EPC User's Guide

Editor:	EURECOM
Deliverable nature:	Public
Due date:	July, 2015
Delivery date:	July, 2015
Version:	0.7
Total number of pages:	37
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 testing 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	29/07/2015	0.4	Minor corrections (build, get sources from git)
Lionel GAUTHIER	09/09/2015	0.5	Minor corrections (build_epc script)
Lionel GAUTHIER	21/09/2015	0.6	History, GTP-U installation for EPC
Lionel GAUTHIER	14/12/2015	0.7	Removed epc config file when eNB and EPC are co-located, added description of S1AP scenario replay

Table of Contents

List of Figures

List of tables

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

1.1 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.

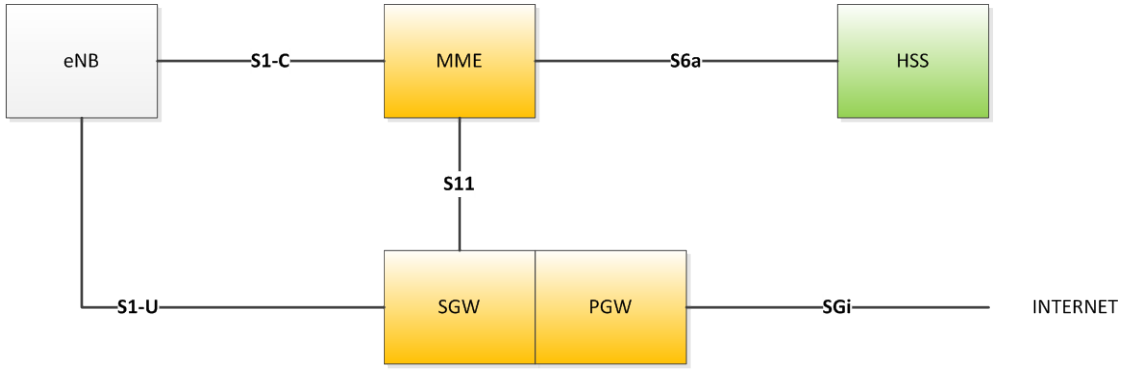


Figure 1 EPC overview

1.2 Deployment scenarios

Two deployment scenarios are considered with the EURECOM EPC.

1.2.1 Separate EPC platform

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

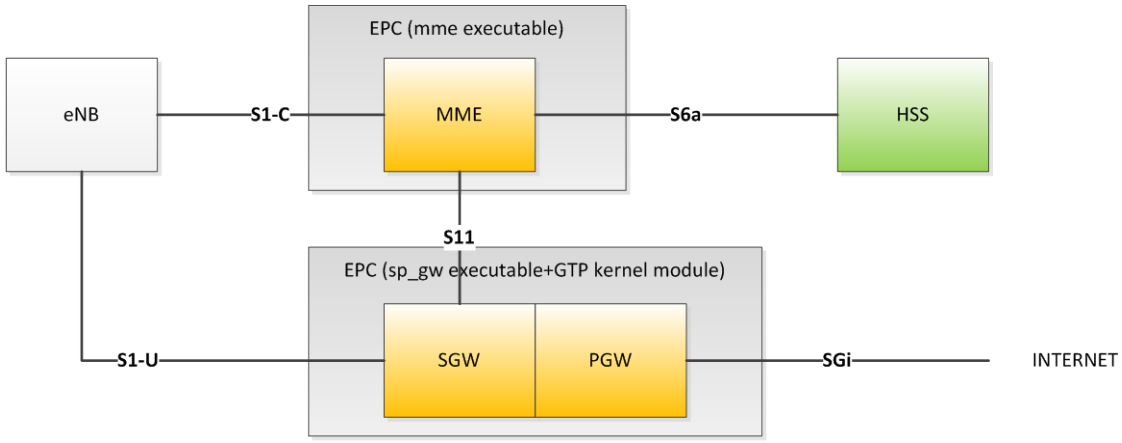


Figure 2 EPC Deployment in MME SP-GW

1.2.2 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).

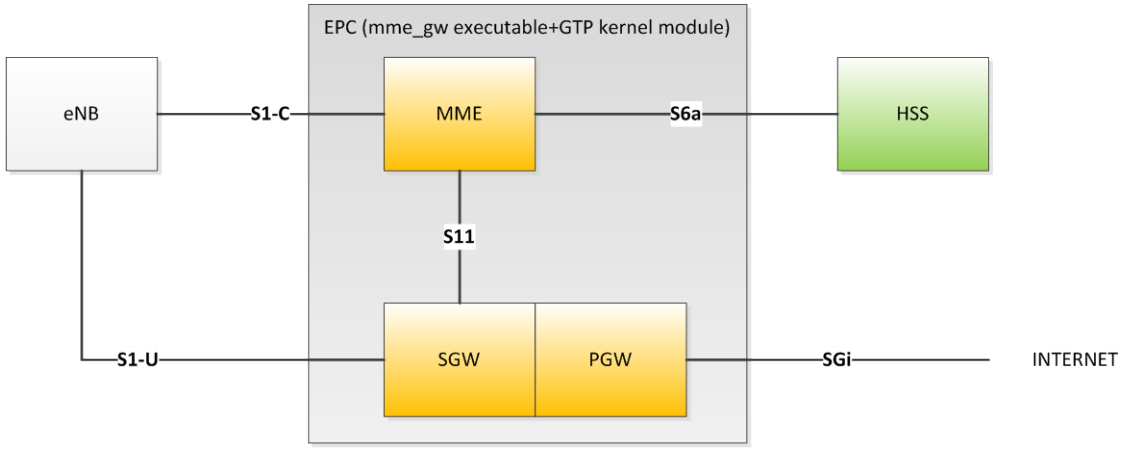


Figure 3 EPC Deployment in MME_GW

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.

2 EPC Installation

2.1 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.

2.2 EPC source code

The 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
```

2.2.1 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`

2.2.2 Get the code with login (contributors)

In order to check out the Git repository (for OAI Developers/admins with login to gitlab server)
Please send an email to 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
```

Checkout with user name/password prompt:
`git clone https://YOUR_USERNAME@gitlab.eurecom.fr/oai/openair-cn.git`

2.3 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.

This command will also install the GTP-U kernel part of the GTP-U protocol layer of the S-GW. For licensing reasons, this code is located in another git repository: git@gitlab.eurecom.fr:oai/xtables-addons-oai.git.

This kernel module requires the installation of a kernel version greater or equal to 3.19, this constraint is handled in the command `build_epc -I, or build_epc -j` (install only the xtables-addons-oai.git software): you are prompted if necessary to install a linux kernel and/or reboot your operating system on another linux kernel.

2.3.1 Mysql server installation details

Enter here the root password of your host.

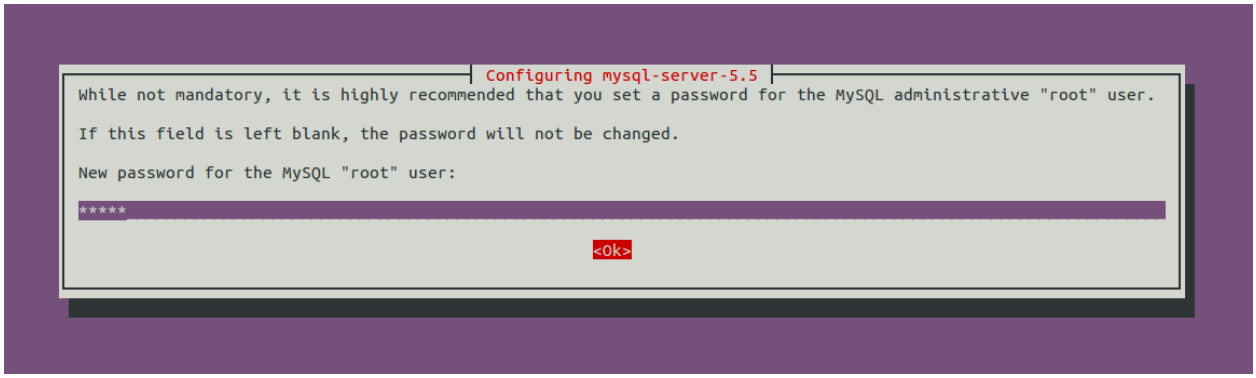
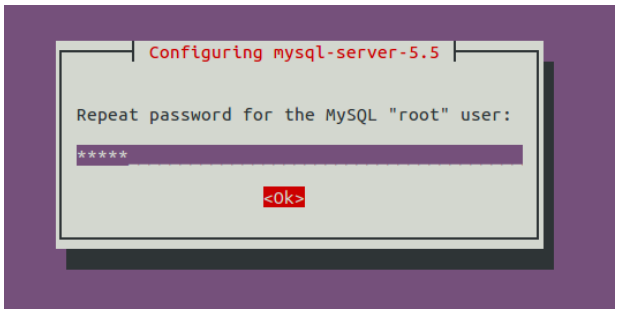


Figure 4 Mysql installation root password



The mysql-server installation process ends here.

2.3.2 Phpmyadmin installation details

You should prefer the easiest way

Figure 5 Phpmyadmin installation conf DB

Enter here the root password of your host:

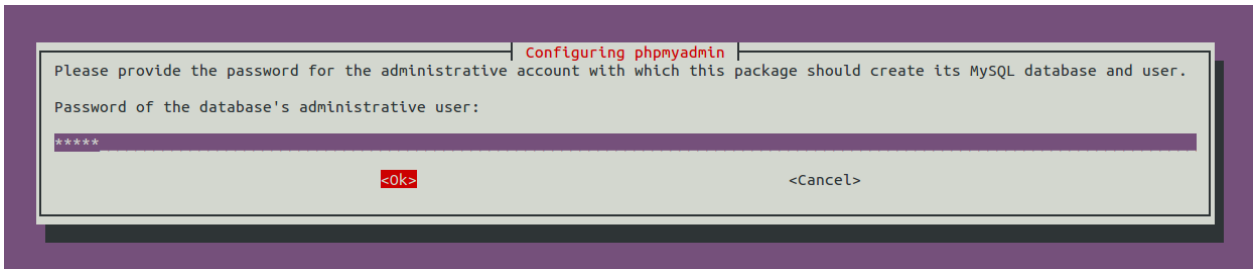


Figure 6 Phpmyadmin installation DB password

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

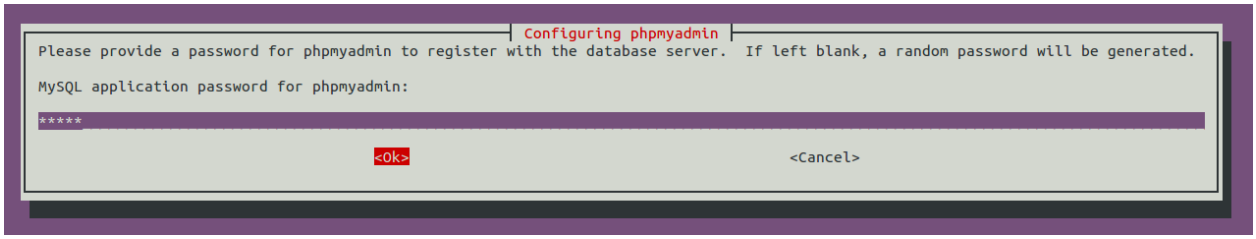
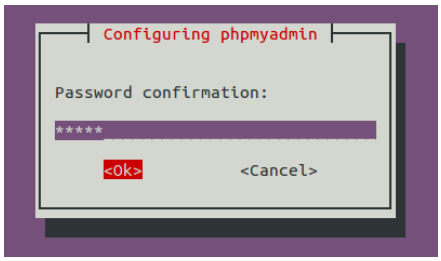


Figure 7 Phpmyadmin installation app password



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

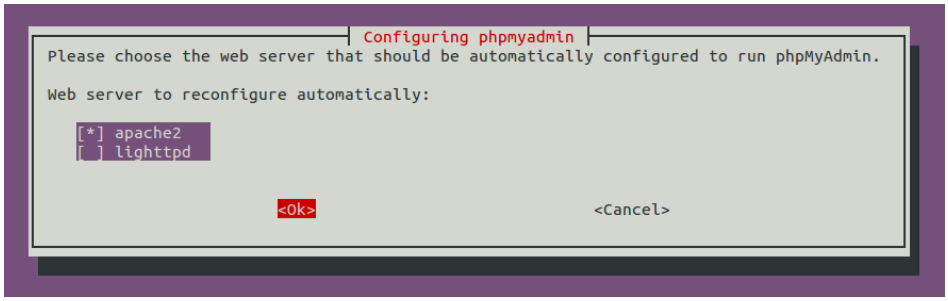


Figure 8 Phpmyadmin installation web server selection

3 EPC Configuration

3.1 MME_GW

3.1.1 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

...
```

3.1.2 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.

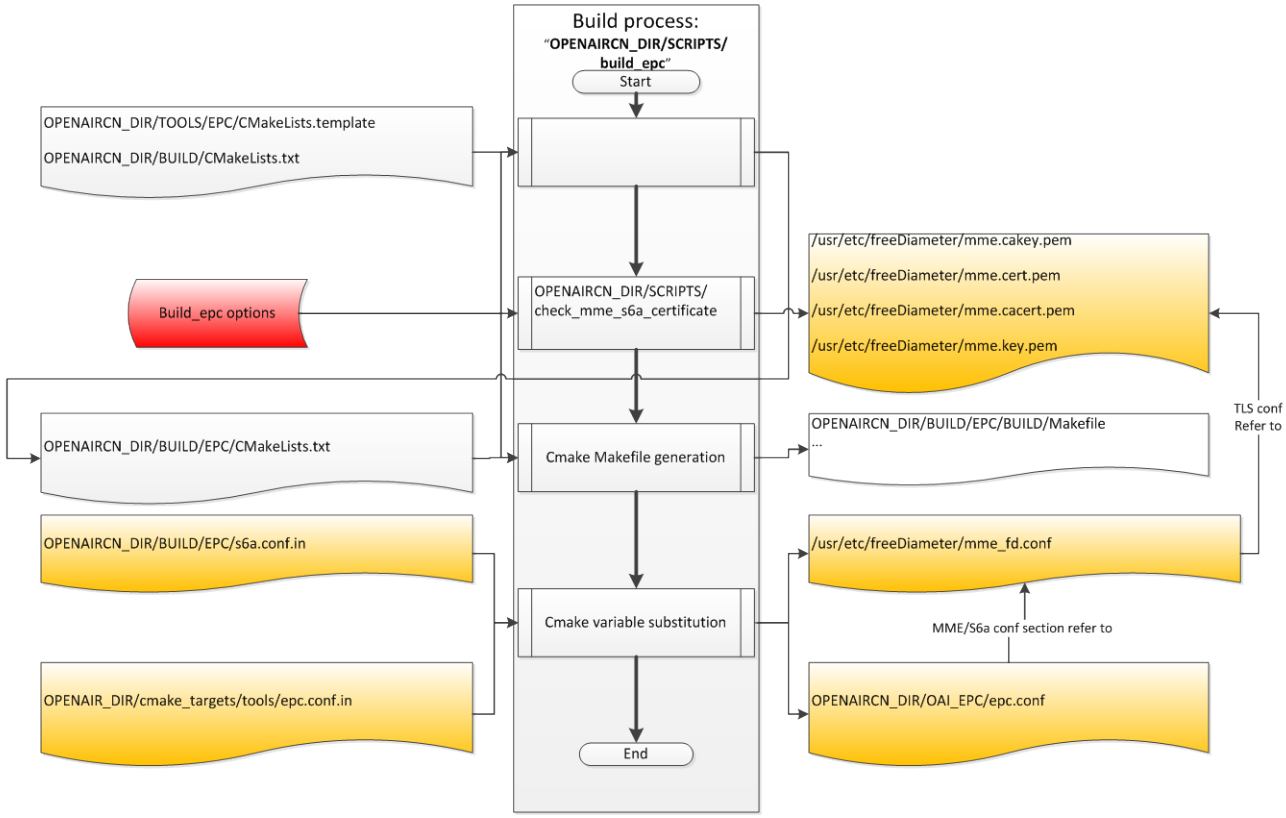


Figure 9 MME_GW configuration files generation

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.

3.2 MME

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

3.3 SP_GW

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

3.4 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

3.4.1 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

3.4.2 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

3.4.3 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

3.4.4 S6A section

Parameter	Type	
S6A_CONF	String	S6A config file path
HSS_HOSTNAME	String	HSS hostname

Table 5 MME configuration subsection S6a

3.4.5 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

3.4.6 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

3.4.7 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

3.5 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

3.6 P-GW configuration content

3.6.1 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

3.6.2 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

3.6.3 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

3.7 HSS

3.7.1 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
...
```

3.7.2 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.

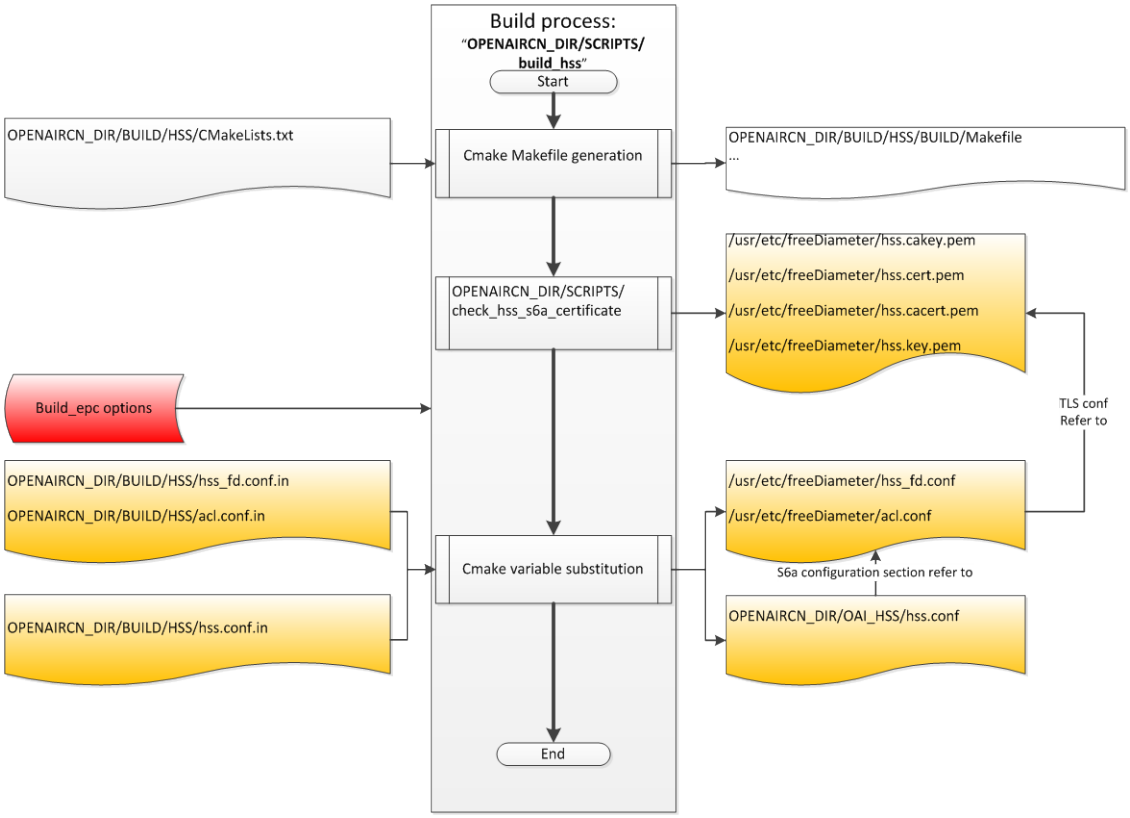


Figure 10 HSS configuration files generation

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

3.7.3 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.

3.7.4 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.

4 Building and running

The EURECOM EPC interact mainly with two other entities: the eNB and the HSS. Depending on the location of the HSS entity, 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.

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

4.1 MME_GW.

Your EURECOM MME_GW host and your EURECOM HSS host (may be the same host)

4.1.1 Configuration files

Configuration files have to be filled prior to compilation.

Fill OPENAIRCN_DIR/BUILD/EPC/epc.conf.in configuration file.

4.1.2 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 --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 --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)

4.1.3 Running EPC

In a shell go to your OPENAIRCN_DIR/SCRIPTS directory:

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
```

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

5 Supported scenarios in EPC

5.1 E-UTRAN Initial attach

5.1.1 Attach with IMSI

TBD

5.1.2 Attach with GUTI

TBD

5.2 Tracking Area Update procedures

TBD

5.3 Routing Area Update procedures

Not supported yet.

5.4 Service Request procedures

5.4.1 UE triggered Service Request

TBD

5.4.2 Network triggered Service Request

Not supported yet.

5.5 S1 Release procedure

TBD

5.6 GUTI Reallocation procedure

5.7 Detach procedure

5.7.1 UE-Initiated Detach procedure for E-UTRAN

5.7.2 MME-Initiated Detach procedure for E-UTRAN

5.7.3 HSS-Initiated Detach procedure for E-UTRAN

Not supported.

5.8 HSS User Profile management function procedure

Not supported.

5.9 Bearer deactivation

5.9.1 PDN GW initiated bearer deactivation

Not supported

5.9.2 MME initiated Dedicated Bearer Deactivation

TBD

5.10 Intra E-UTRAN handover

Not supported yet

6 Annex A: Tools for observing, debugging.

6.1 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.*

6.1.1 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)

6.1.2 Execution

In a shell, execute the following command:

```
user@host:~ itti_analyzer
```

The GUI displayed:

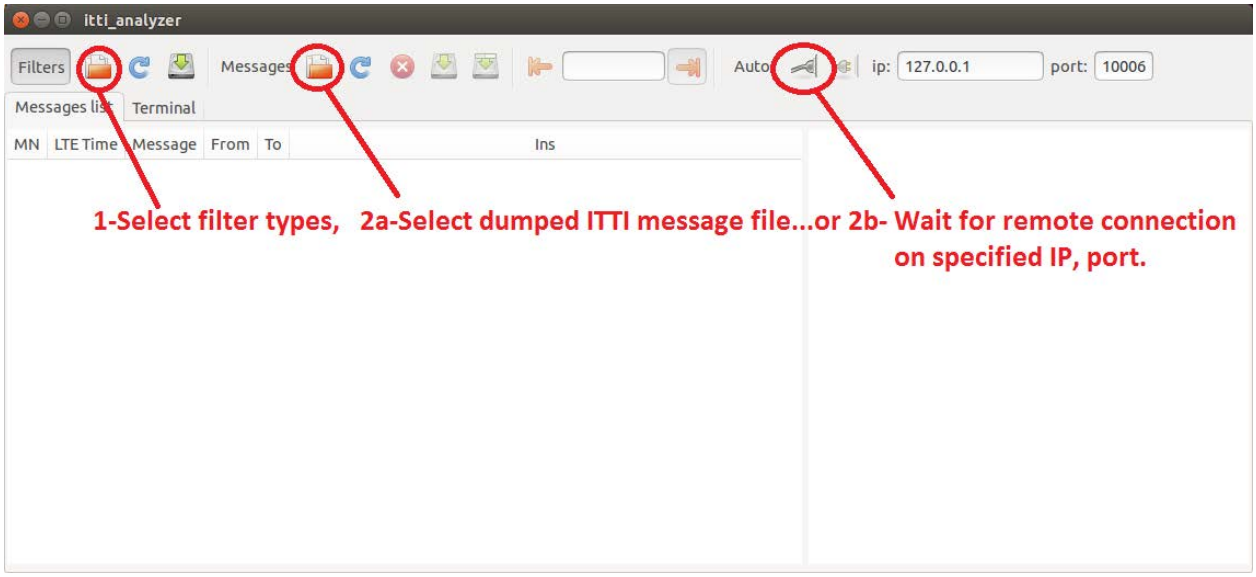


Figure 11 ITTI Analyzer main window

For filter selection, please use filters_mme.xml:

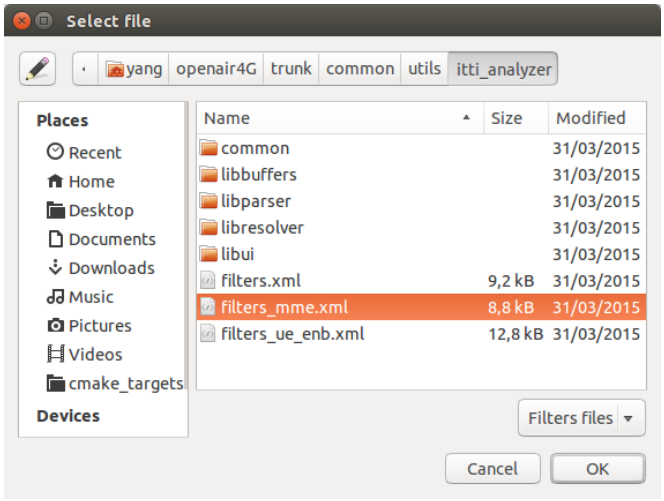


Figure 12 ITTI Analizer select filter menu

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
```

6.2 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).

6.3 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:

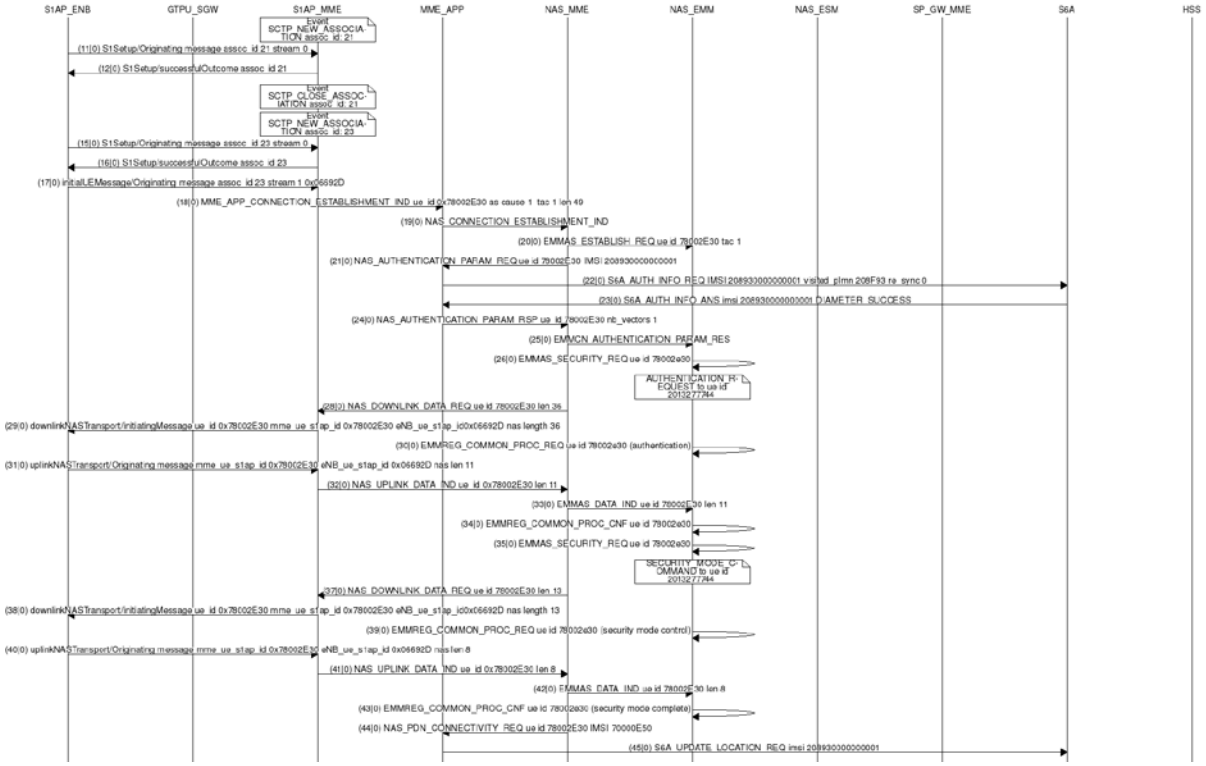


Figure 13 Mscgen output example

6.4 S1AP scenario replay

This tool is available in the **openairinterface5g** git repository, branch "Feature-6-fix_test_core_network_with_scenarios". This branch will be merged in the develop branch as soon as possible.

The aim of this tool is for development, non-regression test, debug purpose, it allows to replay without the help of any eNB(s) or UE(s). a S1AP scenario previously captured as a pcap dumped file.

6.4.1 Overall process

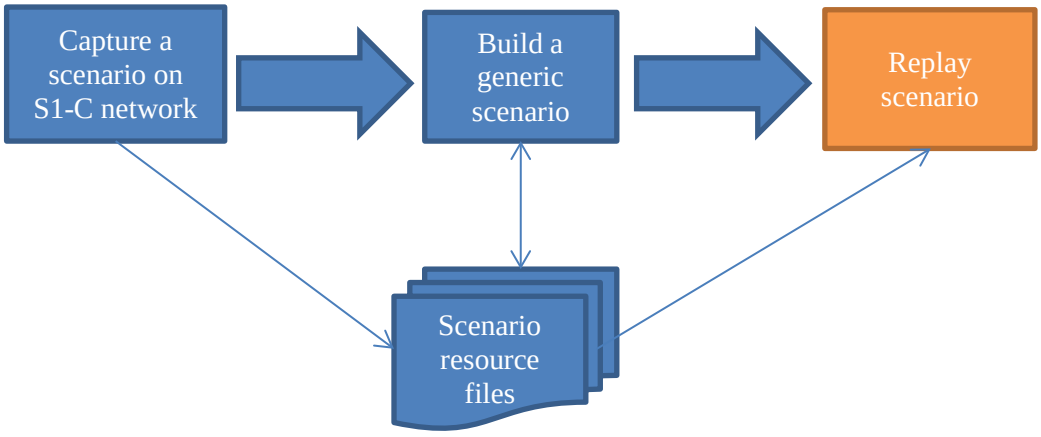


Figure 14 Workflow of scenario replay

In order to replay a scenario, a scenario has to be played/captured (step 1), then the captured artefacts have to processed in order to generate a generic scenario (step 2) ready to be replayed on any other testbed (step 3).

The first step consists in capturing a network trace on S1-C network, the second step consists in building a scenario file that is generic, meaning there is no specific testbed references (IP addresses). The last step is the replaying of the scenario on a openair-cn testbed.

6.4.2 Flowchart of step1: Network trace capture on S1-C

The goal of this step is to capture a SCTP/S1AP trace that we want to be able to replay.

The red items in the following flowchart figure are part of the "scenario resource files".

It is highly recommended to create a dedicated resource directory (*$RESOURCE_SCENARIO_DIR*) for **each** created scenario.

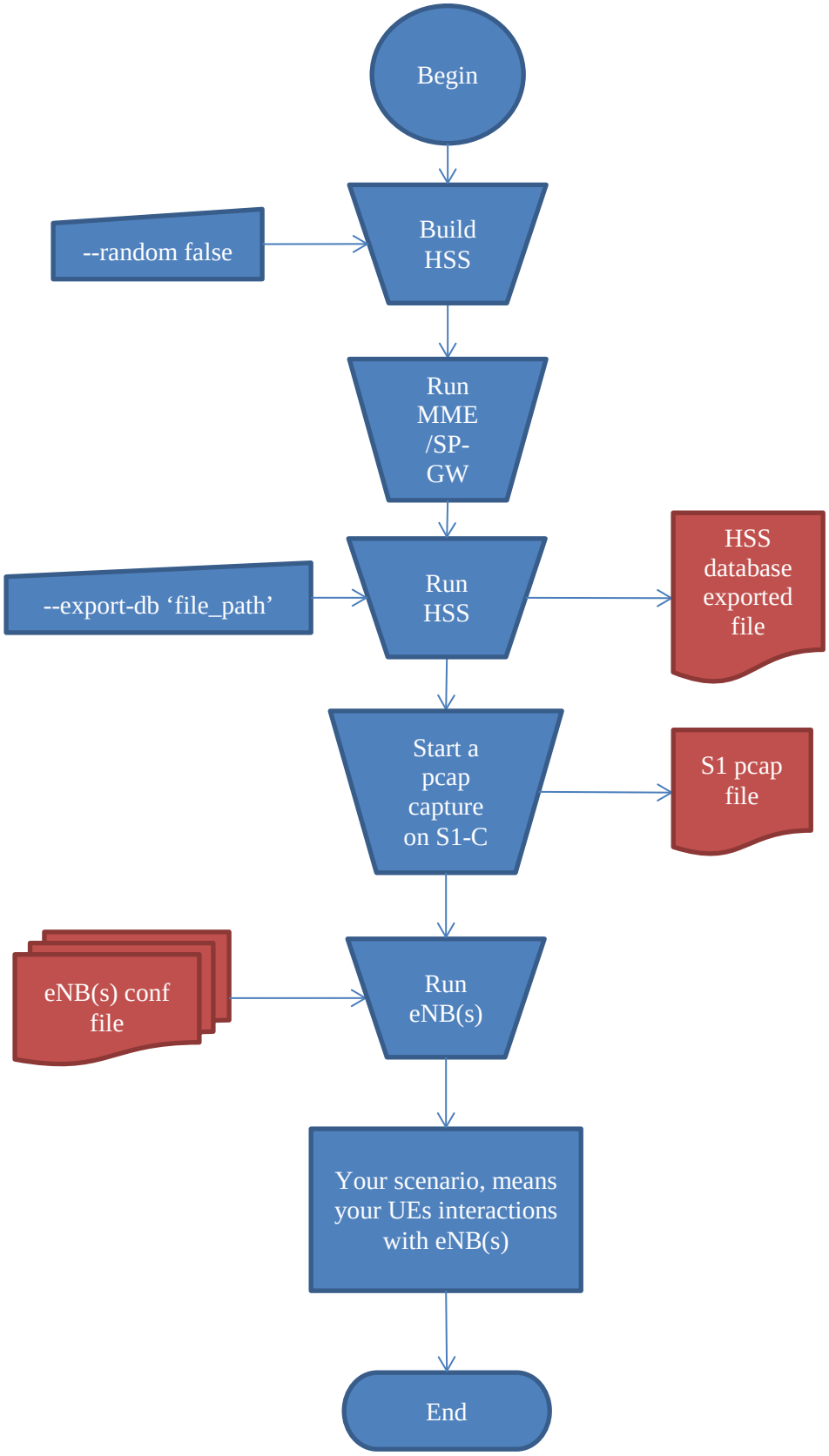


Figure 15 Flowchart of scenario capture

Step 1: build HSS

The HSS database has first to be configured for not generating randoms in security algorithms, otherwise we will not be able to replay trace coming from UE(s).

The argument that has to be passed to the build_hss script is: *--random false*

Step 2: run MME-GW

No change here compared to standard case.

Step 3: run HSS

In order to be able to replay the scenario in the same conditions, the initial content of the database has to be saved.

The argument that has to be passed to the run_hss script is:
--export-db $RESOURCE_SCENARIO_DIR/scenario.sql

Step 4: start a pcap-ng capture on S1-C

This step is not automated, you have to start on your own a tool to capture the network traffic on the S1-C network. (you can use wireshark or tshark)

Step 5: start the eNB(s)

The eNB config files will be used later to make a scenario independent of IP addresses.

Step 6: run your scenario

At the end of this process you will have to save in a directory dedicated to your scenario:

- The eNB config files
- The exported database SQL file.
- The pcap-ng file containing all SCTP and S1AP traffic occurred on S1-C network.

6.4.3 Build a generic scenario

Prerequisites:

1) Build scenario builder
   In openairinterface5g repository
   (https://twiki.eurecom.fr/twiki/bin/view/OpenAirInterface/GetSources):
   ```
   #cd openairinterface5g
   #source ./oaienv
   #cd cmake_targets
   #tools/build_test_epc_tools --debug –clean
   ```
   The executable /usr/local/bin/test_epc_generate_scenario should be generated.
   (The executable /usr/local/bin/test_epc_play_scenario should also be generated.)

2) In your dedicated scenario resource directory(*$RESOURCE_SCENARIO_DIR*), you should have the three files listed in previous section: the eNB(s) configuration file, the exported database SQL file, the pcap-ng file.

For building a generic scenario, follow the steps described below(assuming the EPC has been built on this host):

```
# test_epc --test-dir $RESOURCE_SCENARIO_DIR --old-enb-config-file enb.conf --pcap-file my_test.pcap --build-test
```

This will generate the files my_test.pdml, my_test.xml in the $RESOURCE_SCENARIO_DIR directory.

my_test.pdml file is the xml dump of the pcap file my_test.pcap, it has served as input for building the generic scenario file my_test.xml.

The following figure shows the inner operations of last test_epc command:

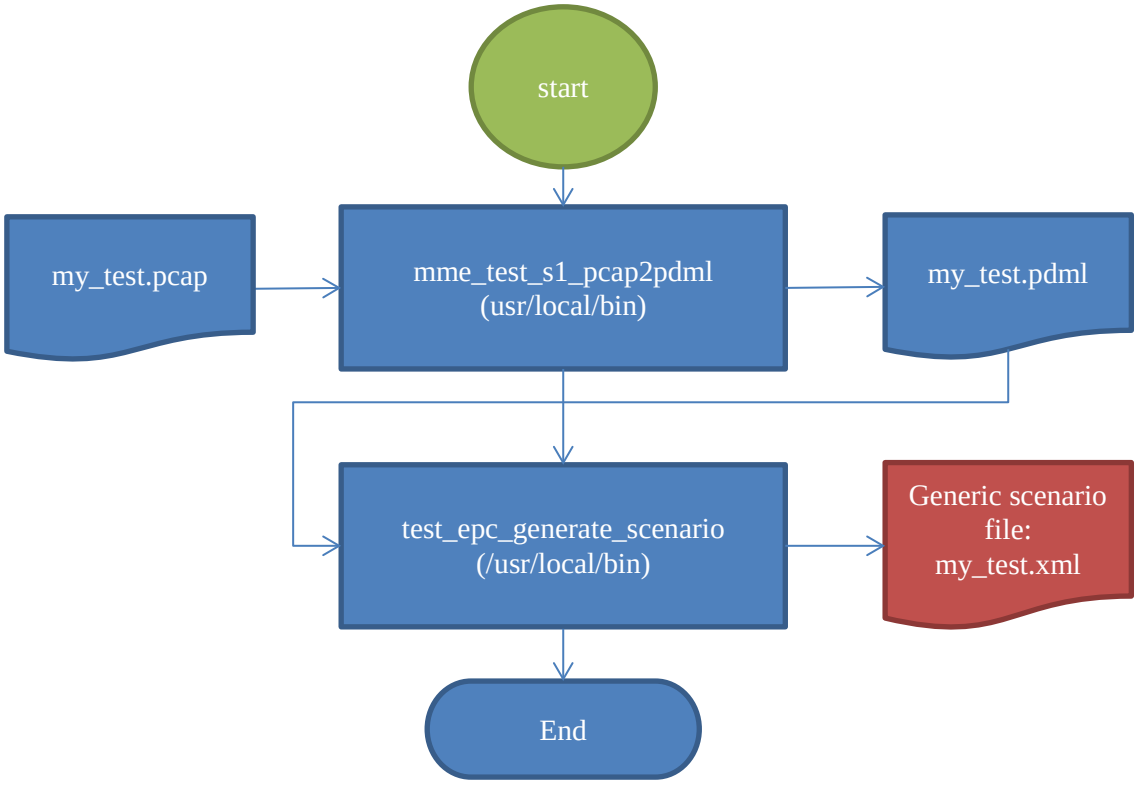


Figure 16 Build scenario detailed operations

6.4.4 Replay a S1AP generic scenario

Prerequisites:

1) Build scenario player (step done in 6.4.3)
   In openairinterface5g repository
   (https://twiki.eurecom.fr/twiki/bin/view/OpenAirInterface/GetSources):
   ```
   #cd openairinterface5g
   #source ./oaienv
   #cd cmake_targets
   #tools/build_test_epc_tools --debug –clean
   ```
   The executable /usr/local/bin/test_epc_play_scenario should be generated.
   (The executable /usr/local/bin/test_epc_generate_scenario should also be generated.)

2) In your dedicated scenario resource directory(*$RESOURCE_SCENARIO_DIR*), you should have almost three mandatory files:
   a. the eNB(s) configuration file for your running testbed (may be different from the original enb.conf file if capture testbed and running testbeds are not the same).
   b. The exported database SQL file.
   c. The generic scenario xml file.

For running a generic scenario, follow the steps described below:

Please note that the eNB(s) emulator “test_epc_play_scenario” can be run exactly like an eNB, from the deployment point of view, it can be run on any EPC host (mme_gw, HSS, or both) or on another host.

```
# test_epc --test-dir $RESOURCE_SCENARIO_DIR --enb-config-file my_new_enb.conf --scenario-file
my_test.xml --run
```