIER | 20/07/2015 | 0.3 | Changes in dir and file names for git repo |
| Lionel GAUTHIER | 20/07/2015 | 0.4 | Minor corrections (build, get sources from git) |

Table of Contents

History 3

Table of Contents 4

List of Figures 6

List of tables 7

Abbreviations 8

1 Introduction 9

1.1 Overview 9

1.2 Deployment scenarios 9

1.2.1 Separate EPC platform 9

1.2.2 All in one EPC platform 10

2 EPC Installation 11

2.1 Operating system 11

2.2 EPC source code 11

2.2.1 Get the code without login 11

2.2.2 Get the code with login (contributors) 11

2.3 Additional software, initial steps. 12

2.3.1 Mysql server installation details 12

2.3.2 Phpmyadmin installation details 12

3 EPC Configuration 14

3.1 MME\_GW 14

3.1.1 Fully Qualified Domain name 14

3.1.2 Configuration files 14

3.2 MME 15

3.3 SP\_GW 15

3.4 MME configuration content 15

3.4.1 GUMMEI section 15

3.4.2 SCTP section 16

3.4.3 S1AP section 16

3.4.4 S6A section 16

3.4.5 NAS section 16

3.4.6 INTERTASK\_INTERFACE section 16

3.4.7 Network interfaces section 16

3.5 S-GW configuration content 17

3.6 P-GW configuration content 18

3.6.1 Main section 18

3.6.2 Network interfaces section 18

3.6.3 IP Address Pool section 18

3.7 HSS 19

3.7.1 Fully Qualified Domain name 19

3.7.2 Configuration files 19

3.7.3 HSS database content 22

3.7.4 Configuring your HSS 24

Adding your MME 24

Adding a user 24

4 Building and running 26

4.1 MME\_GW. 26

4.1.1 Configuration files 26

4.1.2 Building EPC 26

4.1.3 Running EPC 26

5 Supported scenarios in EPC 28

5.1 E-UTRAN Initial attach 28

5.1.1 Attach with IMSI 28

5.1.2 Attach with GUTI 28

5.2 Tracking Area Update procedures 28

5.3 Routing Area Update procedures 28

5.4 Service Request procedures 28

5.4.1 UE triggered Service Request 28

5.4.2 Network triggered Service Request 28

5.5 S1 Release procedure 28

5.6 GUTI Reallocation procedure 28

5.7 Detach procedure 28

5.7.1 UE-Initiated Detach procedure for E-UTRAN 28

5.7.2 MME-Initiated Detach procedure for E-UTRAN 28

5.7.3 HSS-Initiated Detach procedure for E-UTRAN 28

5.8 HSS User Profile management function procedure 28

5.9 Bearer deactivation 28

5.9.1 PDN GW initiated bearer deactivation 28

5.9.2 MME initiated Dedicated Bearer Deactivation 28

5.10 Intra E-UTRAN handover 29

6 Annex A: Tools for observing, debugging. 30

6.1 Itti\_analyzer 30

6.1.1 Installation 30

6.1.2 Execution 30

6.2 Wireshark/tshark 31

6.3 Mscgen 31

6.4 S1AP scenario replay 32

6.4.1 Capture a scenario 32

6.4.2 build a S1AP scenario from pcap trace 33

6.4.3 Replay a S1AP scenario 33

List of Figures

Figure 1 EPC overview 9

Figure 2 EPC Deployment in MME SP-GW 9

Figure 3 EPC Deployment in MME\_GW 10

Figure 4 Mysql installation root password 12

Figure 5 Phpmyadmin installation conf DB 12

Figure 6 Phpmyadmin installation DB password 13

Figure 7 Phpmyadmin installation app password 13

Figure 8 Phpmyadmin installation web server selection 13

Figure 9 MME\_GW configuration files generation 14

Figure 10 HSS configuration files generation 19

Figure 11 ITTI Analyzer main window 30

Figure 12 ITTI Analizer select filter menu 31

Figure 13 Mscgen output example 32

List of tables

[Table 1 MME configuration main section 15](#_Toc425958350)

[Table 2 MME configuration subsection GUMMEI 15](#_Toc425958351)

[Table 3 MME configuration subsection SCTP 16](#_Toc425958352)

[Table 4 MME configuration subsection S1AP 16](#_Toc425958353)

[Table 5 MME configuration subsection S6a 16](#_Toc425958354)

[Table 6 MME configuration subsection NAS 16](#_Toc425958355)

[Table 7 MME configuration subsection ITTI 16](#_Toc425958356)

[Table 8 MME configuration subsection Network Interfaces 17](#_Toc425958357)

[Table 9 S-GW configuration main section 17](#_Toc425958358)

[Table 10 P-GW configuration main section 18](#_Toc425958359)

[Table 11 P-GW configuration subsection Network Interfaces 18](#_Toc425958360)

[Table 12 P-GW configuration subsection IP Address Pool Selection 18](#_Toc425958361)

[Table 13 SQL Table structure mmeidentity 22](#_Toc425958362)

[Table 14 SQL Table structure pdn 23](#_Toc425958363)

[Table 15 SQL Table structure users 24](#_Toc425958364)

Abbreviations

3GPP Third Generation Partnership Project.

APN Access Point Name.

CIDR Classless Inter-Domain Routing.

eNB e Node B.

EPC Evolved Packet Core.

EPS Evolved Packet System.

FQDN Fully qualified domain name.

HSS Home Subscriber Server.

IMEI International Mobile Station Equipment Identity.

IMEISV International Mobile Station Equipment Identity Software

Version.

LTE Long Term Evolution.

MME **Mobility Management Entity.**

MSISDN **Mobile Station International Subscriber Directory Number.**

NW Network.

P-GW PDN Gateway, Packet Data Network Gateway.

PDN **Packet Data Network.**

QoS Quality of Service.

SCTP Stream Control Transmission Protocol.

S-GW Serving Gateway.

SIM Subscriber Identity Module.

TCP Transmission Control Protocol.

USIM Universal Subscriber Identity Module.

1. Introduction
   1. Overview

The EURECOM EPC is a bundle of software components that provides the MME, S+P-GW, HSS functions of the LTE core EPC architecture (<http://www.3gpp.org/DynaReport/23002.htm>).

Actually the SGW and the PGW are merged together, there is no S5 or S8 interface between the two functional entities.



Figure 1 EPC overview

* 1. Deployment scenarios

Two deployment scenarios are considered with the EURECOM EPC.

* + 1. Separate EPC platform

Actually this deployment scenario is under development and cannot be demonstrated yet.



Figure 2 EPC Deployment in MME SP-GW

* + 1. All in one EPC platform

The following picture depicts a EURECOM EPC providing MME and GW functions, and interact with the EURECOM HSS. In this deployment scenario, the S11 interface is virtual in the sense that S11 messages do not go through the network layer but through an inter-task interface message passing middleware (ITTI).



Figure 3 EPC Deployment in MME\_GW

The EPC can be deployed on the same EURECOM eNB host or on its own host.

The HSS can be deployed on the same EPC host, EURECOM eNB host or on its own host. Any combination of deployment with one, two or three host(s) is possible with the EURECOM eNB.

If a third party eNB is used, then it is preferable to run the EPC and HSS on one or two other hosts, indifferently.

1. EPC Installation
   1. Operating system

The EPC software has only been tested on UBUNTU 14.04x64, and UBUNTU 14.10x64 LINUX distributions on Intel x86 64 bits platforms.

If you want to try another LINUX distribution, it is mandatory to have a 64 bits LINUX distribution.

**Important!**

In this document OPENAIRCN\_DIR is the path to the openair-cn working directory.

* 1. EPC source code

The [OpenAirInterface](https://twiki.eurecom.fr/twiki/bin/view/OpenAirInterface/OpenAirInterface) core network software can be obtained from our git server. You will need a git client to get the sources.

If git is not installed on your computer, execute in a shell the following command (Ubuntu):

user@host:~ sudo apt-get install git

Configure git with your name/email address (only important if you are developer and want to checkin code to Git):

git config --global user.name "Your Name"

git config --global user.email "Your email address"

Add a certificate from gitlab.eurecom.fr to your Ubuntu 14.04 installation (you need to be root user):

echo -n | openssl s\_client -showcerts -connect gitlab.eurecom.fr:443 2>/dev/null | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' >> /etc/ssl/certs/ca-certificates.crt

* + 1. Get the code without login

In order to checkout the Git repository (for OAI Users without login to gitlab server)

user@host:~ git clone <https://gitlab.eurecom.fr/oai/openair-cn.git>

* + 1. Get the code with login (contributors)

In order to checkout the Git repository (for OAI Developers/admins with login to gitlab server)

Please send an email to [openair\_tech@eurecom.fr](mailto:openair_tech@eurecom.fr) to be added to the repository as a developer (only important for users who want to commit code to the repository). If you do not have account on gitlab.eurecom.fr, please register yourself to gitlab.eurecom.fr.

**Checkout with using ssh keys:**

You will need to put your ssh keys in https://gitlab.eurecom.fr/profile/keys to access to the git repo. Once that is done, checkout the git repository using:

git clone [git@gitlab.eurecom.fr:oai/openair-cn.git](mailto:git@gitlab.eurecom.fr:oai/openair-cn.git)

**Checkout with user name/password prompt:**

git clone <https://YOUR_USERNAME@gitlab.eurecom.fr/oai/openair-cn.git>

* 1. Additional software, initial steps.

Some software installations have to be done prior to build the EURECOM EPC and the EURECOM HSS.

In OPENAIRCN\_DIR/SCRIPTS directory, execute the following command:

user@host:~/openair-cn/SCRIPTS$ ./build\_epc -i

This command will update the software source list of your Ubuntu installation. It will install miscellaneous software packages, mainly an openair-cn version (patched) of freeDiameter, an openair version (patched) of asn1c, and particularly mysql-server and phpmyadmin software, which steps are described below.

* + 1. Mysql server installation details

Enter here the root password of your host.

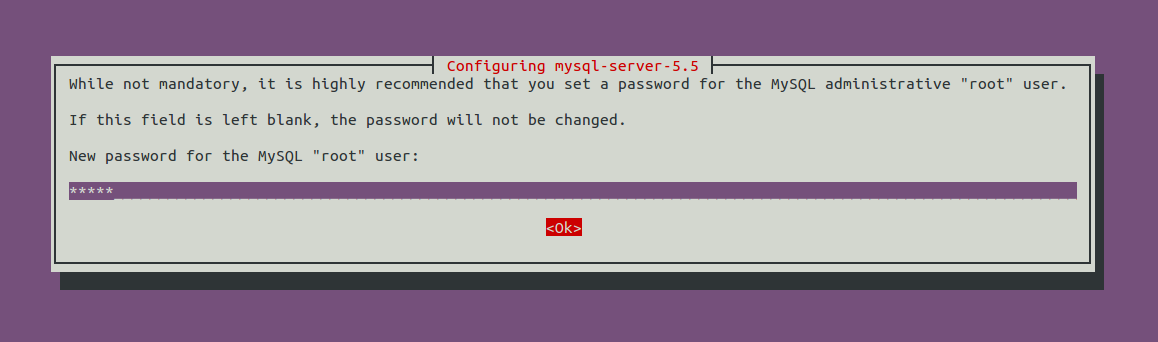
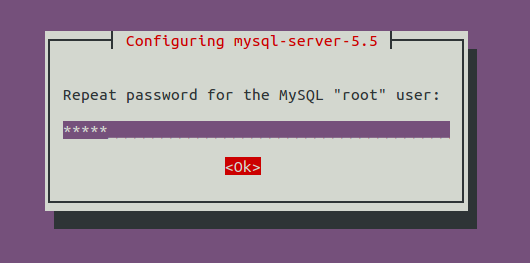


Figure 4 Mysql installation root password



The mysql-server installation process ends here.

* + 1. Phpmyadmin installation details

You should prefer the easiest way

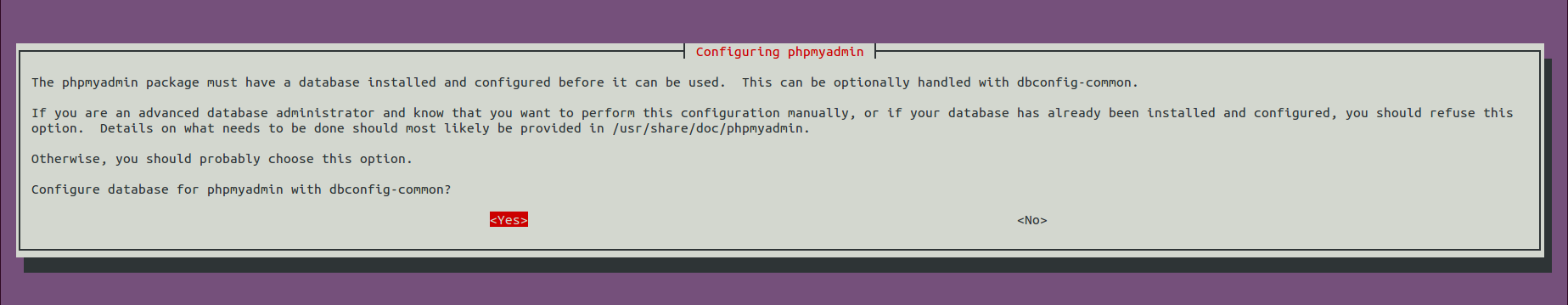


Figure 5 Phpmyadmin installation conf DB

Enter here the root password of your host:

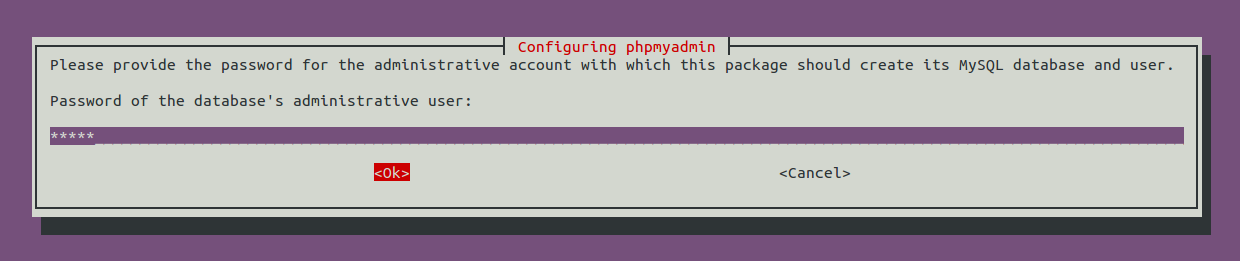


Figure 6 Phpmyadmin installation DB password

Accordingly with the content of openair-cn configuration files, please, enter here admin

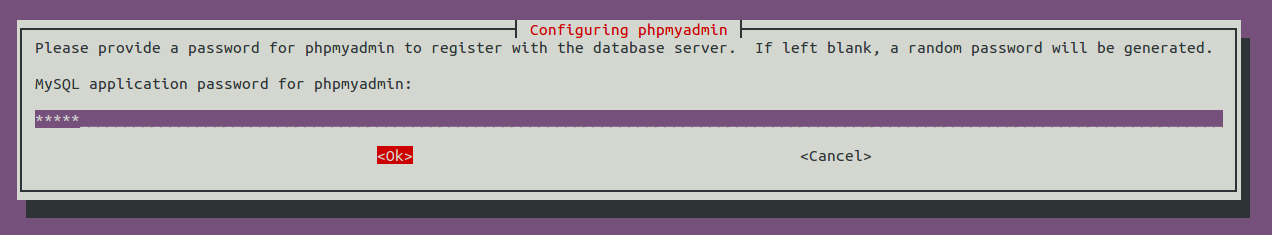
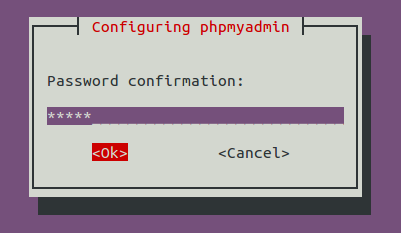


Figure 7 Phpmyadmin installation app password



Choose the web server that has to be configured: Apache.

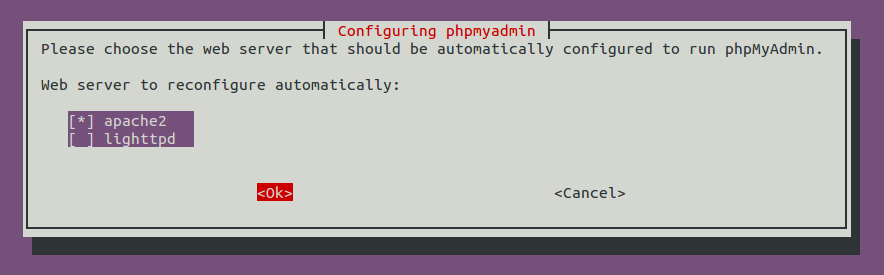


Figure 8 Phpmyadmin installation web server selection

1. EPC Configuration
   1. MME\_GW
      1. Fully Qualified Domain name

A FQDN has to be set for the MME\_GW (freeDiameter constraint ACL about this may not exist anymore, to be checked). An easy way to do that is to fill this FQDN in the /etc/hosts file.

Example:

yang@yang:$ cat /etc/hosts  
127.0.0.1 localhost  
127.0.1.1 **yang.openair4G.eur** yang  
...

192.168.12.175 yin.openair4G.eur hss yin

...

* + 1. Configuration files

Here is view of the build process of MME\_GW, we can see there when and how configuration files are generated. Inputs files and parameters are on the left part of the figure, the build process is in the center part and output configuration files are on the right of the figure.



Figure 9 MME\_GW configuration files generation

**Configuration file epc.conf and epc.local.enb.conf:**

These configuration files, since MME\_GW is an aggregation of a MME, a S-GW and a P-GW, aggregate three configuration sections: a MME, a S-GW, and a P-GW configuration section.

This configuration files follow the libconfig file syntax (<http://www.hyperrealm.com/libconfig>).

These sections are described below.

**Configuration file mme\_fd.conf:**

This configuration file is the input file for configuring the diameter protocol instance of the MME\_GW.

* 1. MME

Empty section, will be updated when a standalone MME will be released.

* 1. SP\_GW

Empty section, , will be updated when a standalone S+P-GW will be released.

* 1. MME configuration content

|  |  |  |
| --- | --- | --- |
| Parameter | Type |  |
| REALM | String | Diameter realm of the MME |
| MAXENB | Num/Integer | Maximum number of eNB that can connect to MME. |
| MAXUE | Num/Integer | For debug purpose, used to restrict the number of served UEs the MME can handle. |
| RELATIVE\_CAPACITY | Num/Integer | Even though this parameter is not used by the MME for controlling the MME load balancing within a pool (at least for now), the parameter has to be forwarded to the eNB during association procedure. Values going from 0 to 255, (Default value is 15) |
| MME\_STATISTIC\_TIMER | Num/Integer | Displayed statistic (stdout) period. |
| EMERGENCY\_ATTACH\_SUPPORTED | String |  |
| UNAUTHENTICATED\_IMSI\_SUPPORTED | String |  |
| IP\_CAPABILITY | String | Choice between IPV4, IPV4V6, IPV4ORV6 |

Table 1 MME configuration main section

* + 1. GUMMEI section

|  |  |  |
| --- | --- | --- |
| Parameter | Type |  |
| MME\_CODE | Array of Num/Integer | List of a maximum of 256 values can be provided. MME code range is [0..255] |
| MME\_GID | Array of Num/Integer | List of maximum 65536 values. MME group id range is [0..65535] |
| TAI | Array of TAI (PLMN:TAC) | List of maximum 32 TAI. (TAI=MCC.MNC:TAC) |

Table 2 MME configuration subsection GUMMEI

* + 1. SCTP section

|  |  |  |
| --- | --- | --- |
| Parameter | Type |  |
| SCTP\_INSTREAMS | Num/Integer | Num streams for UE association signaling, note that stream with id =0 is reserved for non-Ue associated signaling. At least two streams should be used by the MME. (Default value=64) |
| SCTP\_OUTSTREAMS | Num/Integer | Idem above |

Table 3 MME configuration subsection SCTP

* + 1. S1AP section

|  |  |  |
| --- | --- | --- |
| Parameter | Type |  |
| S1AP\_OUTCOME\_TIMER | Num/Integer | Once an outcome is sent from MME to eNB, the MME locally starts a timer to abort the procedure and release UE context if the expected answer to this outcome is not received at the expiry of this timer. This timer is expressed in seconds. (Default value = 5 seconds) |

Table 4 MME configuration subsection S1AP

* + 1. S6A section

|  |  |  |
| --- | --- | --- |
| Parameter | Type |  |
| S6A\_CONF | String | S6A config file path |
| HSS\_HOSTNAME | String | HSS hostname |

Table 5 MME configuration subsection S6a

* + 1. NAS section

|  |  |  |
| --- | --- | --- |
| Parameter | Type |  |
| ORDERED\_SUPPORTED\_INTEGRITY\_ALGORITHM\_LIST | Array of String | Preference list in decreasing order of supported integrity algorithms, actually supported integrity algorithms are EIA0, EIA1, EIA2 |
| ORDERED\_SUPPORTED\_CIPHERING\_ALGORITHM\_LIST | Array of String | Preference list in decreasing order of supported integrity algorithms, actually supported integrity algorithms are EEA0, EEA1, EEA2 |

Table 6 MME configuration subsection NAS

* + 1. INTERTASK\_INTERFACE section

|  |  |  |
| --- | --- | --- |
| Parameter | Type |  |
| ITTI\_QUEUE\_SIZE | Num/Integer | Upper bound for the message queue size expressed in bytes (all messages exchanged by tasks have the same size). Restrict the number of messages in queues or detect a possible MME overload. |

Table 7 MME configuration subsection ITTI

* + 1. Network interfaces section

|  |  |  |
| --- | --- | --- |
| Parameter | Type |  |
| MME\_INTERFACE\_NAME\_FOR\_S1\_MME | String | Interface name for S1-MME (S1-C) |
| MME\_IPV4\_ADDRESS\_FOR\_S1\_MME | String, CIDR | Binded address for S1-MME |
| MME\_INTERFACE\_NAME\_FOR\_S11\_MME | String | Interface name for S11, “none” if S11 unused |
| MME\_IPV4\_ADDRESS\_FOR\_S11\_MME | String, CIDR | Binded address for S11, (0.0.0.0/xx) if S11 unused |

Table 8 MME configuration subsection Network Interfaces

* 1. S-GW configuration content

|  |  |  |
| --- | --- | --- |
| Parameter | Type |  |
| SGW\_INTERFACE\_NAME\_FOR\_S11 | String | Interface name for S11, “none” if S11 unused |
| SGW\_IPV4\_ADDRESS\_FOR\_S11 | String, CIDR notation | Binded address for S11, (0.0.0.0/xx) if S11 unused |
| SGW\_INTERFACE\_NAME\_FOR\_S1U\_S12\_S4\_UP | String | Interface name for S1-U |
| SGW\_IPV4\_ADDRESS\_FOR\_S1U\_S12\_S4\_UP | String, CIDR notation | Binded address for S1-U |
| SGW\_IPV4\_PORT\_FOR\_S1U\_S12\_S4\_UP | Num/Integer | Port number for S1-U (IANA), Should be 2152 |
| SGW\_INTERFACE\_NAME\_FOR\_S5\_S8\_UP | String, | Interface name for S5 or S8, “none” because unused |
| SGW\_IPV4\_ADDRESS\_FOR\_S5\_S8\_UP | String, CIDR notation | Binded address for S5 or S8, (0.0.0.0/xx) because unused |

Table 9 S-GW configuration main section

* 1. P-GW configuration content
     1. Main section

|  |  |  |
| --- | --- | --- |
| Parameter | Type |  |
| DEFAULT\_DNS\_1\_IPV4\_ADDRESS | String, IPv4 dot decimal | IPv4 address of primary default DNS that can be queried by UEs |
| DEFAULT\_DNS\_2\_IPV4\_ADDRESS | String, IPv4 dot decimal | IPv4 address of secondary default DNS that can be queried by UEs |

Table 10 P-GW configuration main section

* + 1. Network interfaces section

|  |  |  |
| --- | --- | --- |
| Parameter | Type |  |
| PGW\_INTERFACE\_NAME\_FOR\_S5\_S8 | String | Interface name for S5 or S8, “none” because unused |
| PGW\_IPV4\_ADDRESS\_FOR\_S5\_S8 | String, CIDR notation | Binded address for S5 or S8, (0.0.0.0/xx) because unused |
| PGW\_INTERFACE\_NAME\_FOR\_SGI | String | Interface name for SGi |
| PGW\_IPV4\_ADDRESS\_FOR\_SGI | String, CIDR notation | Used IPv4 address for SGi, useful if UE traffic is masqueraded. |
| PGW\_MASQUERADE\_SGI | String | Should outgoing UE IPv4 traffic be masqueraded (source NAT), “yes” or “no”. |

Table 11 P-GW configuration subsection Network Interfaces

* + 1. IP Address Pool section

|  |  |  |
| --- | --- | --- |
| Parameter | Type |  |
| IPV4\_LIST | String, CIDR notation | List of IPv4 netmasks that designate a list of available IPv4 addresses for UEs |
| IPV6\_LIST | String, CIDR notation | List of IPv6 netmasks that designate a list of available IPv6 addresses for UEs |

Table 12 P-GW configuration subsection IP Address Pool Selection

* 1. HSS
     1. Fully Qualified Domain name

A FQDN has to be set for the HSS. An easy way to do that is to fill this FQDN in the /etc/hosts file.

Example:

yin@yin:$ cat /etc/hosts  
127.0.0.1 localhost  
127.0.1.1 **yin.openair4G.eur** yin  
...

* + 1. Configuration files

Here is partial view of the build process of HSS, we can see there when and how configuration files are generated. Inputs files and parameters are on the left part of the figure, the build process is in the center part and output configuration files are on the right of the figure.



Figure 10 HSS configuration files generation

**Configuration file hss.conf.in:**

This configuration file is the top configuration file containing all necessary parameters and links to other configuration files. This file do not need to be edited, all parameters passed to the build\_hss executable and also its default parameters are substituted in the right place in this config file.

hss.conf.in content:

## MySQL mandatory options

MYSQL\_server = "@MYSQL\_server@";

MYSQL\_user = "@MYSQL\_user@";

MYSQL\_pass = "@MYSQL\_pass@";

MYSQL\_db = "@MYSQL\_db@";

## HSS options

OPERATOR\_key = "@OPERATOR\_key@";

RANDOM = "@RANDOM\_boolean@";

## Freediameter options

FD\_conf = "@FREEDIAMETER\_PATH@/../etc/freeDiameter/hss\_fd.conf";

The following is an example of the resulting config file hss.conf:

## MySQL mandatory options

MYSQL\_server = "127.0.0.1";

MYSQL\_user = "hssadmin";

MYSQL\_pass = "admin";

MYSQL\_db = "oai\_db";

## HSS options

OPERATOR\_key = "11111111111111111111111111111111";

RANDOM = "FALSE";

## Freediameter options

FD\_conf = "/usr/lib/../etc/freeDiameter/hss\_fd.conf";

**Configuration file hss\_fd.conf.in:**

This configuration file is the input file for configuring the diameter protocol instance of the HSS.

All parameters values between ‘@’ are filled by the cmake process. These parameters are set with the help of input parameters passed to the build\_hss executable, and with the help of default values set in the OPENAIRCN\_DIR/BUILD/HSS/CMakeLists.txt file.

You can see here what are default values defined in OPENAIRCN\_DIR/BUILD/HSS/CMakeLists.txt and set your own:

set(MYSQL\_server "127.0.0.1" CACHE STRING "Database server IP address")

set(MYSQL\_admin root CACHE STRING "Database admin login")

set(MYSQL\_admin\_pass linux CACHE STRING "Database admin password")

set(MYSQL\_user hssadmin CACHE STRING "Database username login")

set(MYSQL\_pass admin CACHE STRING "Database username password")

set(MYSQL\_db oai\_db CACHE STRING "Database name")

set(TRANSPORT\_option "#No\_TCP" CACHE STRING "No\_TCP or No\_SCTP or comment string, FreeDiameter config option")

set(TRANSPORT\_PREFER\_TCP\_option "#Prefer\_TCP" CACHE STRING "Prefer\_TCP or comment string, FreeDiameter config option")

set(AppServThreads 2 CACHE STRING "FreeDiameter AppServThreads config option")

set(OPERATOR\_key "" CACHE STRING "LTE operator clear text key (hex bytes) example 11111111111111111111111111111111")

set(RANDOM\_boolean "true" CACHE STRING "If false, random function returns always 0, else random as usual.")

set(REMOTE\_PEER\_WHITELIST "\*.${REALM}" CACHE STRING "Remote peer whitelist (separated by spaces), for freediameter acl.conf config file")

hss\_fd.conf.in content:

# -------- Local ---------

# The first parameter in this section is Identity, which will be used to

# identify this peer in the Diameter network. The Diameter protocol mandates

# that the Identity used is a valid FQDN for the peer. This parameter can be

# omitted, in that case the framework will attempt to use system default value

# (as returned by hostname --fqdn).

Identity = "@HSS\_FQDN@";

# In Diameter, all peers also belong to a Realm. If the realm is not specified,

# the framework uses the part of the Identity after the first dot.

Realm = "@REALM@";

# This parameter is mandatory, even if it is possible to disable TLS for peers

# connections. A valid certificate for this Diameter Identity is expected.

TLS\_Cred = "@FREEDIAMETER\_PATH@/../etc/freeDiameter/hss.cert.pem", "@FREEDIAMETER\_PATH@/../etc/freeDiameter/hss.key.pem";

TLS\_CA = "@FREEDIAMETER\_PATH@/../etc/freeDiameter/hss.cacert.pem";

# Disable use of TCP protocol (only listen and connect in SCTP)

# Default : TCP enabled

@TRANSPORT\_option@;

# This option is ignored if freeDiameter is compiled with DISABLE\_SCTP option.

# Prefer TCP instead of SCTP for establishing new connections.

# This setting may be overwritten per peer in peer configuration blocs.

# Default : SCTP is attempted first.

@TRANSPORT\_PREFER\_TCP\_option@;

# Disable use of IPv6 addresses (only IP)

# Default : IPv6 enabled

No\_IPv6;

# Overwrite the number of SCTP streams. This value should be kept low,

# especially if you are using TLS over SCTP, because it consumes a lot of

# resources in that case. See tickets 19 and 27 for some additional details on

# this.

# Limit the number of SCTP streams

SCTP\_streams = 3;

# By default, freeDiameter acts as a Diameter Relay Agent by forwarding all

# messages it cannot handle locally. This parameter disables this behavior.

NoRelay;

TLS\_old\_method;

# Number of parallel threads that will handle incoming application messages.

# This parameter may be deprecated later in favor of a dynamic number of threads

# depending on the load.

AppServThreads = @AppServThreads@;

# Specify the addresses on which to bind the listening server. This must be

# specified if the framework is unable to auto-detect these addresses, or if the

# auto-detected values are incorrect. Note that the list of addresses is sent

# in CER or CEA message, so one should pay attention to this parameter if some

# adresses should be kept hidden.

@ListenOn@;

@DIAMETER\_PORT@;

@DIAMETER\_SEC\_PORT@;

# -------- Extensions ---------

# Uncomment (and create rtd.conf) to specify routing table for this peer.

#LoadExtension = "rt\_default.fdx" : "rtd.conf";

# Uncomment (and create acl.conf) to allow incoming connections from other peers.

LoadExtension = "acl\_wl.fdx" : "@FREEDIAMETER\_PATH@/../etc/freeDiameter/acl.conf";

# Uncomment to display periodic state information

#LoadExtension = "dbg\_monitor.fdx";

# Uncomment to enable an interactive Python interpreter session.

# (see doc/dbg\_interactive.py.sample for more information)

#LoadExtension = "dbg\_interactive.fdx";

# Load the RFC4005 dictionary objects

#LoadExtension = "dict\_nasreq.fdx";

LoadExtension = "dict\_nas\_mipv6.fdx";

LoadExtension = "dict\_s6a.fdx";

# Load RFC4072 dictionary objects

#LoadExtension = "dict\_eap.fdx";

# Load the Diameter EAP server extension (requires diameap.conf)

#LoadExtension = "app\_diameap.fdx" : "diameap.conf";

# Load the Accounting Server extension (requires app\_acct.conf)

#LoadExtension = "app\_acct.fdx" : "app\_acct.conf";

# -------- Peers ---------

# The framework will actively attempt to establish and maintain a connection

# with the peers listed here.

# For only accepting incoming connections, see the acl\_wl.fx extension.

#ConnectPeer = "ubuntu.localdomain" { ConnectTo = "127.0.0.1"; No\_TLS; };

@ConnectPeer@ = "@MME\_FQDN@" { ConnectTo = "@MME\_IP@"; Realm = "@REALM@"; No\_IPv6; No\_TLS ; port = 3870; };

**Configuration file acl.conf.in:**

TODO

* + 1. HSS database content

SQL operations (display, update, export, etc) can be done easily with the help of phpMyAdmin, you have to open the following URL with your browser: <http://yourhsshost/phpmyadmin>.

Otherwise you can use any other MySQL tool, script compatible with MySQL.

**Table mmeidentity:**

Structure:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| idmmeidentity | int(11) | NO | PRI | NULL | auto\_increment |
| mmehost | varchar(255) | YES |  | NULL |  |
| mmerealm | varchar(200) | YES |  | NULL |  |
| UE-Reachability | tinyint(1) | NO |  | NULL |  |

Table 13 SQL Table structure mmeidentity

Column idmmeIdentity is the primary key of a MME.

Column mmehost contains the FQDN of a MME.

Column mmerealm contains the realm of a MME.

Example of content:

+---------------+------------------------+---------------+-----------------+  
| idmmeidentity | mmehost                | mmerealm      | UE-Reachability |  
+---------------+------------------------+---------------+-----------------+  
|             2 | yang.openair4G.eur     | openair4G.eur |               0 |  
|             1 | ng40-erc.openair4G.eur | openair4G.eur |               0 |  
|             3 | ABEILLE.openair4G.eur  | openair4G.eur |               0 |  
+---------------+------------------------+---------------+-----------------+

**Table pdn:**

This table contains mainly the association between a user and a APN, and its QOS parameters.

Structure:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| id | int(11) | NO | PRI | NULL | auto\_increment |
| apn | varchar(60) | NO |  | NULL |  |
| pdn\_type | enum('IPv4','IPv6','IPv4v6','IPv4\_or\_IPv6') | NO |  | NULL |  |
| pdn\_ipv4 | varchar(15) | YES |  | NULL | 0.0.0.0 |
| pdn\_ipv6 | varchar(45) | YES |  | NULL | 0:0:0:0:0:0:0:0 |
| aggregate\_ambr\_ul | int(10) unsigned | YES |  | 50000000 |  |
| aggregate\_ambr\_dl | int(10) unsigned | YES |  | 100000000 |  |
| pgw\_id | int(11) | NO | PRI | NULL |  |
| users\_imsi | varchar(15) | NO | PRI |  |  |
| qci | tinyint(3) unsigned | NO |  | 9 |  |
| priority\_level | tinyint(3) unsigned | NO |  | 15 |  |
| pre\_emp\_cap | enum('ENABLED','DISABLED') | YES |  | DISABLED |  |
| pre\_emp\_vul | enum('ENABLED','DISABLED') | YES |  | DISABLED |  |
| LIPA-Permissions | enum('LIPA-prohibited','LIPA-only','LIPA-conditional') | YES |  | LIPA-only |  |

Table 14 SQL Table structure pdn

Column id is the primary key of a pdn entry.

Column pdn\_type contains the type of PDN, actually only IPv4 is supported.

Column pdn\_ipv4 contains the IPv4 address of the PDN (unused).

Column pdn\_ipv6 contains the IPv6 address of the PDN (unused).

Column aggregate\_ambr\_ul TODO

Column aggregate\_ambr\_dl TODO

Column pgw\_id TODO

Column users\_imsi TODO

Column qci TODO

Column priority\_level TODO

Column pre\_emp\_capability TODO

Column pre\_emp\_vulnerability TODO

Column LIPA\_Permissions TODO

**Table users**

Structure:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| imsi | varchar(15) | NO | PRI | NULL |  |
| msisdn | varchar(46) | YES |  | NULL |  |
| imei | varchar(15) | YES |  | NULL |  |
| imei\_sv | varchar(2) | YES |  | NULL |  |
| ms\_ps\_status | enum('PURGED','NOT\_PURGED') | YES |  | PURGED |  |
| rau\_tau\_timer | int(10) unsigned | YES |  | 120 |  |
| ue\_ambr\_ul | bigint(20) unsigned | YES |  | 50000000 |  |
| ue\_ambr\_dl | bigint(20) unsigned | YES |  | 100000000 |  |
| access\_restriction | int(10) unsigned | YES |  | 60 |  |
| mme\_cap | int(10) unsigned zerofill | YES |  | NULL |  |
| mmeidentity\_idmmeidentity | int(11) | NO | PRI | 0 |  |
| key | varbinary(16) | NO |  | 0 |  |
| RFSP-Index | smallint(5) unsigned | NO |  | 1 |  |
| urrp\_mme | tinyint(1) | NO |  | 0 |  |
| sqn | bigint(20) unsigned zerofill | NO |  | NULL |  |
| rand | varbinary(16) | NO |  | NULL |  |
| OPc | varbinary(16) | YES |  | NULL |  |

Table 15 SQL Table structure users

TODO column description.

* + 1. Configuring your HSS

#### Adding your MME

With the help of phpmyadmin: in your database (default is oai\_db), in table mmeidentity add your MME:

mmeidentity.idmmeidentity= your MME new key (unique id in 1..N)

mmeidentity.mmehost= your MME fqdn

mmeidentity.mmerealm= your MME realm (should be your MME FQDN without the host name)

mmeidentity.UE-Reachability= 0

#### Adding a user

In table users add your user informations:

user.imsi=IMSI of your USIM.

user.msisdn= MSISDN of your USIM (unused).

users.imei=NULL

users.imei\_sv=NULL

users.ms\_ps\_status=’PURGED’

users.rau\_tau\_timer=120

users.ue\_ambr\_ul=50000000

users.ue\_ambr\_dl=100000000

users.access\_restriction=47

users.mme\_cap=0

users.mmeidentity\_idmmeidentity=’your MME key’

users.RFSP-Index=1

users.urrp\_mme=0

users.sqn=’your USIM programmed SQN’

users.rand=0

users.OPc=’the OPc key’ (can be computed by the oai\_hss executable when using –k option in build\_hss script)

In table pdn allow your user to be served by a APN: Insert a new record and fill all column like other records except for users\_imsi column.

1. Building and running

The EURECOM EPC interact mainly with two other entities: the eNB and the HSS. Depending on the location of these entities on the same host or not, the building and running options differ:

* When EPC and HSS run on the same host, TCP must be selected as the underlying protocol for DIAMETER on the S6a interface. If EPC and HSS run on separate hosts, SCTP can be selected as the underlying protocol for DIAMETER on the S6a interface. Choosing SCTP instead of TCP makes the network capture of S1-MME traffic easier.
* Depending if EPC and EURECOM eNB run on the same host or not, for convenience, two different configuration files are provided, one for each situation.

We recommend to follow the step described below, unless you know what you are doing.

* 1. MME\_GW.

Your EURECOM MME\_GW host and your EURECOM HSS host (may be the same host)

* + 1. Configuration files

Configuration files have to be filled prior to compilation.

If the MME\_GW and the eNB run on the same host, fill OPENAIRCN\_DIR/BUILD/EPC/epc.local.enb.conf.in configuration file, else fill OPENAIRCN\_DIR/BUILD/EPC/epc.conf.in configuration file.

* + 1. Building EPC

In a shell go to your OPENAIRCN\_DIR/SCRIPTS directory:

If MME\_GW and the HSS run on the same host, execute the following commands:

user@mmegwhost:~/openair-cn/SCRIPTS$ ./build\_epc --debug --local-hss (optional parameter --clean)

user@mmegwhost:~/openair-cn/SCRIPTS$ ./build\_hss --debug –-local-mme (optional parameters: --clean –-operator-key 11111111111111111111111111111111 for example)

Else, execute the following command:

* On MME\_GW host:

user@mmegwhost:~/openair-cn/SCRIPTS$ ./build\_epc --debug --remote-hss ***yourhssfqdn*** (optional parameter --clean)

* On HSS host:

user@hsshost:~/openair-cn/SCRIPTS$ ./build\_hss --debug (optional parameters: --clean –-operator-key 11111111111111111111111111111111 for example)

* + 1. Running EPC

In a shell go to your OPENAIRCN\_DIR/SCRIPTS directory:

If MME\_GW and the eNB run on the same host, execute the following commands:

On MME\_GW host or HSS host:

user@host:~/openair-cn/SCRIPTS$ ./run\_hss

On MME\_GW host:

user@mmegwhost:~/openair-cn/SCRIPTS$ ./run\_epc –l

Else, execute the following command:

On MME\_GW host or HSS host:

user@hsshost:~/openair-cn/SCRIPTS $ ./run\_hss

On MME\_GW host:

user@mmegwhost:~/openair-cn/SCRIPTS $ ./run\_epc

Have a look at all these executables options (-h option)

1. Supported scenarios in EPC
   1. E-UTRAN Initial attach
      1. Attach with IMSI

TBD

* + 1. Attach with GUTI

TBD

* 1. Tracking Area Update procedures

TBD

* 1. Routing Area Update procedures

Not supported yet.

* 1. Service Request procedures
     1. UE triggered Service Request

TBD

* + 1. Network triggered Service Request

Not supported yet.

* 1. S1 Release procedure

TBD

* 1. GUTI Reallocation procedure
  2. Detach procedure
     1. UE-Initiated Detach procedure for E-UTRAN
     2. MME-Initiated Detach procedure for E-UTRAN
     3. HSS-Initiated Detach procedure for E-UTRAN

Not supported.

* 1. HSS User Profile management function procedure

Not supported.

* 1. Bearer deactivation
     1. PDN GW initiated bearer deactivation

Not supported

* + 1. MME initiated Dedicated Bearer Deactivation

TBD

* 1. Intra E-UTRAN handover

Not supported yet

1. Annex A: Tools for observing, debugging.
   1. Itti\_analyzer

Itti\_analyzer takes a dump of messages exchanges between the executable (mme\_gw or eNB, UE) tasks as input and display these messages in a human readable and comprehensible way. This tool can take as input a file whose content is the XML dump of ITTI messages exchanged between tasks or can act as a server and listen on a socket that a openair executable connects and dump messages in pseudo real-time. Trace messages are also displayed with the tool, but in a second view, that means not interlaced with ITTI messages.

**Important:**

Prior to use itti\_analyzer, you have to instruct the openair-cn executable to dump the ITTI messages to a file with the argument –K *path\_to\_file.*

* + 1. Installation

In OPENAIRCN\_DIR/COMMON/ITTI\_ANALYZER directory, execute the following command:

user@host:~ autoreconf –i

user@host:~ ./configure

user@host:~ make

user@host:~ sudo make install

The itti\_analyzer executable is now installed on the computer (/usr/local/bin)

* + 1. Execution

In a shell, execute the following command:

user@host:~ itti\_analyzer

The GUI displayed:

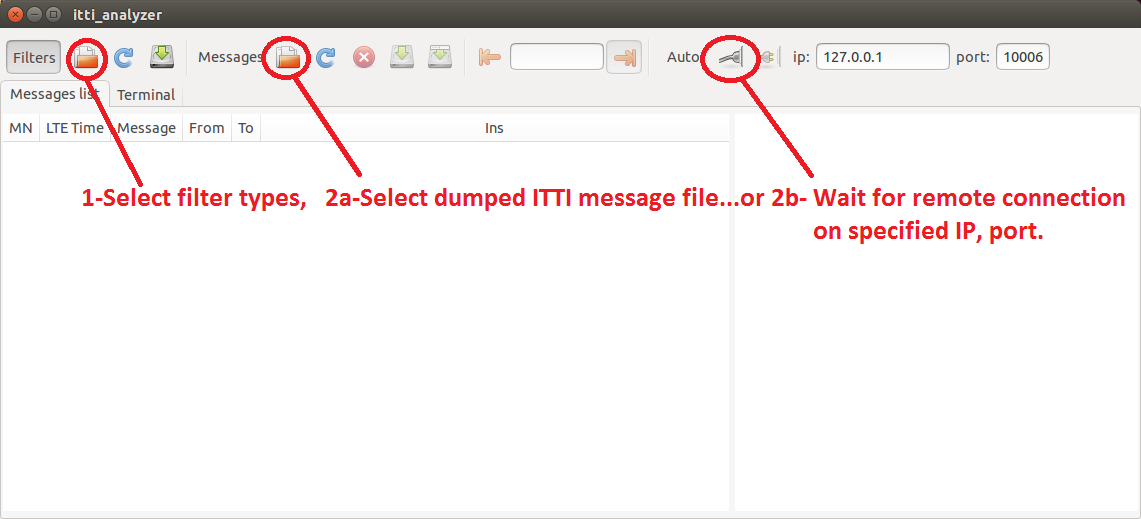


Figure 11 ITTI Analyzer main window

For filter selection, please use filters\_mme.xml:

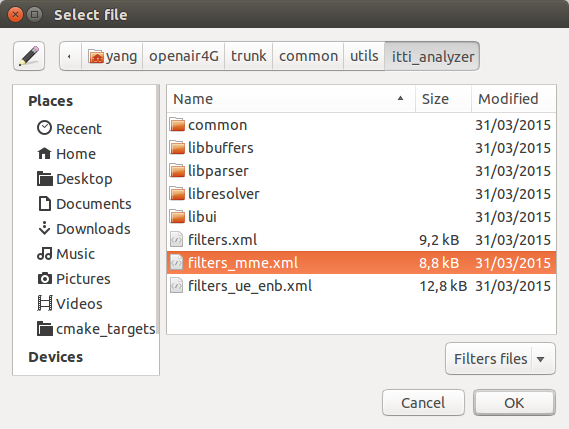


Figure 12 ITTI Analizer select filter menu

You can also use options for fastest operations:

user@host:~itti\_analyzer -h  
Usage: itti\_analyser [options]  
  
Options:  
  -d DISSECT   write DISSECT file with message types parse details  
  -f FILTERS   read filters from FILTERS file  
  -h           display this help and exit  
  -i IP        set ip address to IP  
  -l LEVEL     set log level to LEVEL in the range of 2 to 7  
  -m MESSAGES  read messages from MESSAGES file  
  -p PORT      set port to PORT

* 1. Wireshark/tshark

You can launch wireshark instances on S1 (filter s1ap, gtpu), S6A (filter diameter, if TCP is the undelying protocol, you can select a TCP packet relative to the DIAMETER exchange and the select decode as DIAMETER).

* 1. Mscgen

Extract from <http://www.mcternan.me.uk/mscgen/>: “Mscgen is a small program that parses Message Sequence Chart descriptions and produces PNG, SVG, EPS or server side image maps (ismaps) as the output. Message Sequence Charts (MSCs) are a way of representing entities and interactions over some time period”…” Mscgen aims to provide a simple text language that is clear to create, edit and understand, which can also be transformed into common image formats for display or printing.”…

Openair use mscgen to offer another view of events (SDUs, timers, etc) that happens inside an executable and also (still under development) PDUs exchanged between protocol entities.

Openair HSS do not have the msgen feature.

**Important**:

Check that mscgen traces are configured for being generated (CFLAG MESSAGE\_CHART\_GENERATOR set to true in OPENAIRCN\_DIR/BUILD/EPC/CMakeLists.template)

You have to instruct the openair mme\_gw executable to dump the ITTI messages to a file with the argument -m *path\_to\_directory*. The mscgen files will be located under the specified directory, in a directory containing the time of the generated traces (text and png files).

Example:

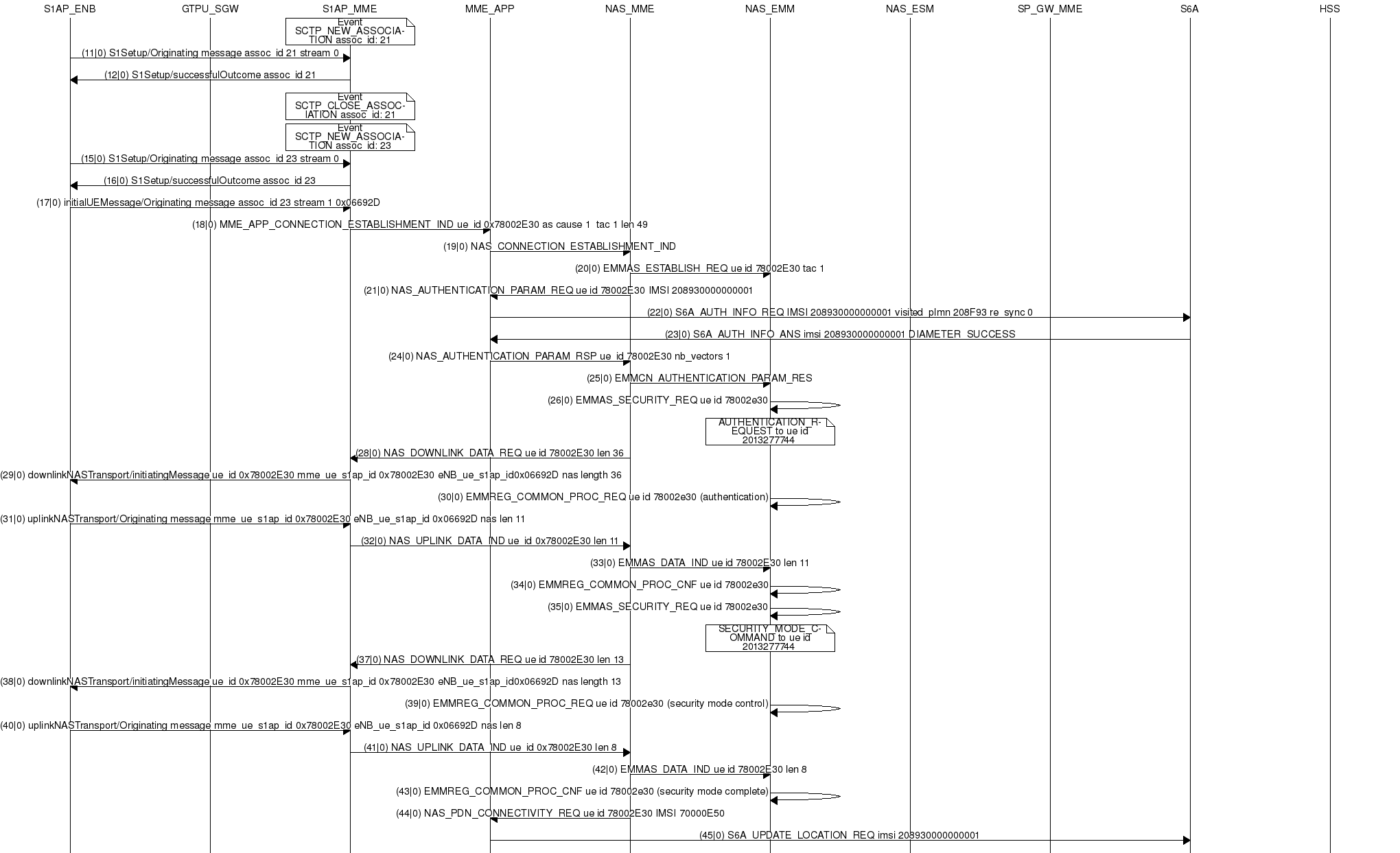
**

Figure 13 Mscgen output example

* 1. S1AP scenario replay

(Not released, under development)

The aim of this tool is for debug purpose when replaying a scenario is needed, it can also be used for non-regression tests.

This tool takes as input the pcap record of S1AP exchanges between eNB and MME, and also some records of the HSS database, then generate C code that replays the scenario.

To make this possible it is necessary to configure the HSS not to randomize the keys (build\_hss *[your options]* --random no).

Steps:

* + 1. Capture a scenario

1. Configure your EPC environment.
2. Start a pcap capture of s1ap protocol on S1 interface with wireshark or tshark.
3. Capture a snapshot of the database with the tool xxxx
4. Play the scenario with the EPC, eNB(s), UE(s).
5. Save the pcap trace and the snapshot of the database.
   * 1. build a S1AP scenario from pcap trace

TODO

* + 1. Replay a S1AP scenario

TODO.