sysmocom

sysmocom - s.f.m.c. GmbH

OsmoMSC User Manual

by Neels Hofmeyr

Copyright © 2017 sysmocom - s.f.m.c. GmbH

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with the Invariant Sections being just 'Foreword', 'Acknowledgements' and 'Preface', with no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

The Asciidoc source code of this manual can be found at http://git.osmocom.org/osmo-gsm-manuals/

HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
1	September 18th, 2017	Initial version; based on OsmoNITB manual version 2.	NH

Contents

1 Foreword **1**

1.1 Acknowledgements . . . 1

1.2 Endorsements . . . 2

2 Preface **2**

2.1 FOSS lives by contribution! . . . 2

2.2 Osmocom and sysmocom . . . 3

2.3 Corrections . . . 3

2.4 Legal disclaimers . . . 3

2.4.1 Spectrum License . . . 3

2.4.2 Software License . . . 3

2.4.3 Trademarks . . . 3

2.4.4 Liability . . . 4

2.4.5 Documentation License . . . 4

3 Introduction **4**

3.1 Required Skills . . . 4

3.2 Getting assistance . . . 5

4 Overview **5**

4.1 About OsmoMSC . . . 5

4.2 Software Components . . . 6

4.2.1 SMSC . . . 6

4.2.2 MSC . . . 6

4.2.3 VLR . . . 6

5 Running OsmoMSC **6**

5.1 SYNOPSIS . . . 6

5.2 OPTIONS . . . 7

5.3 Multiple instances . . . 7

5.4 Configure primary links . . . 8

5.4.1 Configure SCCP/M3UA to accept *A* and *IuCS* links . . . 8

5.4.2 Configure GSUP to reach the HLR . . . 8

5.4.3 Configure MGCP to connect to an MGW . . . 8

6 Control interface **9**

6.1 subscriber-list-active-v1 . . . 9

7 Osmocom Counters 9
7.1 Osmo Counters (deprecated) 10
7.2 Rate Counters 10
7.3 Stat Item 10
7.4 Statistic Levels 10
7.4.1 Global 10
7.4.2 Peer 10
7.4.3 Subscriber 11
7.5 Stats Reporter 11
7.5.1 Configuring a stats reporter 11
7.6 Socket stats 12
7.6.1 Configuration 12
7.6.2 Generated stats items 12

8 Counters 13
8.1 Rate Counters 13

9 Osmo Stat Items 14

10 Osmo Counters 14

11 The Osmocom VTY Interface 14
11.1 Accessing the telnet VTY 15
11.2 VTY Nodes 16
11.3 Interactive help 16
11.3.1 The question-mark (?) command 16
11.3.2 TAB completion 18
11.3.3 The `list` command 18
11.3.4 The attribute system 20
11.3.5 The expert mode 21

12 libosmocore Logging System 22
12.1 Log categories 22
12.2 Log levels 22
12.3 Log printing options 23
12.4 Log filters 23
12.5 Log targets 24
12.5.1 Logging to the VTY 24
12.5.2 Logging to the ring buffer 24
12.5.3 Logging via gsmtap 24
12.5.4 Logging to a file 26
12.5.5 Logging to syslog 26
12.5.6 Logging to systemd-journal 27
12.5.7 Logging to stderr 28

13 Osmocom SS7 + SIGTRAN support **28**
13.1 History / Background . . . 28
13.1.1 The Past (before 2017) . . . 28
13.1.2 The present (2017) . . . 28
13.2 Osmocom extensions to SIGTRAN . . . 29
13.2.1 Osmocom M3UA Routing Key Management Extensions . . . 29
13.2.2 IPA / SCCPlite backwards compatibility . . . 30
13.3 Minimal Osmocom SIGTRAN configurations for small networks . . . 30
13.3.1 A minimal 2G configuration to get started . . . 30
13.3.2 A minimal 3G configuration to get started . . . 31
13.4 Osmocom SS7 Instances . . . 32
13.5 Osmocom SS7 xUA Server . . . 32
13.6 Osmocom SS7 Users . . . 32
13.7 Osmocom SS7 Links . . . 32
13.8 Osmocom SS7 Linksets . . . 33
13.9 Osmocom SS7 Application Servers . . . 33
13.10 Osmocom SS7 Application Server Processes . . . 33
13.11 Osmocom SS7 Routes . . . 34
13.12 Osmocom SCCP Instances . . . 34
13.13 Osmocom SCCP User . . . 34
13.14 Osmocom SCCP Connection . . . 34
13.15 Osmocom SCCP User SAP . . . 34
13.16 Osmocom MTP User SAP . . . 35

14 Configure SCCP/M3UA **35**
14.1 Connect to STP Instance . . . 36
14.2 Local Point-Code . . . 36
14.3 Remote Point-Code . . . 37
14.4 Point-Code Format . . . 38
14.5 AS and ASP . . . 38
14.6 Subsystem Number (SSN) . . . 39
14.7 Routing Context / Routing Key . . . 39
14.7.1 M3UA without Routing Context IE / Routing Context *0* . . . 39
14.7.2 Example: Static Routing . . . 40

15 Configure SCCP/M3UA **42**
15.1 Connect to STP Instance . . . 43
15.2 Local Point-Code . . . 44
15.3 Remote Point-Code . . . 44
15.4 Point-Code Format . . . 45
15.5 AS and ASP . . . 46
15.6 Subsystem Number (SSN) . . . 46
15.7 Routing Context / Routing Key . . . 47
15.7.1 M3UA without Routing Context IE / Routing Context *0* . . . 47
15.7.2 Example: Static Routing . . . 47

16 Configuring the Core Network **50**
16.1 MCC/MNC . . . 50
16.2 Configuring MM INFO . . . 51
16.3 Authentication . . . 51
16.3.1 Authentication on 2G . . . 51
16.3.2 Authentication on 3G . . . 51
16.4 Ciphering . . . 51
16.4.1 Ciphering on 2G . . . 52
16.4.2 Ciphering on 3G . . . 52

17 Short Message Peer to Peer (SMPP) **53**
17.1 Global SMPP configuration . . . 53
17.2 ESME configuration . . . 53
17.3 Example configuration snippet . . . 54
17.4 Osmocom SMPP protocol extensions . . . 54
17.4.1 RF channel measuremets . . . 54
17.4.2 Equipment IMEI . . . 54

18 MNCC for External Call Control **55**
18.1 Internal MNCC handler . . . 55
18.1.1 Internal MNCC Configuration . . . 55
18.1.1.1 `default-codec tch-f (fr|efr|amr)` . . . 55
18.1.1.2 `default-codec tch-h (hr|amr)` . . . 55
18.2 External MNCC handler . . . 55
18.3 DTMF considerations . . . 56
18.4 MNCC protocol description . . . 56
18.4.1 MNCC_HOLD_IND . . . 56
18.4.2 MNCC_HOLD_CNF . . . 56
18.4.3 MNCC_HOLD_REJ . . . 56

18.4.4 MNCC_RETRIEVE_IND . . . 56
18.4.5 MNCC_RETRIEVE_CNF . . . 56
18.4.6 MNCC_RETRIEVE_REJ . . . 57
18.4.7 MNCC_USERINFO_REQ . . . 57
18.4.8 MNCC_USERINFO_IND . . . 57
18.4.9 MNCC_BRIDGE . . . 57
18.4.10 MNCC_FRAME_RECV . . . 57
18.4.11 MNCC_FRAME_DROP . . . 57
18.4.12 MNCC_LCHAN_MODIFY . . . 57
18.4.13 MNCC_RTP_CREATE . . . 58
18.4.14 MNCC_RTP_CONNECT . . . 58
18.4.15 MNCC_RTP_FREE . . . 58
18.4.16 GSM_TCHF_FRAME . . . 58
18.4.17 GSM_TCHF_FRAME_EFR . . . 58
18.4.18 GSM_TCHH_FRAME . . . 58
18.4.19 GSM_TCH_FRAE_AMR . . . 58
18.4.20 GSM_BAD_FRAME . . . 58
18.4.21 MNCC_START_DTMF_IND . . . 58
18.4.22 MNCC_START_DTMF_RSP . . . 59
18.4.23 MNCC_START_DTMF_REJ . . . 59
18.4.24 MNCC_STOP_DTMF_IND . . . 59
18.4.25 MNCC_STOP_DTMF_RSP . . . 59

19 Osmux **59**
19.1 Osmux and NAT . . . 59
19.2 CID allocation . . . 60
19.3 3GPP AoIP network setup with Osmux . . . 61
19.4 SCCPLite network setup with Osmux . . . 63
19.5 SCCPLite network setup with Osmux + BSC-NAT . . . 65
19.6 Osmux and MGCP . . . 67
19.6.1 `X-Osmux` Format . . . 67
19.6.2 `X-Osmux` Considerations . . . 67
19.6.3 `X-Osmux` Support . . . 68
19.7 Abis setup with Osmux . . . 68
19.8 Osmux Support in OsmoMSC . . . 69
19.8.1 OsmoMSC in a A/IP with IPA/SCCPlite network setup . . . 69
19.8.2 OsmoMSC in a 3GPP AoIP network setup . . . 70

20 MGW Pooling **70**
20.1 MGW pool VTY configuration . . . 70
20.2 MGW pool management . . . 71
20.2.1 MGW pool status . . . 71
20.2.2 Adding an MGW / MGCP-Client to the MGW pool . . . 71
20.2.3 Reconnecting an MGW / MGCP-Client . . . 72
20.2.4 Monitor MGCP link (keepalive) / MGCP-Client . . . 73
20.2.5 Blocking an MGW / MGCP-Client . . . 73
20.2.6 Removing an MGW / MGCP-Client . . . 74

21 SGs interface **74**
21.1 VTY configuration . . . 74
21.2 Connection monitoring . . . 75

22 Osmocom Control Interface **75**
22.1 Control Interface Protocol . . . 75
22.1.1 GET operation . . . 76
22.1.2 SET operation . . . 77
22.1.3 TRAP operation . . . 77
22.2 Common variables . . . 77
22.3 Control Interface python examples . . . 78
22.3.1 Getting rate counters . . . 78
22.3.2 Setting a value . . . 79
22.3.3 Getting a value . . . 79
22.3.4 Listening for traps . . . 79

23 Generic Subscriber Update Protocol **79**
23.1 General . . . 79
23.2 Connection . . . 80
23.3 Using IPA . . . 80
23.4 Procedures . . . 80
23.4.1 Authentication management . . . 80
23.4.2 Reporting of Authentication Failure . . . 80
23.4.3 Location Updating . . . 81
23.4.4 Location Cancellation . . . 81
23.4.5 Purge MS . . . 81
23.4.6 Delete Subscriber Data . . . 82
23.4.7 Check IMEI . . . 82
23.5 Procedures (E Interface) . . . 82
23.5.1 E Handover . . . 82

23.5.2 E Subsequent Handover . . . 83
23.5.3 E Forward and Process Access Signalling . . . 83
23.5.4 E Routing Error . . . 84
23.6 Message Format . . . 84
23.6.1 General . . . 84
23.6.2 Send Authentication Info Request . . . 84
23.6.3 Send Authentication Info Error . . . 85
23.6.4 Send Authentication Info Response . . . 85
23.6.5 Authentication Failure Report . . . 85
23.6.6 Update Location Request . . . 85
23.6.7 Update Location Error . . . 85
23.6.8 Update Location Result . . . 85
23.6.9 Location Cancellation Request . . . 86
23.6.10 Location Cancellation Error . . . 86
23.6.11 Location Cancellation Result . . . 86
23.6.12 Purge MS Request . . . 86
23.6.13 Purge MS Error . . . 86
23.6.14 Purge MS Result . . . 87
23.6.15 Insert Subscriber Data Request . . . 87
23.6.16 Insert Subscriber Data Error . . . 87
23.6.17 Insert Subscriber Data Result . . . 87
23.6.18 Delete Subscriber Data Request . . . 87
23.6.19 Delete Subscriber Data Error . . . 88
23.6.20 Delete Subscriber Data Result . . . 88
23.6.21 Process Supplementary Service Request . . . 88
23.6.22 Process Supplementary Service Error . . . 88
23.6.23 Process Supplementary Service Response . . . 88
23.6.24 MO-forwardSM Request . . . 89
23.6.25 MO-forwardSM Error . . . 89
23.6.26 MO-forwardSM Result . . . 89
23.6.27 MT-forwardSM Request . . . 89
23.6.28 MT-forwardSM Error . . . 90
23.6.29 MT-forwardSM Result . . . 90
23.6.30 READY-FOR-SM Request . . . 90
23.6.31 READY-FOR-SM Error . . . 90
23.6.32 READY-FOR-SM Result . . . 91
23.6.33 CHECK-IMEI Request . . . 91
23.6.34 CHECK-IMEI Error . . . 91
23.6.35 CHECK-IMEI Result . . . 91

23.6.36 E Prepare Handover Request . . . 92
23.6.37 E Prepare Handover Error . . . 92
23.6.38 E Prepare Handover Result . . . 92
23.6.39 E Prepare Subsequent Handover Request . . . 92
23.6.40 E Prepare Subsequent Handover Error . . . 92
23.6.41 E Prepare Subsequent Handover Result . . . 93
23.6.42 E Send End Signal Request . . . 93
23.6.43 E Send End Signal Error . . . 93
23.6.44 E Send End Signal Result . . . 93
23.6.45 E Process Access Signalling Request . . . 94
23.6.46 E Forward Access Signalling Request . . . 94
23.6.47 E Close . . . 94
23.6.48 E Abort . . . 94
23.6.49 E Routing Error . . . 94
23.6.50 ePDG Tunnel Request . . . 95
23.6.51 ePDG Tunnel Error . . . 95
23.6.52 ePDG Tunnel Result . . . 95
23.7 Information Elements . . . 95
23.7.1 Message Type . . . 95
23.7.2 IP Address . . . 98
23.7.3 PDP Info . . . 98
23.7.4 PDP Address . . . 98
23.7.5 PDP Context ID . . . 100
23.7.6 Auth tuple . . . 100
23.7.7 RAND . . . 100
23.7.8 SRES . . . 101
23.7.9 Kc . . . 101
23.7.10 IK . . . 101
23.7.11 CK . . . 101
23.7.12 AUTN . . . 101
23.7.13 AUTS . . . 101
23.7.14 RES . . . 101
23.7.15 CN Domain . . . 101
23.7.16 Cancellation Type . . . 102
23.7.17 IE Identifier (informational) . . . 102
23.7.18 Empty field . . . 103
23.7.19 IMSI . . . 103
23.7.20 ISDN-AddressString / MSISDN / Called Party BCD Number . . . 104
23.7.21 Access Point Name . . . 104

23.7.22 Quality of Service Subscribed Service . . . 105
23.7.23 PDP-Charging Characteristics . . . 105
23.7.24 Protocol Configuration Options (PCO) . . . 105
23.7.25 HLR Number encoded as 3GPP TS 09.02 ISDN-AddressString . . . 105
23.7.26 Cause . . . 106
23.7.27 Supplementary Service Info . . . 106
23.7.28 IMEI . . . 106
23.7.29 IMEI Check Result . . . 106
23.7.30 Message Class . . . 106
23.7.31 Source Name . . . 107
23.7.32 Destination Name . . . 107
23.7.33 AN-APDU . . . 107
23.7.34 RR Cause . . . 108
23.7.35 BSSAP Cause . . . 108
23.7.36 Current RAT Type . . . 108
23.7.37 Session Management Cause . . . 108
23.8 Session (transaction) management . . . 108
23.8.1 Session ID . . . 109
23.8.2 Session State . . . 109
23.8.3 SM-RP-MR (Message Reference) . . . 109
23.8.4 SM-RP-DA (Destination Address) . . . 109
23.8.5 SM-RP-OA (Originating Address) . . . 109
23.8.6 Coding of SM-RP-DA / SM-RP-OA IEs . . . 110
23.8.7 SM-RP-UI (SM TPDU) . . . 111
23.8.8 SM-RP-Cause (RP Cause value) . . . 112
23.8.9 SM-RP-MMS (More Messages to Send) . . . 112
23.8.10 SM Alert Reason . . . 112

24 VTY Process and Thread management **112**
24.1 Scheduling Policy . . . 112
24.2 CPU-Affinity Mask . . . 113

25 Glossary **114**

A Osmocom TCP/UDP Port Numbers **122**

B Bibliography / References **124**
B.0.0.0.1 References . . . 124

C GNU Free Documentation License 128
C.1 PREAMBLE . . . 128
C.2 APPLICABILITY AND DEFINITIONS . . . 128
C.3 VERBATIM COPYING . . . 129
C.4 COPYING IN QUANTITY . . . 129
C.5 MODIFICATIONS . . . 130
C.6 COMBINING DOCUMENTS . . . 131
C.7 COLLECTIONS OF DOCUMENTS . . . 131
C.8 AGGREGATION WITH INDEPENDENT WORKS . . . 131
C.9 TRANSLATION . . . 132
C.10 TERMINATION . . . 132
C.11 FUTURE REVISIONS OF THIS LICENSE . . . 132
C.12 RELICENSING . . . 132
C.13 ADDENDUM: How to use this License for your documents . . . 133

1 Foreword

Digital cellular networks based on the GSM specification were designed in the late 1980s and first deployed in the early 1990s in Europe. Over the last 25 years, hundreds of networks were established globally and billions of subscribers have joined the associated networks.

The technological foundation of GSM was based on multi-vendor interoperable standards, first created by government bodies within CEPT, then handed over to ETSI, and now in the hands of 3GPP. Nevertheless, for the first 17 years of GSM technology, the associated protocol stacks and network elements have only existed in proprietary *black-box* implementations and not as Free Software.

In 2008 Dieter Spaar and I started to experiment with inexpensive end-of-life surplus Siemens GSM BTSs. We learned about the A-bis protocol specifications, reviewed protocol traces and started to implement the BSC-side of the A-bis protocol as something originally called `bs11-abis`. All of this was *just for fun*, in order to learn more and to boldly go where no Free Software developer has gone before. The goal was to learn and to bring Free Software into a domain that despite its ubiquity, had not yet seen any Free / Open Source software implementations.

`bs11-abis` quickly turned into `bsc-hack`, then *OpenBSC* and its *OsmoNITB* variant: A minimal implementation of all the required functionality of an entire GSM network, exposing A-bis towards the BTS. The project attracted more interested developers, and surprisingly quickly also commercial interest, contribution and adoption. This allowed adding support for more BTS models.

After having implemented the network-side GSM protocol stack in 2008 and 2009, in 2010 the same group of people set out to create a telephone-side implementation of the GSM protocol stack. This established the creation of the Osmocom umbrella project, under which OpenBSC and the OsmocomBB projects were hosted.

Meanwhile, more interesting telecom standards were discovered and implemented, including TETRA professional mobile radio, DECT cordless telephony, GMR satellite telephony, some SDR hardware, a SIM card protocol tracer and many others.

Increasing commercial interest particularly in the BSS and core network components has lead the way to 3G support in Osmocom, as well as the split of the minimal *OsmoNITB* implementation into separate and fully featured network components: OsmoBSC, OsmoMSC, OsmoHLR, OsmoMGW and OsmoSTP (among others), which allow seamless scaling from a simple "Network In The Box" to a distributed installation for serious load.

It has been a most exciting ride during the last eight-odd years. I would not have wanted to miss it under any circumstances.

— Harald Welte, Osmocom.org and OpenBSC founder, December 2017.

1.1 Acknowledgements

My deep thanks to everyone who has contributed to Osmocom. The list of contributors is too long to mention here, but I'd like to call out the following key individuals and organizations, in no particular order:

- Dieter Spaar for being the most amazing reverse engineer I've met in my career
- Holger Freyther for his many code contributions and for shouldering a lot of the maintenance work, setting up Jenkins - and being crazy enough to co-start sysmocom as a company with me ;)
- Andreas Eversberg for taking care of Layer2 and Layer3 of OsmocomBB, and for his work on OsmoBTS and OsmoPCU
- Sylvain Munaut for always tackling the hardest problems, particularly when it comes closer to the physical layer
- Chaos Computer Club for providing us a chance to run real-world deployments with tens of thousands of subscribers every year
- Bernd Schneider of Netzing AG for funding early ip.access nanoBTS support
- On-Waves ehf for being one of the early adopters of OpenBSC and funding a never ending list of features, fixes and general improvement of pretty much all of our GSM network element implementations
- sysmocom, for hosting and funding a lot of Osmocom development, the annual Osmocom Developer Conference and releasing this manual.

- Jan Luebbe, Stefan Schmidt, Daniel Willmann, Pablo Neira, Nico Golde, Kevin Redon, Ingo Albrecht, Alexander Huemer, Alexander Chemeris, Max Suraev, Tobias Engel, Jacob Erlbeck, Ivan Kluchnikov
- NLnet Foundation, for providing funding for a number of individual work items within the Osmocom universe, such as LTE support in OsmoCBC or GPRS/EGPRS support for Ericsson RBS6000.
- WaveMobile Ltd, for many years of sponsoring.

May the source be with you!

— Harald Welte, Osmocom.org and OpenBSC founder, January 2016.

1.2 Endorsements

This version of the manual is endorsed by Harald Welte as the official version of the manual.

While the GFDL license (see Appendix C) permits anyone to create and distribute modified versions of this manual, such modified versions must remove the above endorsement.

2 Preface

First of all, we appreciate your interest in Osmocom software.

Osmocom is a Free and Open Source Software (FOSS) community that develops and maintains a variety of software (and partially also hardware) projects related to mobile communications.

Founded by people with decades of experience in community-driven FOSS projects like the Linux kernel, this community is built on a strong belief in FOSS methodology, open standards and vendor neutrality.

2.1 FOSS lives by contribution!

If you are new to FOSS, please try to understand that this development model is not primarily about "free of cost to the GSM network operator", but it is about a collaborative, open development model. It is about sharing ideas and code, but also about sharing the effort of software development and maintenance.

If your organization is benefiting from using Osmocom software, please consider ways how you can contribute back to that community. Such contributions can be many-fold, for example

- sharing your experience about using the software on the public mailing lists, helping to establish best practises in using/operating it,
- providing qualified bug reports, workarounds
- sharing any modifications to the software you may have made, whether bug fixes or new features, even experimental ones
- providing review of patches
- testing new versions of the related software, either in its current "master" branch or even more experimental feature branches
- sharing your part of the maintenance and/or development work, either by donating developer resources or by (partially) funding those people in the community who do.

We're looking forward to receiving your contributions.

2.2 Osmocom and sysmocom

Some of the founders of the Osmocom project have established *sysmocom - systems for mobile communications GmbH* as a company to provide products and services related to Osmocom.

sysmocom and its staff have contributed by far the largest part of development and maintenance to the Osmocom mobile network infrastructure projects.

As part of this work, sysmocom has also created the manual you are reading.

At sysmocom, we draw a clear line between what is the Osmocom FOSS project, and what is sysmocom as a commercial entity. Under no circumstances does participation in the FOSS projects require any commercial relationship with sysmocom as a company.

2.3 Corrections

We have prepared this manual in the hope that it will guide you through the process of installing, configuring and debugging your deployment of cellular network infrastructure elements using Osmocom software. If you do find errors, typos and/or omissions, or have any suggestions on missing topics, please do take the extra time and let us know.

2.4 Legal disclaimers

2.4.1 Spectrum License

As GSM and UMTS operate in licensed spectrum, please always double-check that you have all required licenses and that you do not transmit on any ARFCN or UARFCN that is not explicitly allocated to you by the applicable regulatory authority in your country.

Warning
Depending on your jurisdiction, operating a radio transmitter without a proper license may be considered a felony under criminal law!

2.4.2 Software License

The software developed by the Osmocom project and described in this manual is Free / Open Source Software (FOSS) and subject to so-called *copyleft* licensing.

Copyleft licensing is a legal instrument to ensure that this software and any modifications, extensions or derivative versions will always be publicly available to anyone, for any purpose, under the same terms as the original program as developed by Osmocom.

This means that you are free to use the software for whatever purpose, make copies and distribute them - just as long as you ensure to always provide/release the *complete and corresponding* source code.

Every Osmocom software includes a file called `COPYING` in its source code repository which explains the details of the license. The majority of programs is released under GNU Affero General Public License, Version 3 (AGPLv3).

If you have any questions about licensing, don't hesitate to contact the Osmocom community. We're more than happy to clarify if your intended use case is compliant with the software licenses.

2.4.3 Trademarks

All trademarks, service marks, trade names, trade dress, product names and logos appearing in this manual are the property of their respective owners. All rights not expressly granted herein are reserved.

For your convenience we have listed below some of the registered trademarks referenced herein. This is not a definitive or complete list of the trademarks used.

Osmocom® and *OpenBSC*® are registered trademarks of Holger Freyther and Harald Welte.

sysmocom® and *sysmoBTS*® are registered trademarks of *sysmocom - systems for mobile communications GmbH*.

ip.access® and *nanoBTS*® are registered trademarks of *ip.access Ltd.*

2.4.4 Liability

The software is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the License text included with the software for more details.

2.4.5 Documentation License

Please see Appendix C for further information.

3 Introduction

3.1 Required Skills

Please note that even while the capital expenses of running mobile networks has decreased significantly due to Osmocom software and associated hardware like sysmoBTS, GSM networks are still primarily operated by large GSM operators.

Neither the GSM specification nor the GSM equipment was ever designed for networks to be installed and configured by anyone but professional GSM engineers, specialized in their respective area like radio planning, radio access network, back-haul or core network.

If you do not share an existing background in GSM network architecture and GSM protocols, correctly installing, configuring and optimizing your GSM network will be tough, irrespective whether you use products with Osmocom software or those of traditional telecom suppliers.

GSM knowledge has many different fields, from radio planning through site installation to core network configuration/administration.

The detailed skills required will depend on the type of installation and/or deployment that you are planning, as well as its associated network architecture. A small laboratory deployment for research at a university is something else than a rural network for a given village with a handful of cells, which is again entirely different from an urban network in a dense city.

Some of the useful skills we recommend are:

- general understanding about RF propagation and path loss in order to estimate coverage of your cells and do RF network planning.
- general understanding about GSM network architecture, its network elements and key transactions on the Layer 3 protocol
- general understanding about voice telephony, particularly those of ISDN heritage (Q.931 call control)
- understanding of GNU/Linux system administration and working on the shell
- understanding of TCP/IP networks and network administration, including tcpdump, tshark, wireshark protocol analyzers.
- ability to work with text based configuration files and command-line based interfaces such as the VTY of the Osmocom network elements

3.2 Getting assistance

If you do have a support package / contract with sysmocom (or want to get one), please contact support@sysmocom.de with any issues you may have.

If you don't have a support package / contract, you have the option of using the resources put together by the Osmocom community at https://projects.osmocom.org/, checking out the wiki and the mailing-list for community-based assistance. Please always remember, though: The community has no obligation to help you, and you should address your requests politely to them. The information (and software) provided at osmocom.org is put together by volunteers for free. Treat them like a friend whom you're asking for help, not like a supplier from whom you have bought a service.

If you would like to obtain professional/commercial support on Osmocom CNI, you can always reach out to sales@sysmocom.de to discuss your support needs. Purchasing support from sysmocom helps to cover the ongoing maintenance of the Osmocom CNI software stack.

4 Overview

This manual should help you getting started with OsmoMSC. It will cover aspects of configuring and running the OsmoMSC.

4.1 About OsmoMSC

OsmoMSC is the Osmocom implementation of a Mobile Switching Center (MSC) for 2G and 3G GSM and UMTS mobile networks. Its interfaces are:

- GSUP towards OsmoHLR (or a MAP proxy);
- A over IP towards a BSC (e.g. OsmoBSC);
- IuCS towards an RNC or HNB-GW (e.g. OsmoHNBGW) for 3G voice;
- MNCC (Mobile Network Call Control derived from GSM TS 04.07);
- SMPP 3.4 (Short Message Peer-to-Peer);
- The Osmocom typical telnet VTY and CTRL interfaces.

OsmoMSC originated from the OpenBSC project, which started as a minimalistic all-in-one implementation of the GSM Network. In 2017, OpenBSC had reached maturity and diversity (including M3UA SIGTRAN and 3G support in the form of IuCS and IuPS interfaces) that naturally lead to a separation of the all-in-one approach to fully independent separate programs as in typical GSM networks. Before it was split off, OsmoMSC originated from libmsc of the old openbsc.git. Since a true *A* interface and IuCS for 3G support is available, OsmoMSC exists only as a separate standalone entity.

Key differences of the new OsmoMSC compared to the old OsmoNITB are:

- The complete VLR implementation that communicates with the separate HLR (OsmoHLR) for subscriber management. In contrast to the OsmoNITB, HLR queries are fully asynchronous, and the separate HLR allows using centralized subscriber management for both circuit-switched and packet-switched domains (i.e. one OsmoHLR for both OsmoMSC and OsmoSGSN).
- VLR and HLR brought full UMTS AKA (Authentication and Key Agreement) support, i.e. Milenage authentication in both the full 3G variant as well as the backwards compatible 2G variant.
- Addition of a true *A* interface for 2G voice services. Previously, OsmoBSC had an SCCPlite based *A* interface towards 3rd party MSC implementations. OsmoMSC features a true SCCP/M3UA *A* interface, which allows running OsmoBSC against this Osmocom based MSC implementation. The new SCCP/M3UA SIGTRAN for the *A* interface is implemented in libosmosigtran, which is used by OsmoMSC and OsmoBSC (and others), to establish a link via an STP (e.g. OsmoSTP).
- Addition of an *IuCS* interface to allow operating 3G voice services, also via SCCP/M3UA SIGTRAN, for example connecting via OsmoHNBGW to a 3G small cell device.

Find the OsmoMSC issue tracker and wiki online at

- https://osmocom.org/projects/osmomsc
- https://osmocom.org/projects/osmomsc/wiki

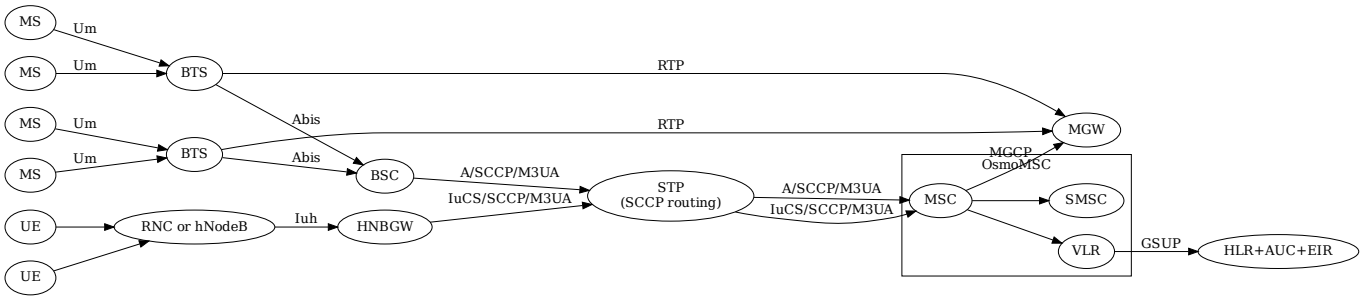


Figure 1: Typical GSM network architecture used with OsmoMSC

4.2 Software Components

This is a brief description of OsmoMSC's internal software components.

4.2.1 SMSC

A minimal store-and-forward server for SMS, supporting both MO and MT SMS service, as well as multi-part messages.

The built-in SMSC also supports an external SMSC interface. For more information, see Section 17.

4.2.2 MSC

The MSC component implements the mobility management (MM) functions of the TS 04.08 and delegates to SMSC for SMS message handling and the VLR for subscriber management.

Furthermore, it can handle TS 04.08 Call Control (CC), either by use of an internal MNCC handler, or by use of an external MNCC agent. For more information see Section 18.

4.2.3 VLR

A fully featured Visitor Location Register handles the subscriber management and authentication, and interfaces via GSUP to the external HLR.

5 Running OsmoMSC

The OsmoMSC executable (`osmo-msc`) offers the following command-line arguments:

5.1 SYNOPSIS

osmo-msc [-h|-V] [-d *DBGMASK*] [-D] [-c *CONFIGFILE*] [-s] [-T] [-e *LOGLEVEL*]

5.2 OPTIONS

-h, --help
Print a short help message about the supported options

-V, --version
Print the compile-time version number of the program

-d, --debug *DBGMASK,DBGLEVELS*
Set the log subsystems and levels for logging to stderr. This has mostly been superseded by VTY-based logging configuration, see Section 12 for further information.

-D, --daemonize
Fork the process as a daemon into background.

-c, --config-file *CONFIGFILE*
Specify the file and path name of the configuration file to be used. If none is specified, use `osmo-msc.cfg` in the current working directory.

-s, --disable-color
Disable colors for logging to stderr. This has mostly been deprecated by VTY based logging configuration, see Section 12 for more information.

-T, --timestamp
Enable time-stamping of log messages to stderr. This has mostly been deprecated by VTY based logging configuration, see Section 12 for more information.

-e, --log-level *LOGLEVEL*
Set the global log level for logging to stderr. This has mostly been deprecated by VTY based logging configuration, see Section 12 for more information.

5.3 Multiple instances

Running multiple instances of `osmo-msc` on the same computer is possible if all interfaces (VTY, CTRL) are separated using the appropriate configuration options. The IP based interfaces are binding to local host by default. In order to separate the processes, the user has to bind those services to specific but different IP addresses and/or ports.

The VTY and the Control interface can be bound to IP addresses from the loopback address range, for example:

```
line vty
 bind 127.0.0.2
ctrl
 bind 127.0.0.2
```

If external SMPP is enabled, you may bind it to a different interface using:

```
smpp
 local-tcp-ip 10.23.42.1 2775
```

More on SMPP configuration in [?].

The external MNCC handler is a UNIX domain socket that is created when external MNCC handling is configured. A separate path must be used per osmo-msc instance:

```
msc
 mncc external /tmp/mncc_socket
```

More on MNCC in Section 18.2.

The SGs interface by default listens on 0.0.0.0:29118 (SCTP). Each instance of OsmoMSC must use a different IP address and/or port. For details about the configuration of the SGs interface, see section Section 21.

For the following links, OsmoMSC acts as a client and does not listen/bind to a specific interface, and will hence not encounter conflicts for multiple instances running on the same interface:

- The SCCP/M3UA links are established by OsmoMSC contacting an STP.
- The GSUP link is established by OsmoMSC contacting an HLR.
- The MGCP link is established by OsmoMSC contacting an MGW.

5.4 Configure primary links

5.4.1 Configure SCCP/M3UA to accept *A* and *IuCS* links

OsmoMSC acts as client to contact an STP instance and establish an SCCP/M3UA link.

An example configuration of OsmoMSC's SCCP link:

```
cs7 instance 0
 point-code 0.23.1
 asp asp-clnt-OsmoMSC-A-Iu 2905 0 m3ua
  remote-ip 127.0.0.1
  role asp
  sctp-role client
```

This configuration is explained in detail in Section 15.

Note that *A* and *IuCS* may use different SCCP instances, if so desired:

```
cs7 instance 0
 asp my-OsmoMSC-A 2905 0 m3ua
  remote-ip 10.23.42.1
  role asp
  sctp-role client
cs7 instance 1
 asp my-OsmoMSC-Iu 2905 0 m3ua
  remote-ip 10.23.42.2
  role asp
  sctp-role client
msc
 cs7-instance-a 0
 cs7-instance-iu 1
```

5.4.2 Configure GSUP to reach the HLR

OsmoMSC will assume a GSUP server (OsmoHLR) to run on the local host and the default GSUP port (4222). Contacting an HLR at a different IP address can be configured as follows:

```
hlr
 ! IP address of the remote HLR:
 remote-ip 10.23.42.1
 ! default port is 4222, optionally configurable by:
 remote-port 1234
```

5.4.3 Configure MGCP to connect to an MGW

OsmoMSC uses a media gateway (typically OsmoMGW) to direct RTP streams. By default, an MGW is expected to receive MGCP requests on the IANA-registered default port for MGCP (2427) on local host (127.0.0.1).

Here is an example configuration for a remote MGW:

```
network
 mgw 0
  remote-ip 10.9.8.7
  remote-port 2427
  reset-endpoint rtpbridge/* ❶
```

❶ The *reset-endpoint* setting instructs the OsmoMGW to send a wildcarded DLCX to the media gateway. This helps to clear lingering calls from the media gateway when the OsmoMSC is restarted.

OsmoMSC is also able to handle a pool of media gateways for load distribution. See also Section 20.

Note
Previous versions of OsmoMSC (1.9.0 and below) didn't have the *mgw* VTY node and hence didn't support the MGW pooling feature. Therefore, historically the MGW related commands where placed under the `msc` VTY node. The MGW related commands under the `msc` VTY are still parsed and used but its use is deprecated and hence discouraged in favour of the new `mgw` node. Writing the config to a file from within OsmoMSC will automatically convert the config to use the new `mgw` node.

6 Control interface

The actual protocol is described in Section 22, the variables common to all programs using it are described in Section 22.2. This section describes the CTRL interface variables specific to OsmoMSC.

Table 1: Variables available on OsmoMSC's Control interface

Name	Access	Trap	Value	Comment
subscriber-list-active-v1	RO	No		Return list of active subscribers.

6.1 subscriber-list-active-v1

Return a list of subscribers that are successfully attached (including full successful authentication and ciphering if those are enabled).

The reply comprises of one subscriber per line, of the format

```
<IMSI>,<MSISDN>\n[<IMSI>,<MSISDN>\n[...]]
```

For example:

```
901700000015252,22801
901700000015253,22802
```

7 Osmocom Counters

The following gives an overview of all the types of counters available:

7.1 Osmo Counters (deprecated)

Osmo counters are the oldest type of counters added to Osmocom projects. They are not grouped.

- Printed as part of VTY show stats
- Increment, Decrement
- Accessible through the control interface: counter.<counter_name>

7.2 Rate Counters

Rate counters count rates of events.

- Printed as part of VTY show stats
- Intervals: per second, minute, hour, day or absolute value
- Increment only
- Accessible through the control interface
- Rate counters are grouped and different instances per group can exist

The control interface command to get a counter (group) is:

rate_ctr.per_{sec,min,hour,day,abs}.<group_name>.<idx>.[counter_name]

It is possible to get all counters in a group by omitting the counter name

7.3 Stat Item

Stat items are a grouped replacement for osmo counters.

- Printed as part of VTY show stats
- Replacement for osmo counters
- Not yet available through the control interface
- Grouped and indexed like rate counters
- Items have a unit
- Keeps a list of the last values measured, so could return an average, min, max, std. deviation. So far this is not implemented in any of the reporting options.

7.4 Statistic Levels

There are three levels on which a statistic can be aggregated in Osmocom projects: globally, per-peer and per-subscriber.

7.4.1 Global

These are global statistics.

7.4.2 Peer

These statistics relate to a peer the program connects to such as the NSVC in an SGSN.

This level also includes reporting global statistics.

7.4.3 Subscriber

These statistics are related to an individual mobile subscriber. An example would be bytes transferred in an SGSN PDP context.

This level also includes global and peer-based statistics.

7.5 Stats Reporter

The stats reporter periodically collects osmo counter, rate counter and stat item values and sends them to a backend. Currently implemented are outputting to the configured log targets and a statsd connector.

7.5.1 Configuring a stats reporter

Periodically printing the statistics to the log can be done in the following way:

Example 7.1 Log statistics

```
OsmoBSC> enable
OsmoBSC# configure terminal
OsmoBSC(config)# stats interval 60 ❶
OsmoBSC(config)# stats reporter log ❷
OsmoBSC(config-stats)# level global ❸
OsmoBSC(config-stats)# enable ❹
```

❶ The interval determines how often the statistics are reported.

❷ Write the statistic information to any configured log target.

❸ Report only `global` statistics (can be `global`, `peer`, or `subscriber`).

❹ Enable the reporter, `disable` will disable it again.

The counter values can also be sent to any aggregation/visualization tool that understands the statsd format, for example a statsd server with graphite or prometheus using the statsd_exporter together with grafana.

The statsd format is specified in https://github.com/b/statsd_spec

Example 7.2 Report statistics to statsd

```
OsmoBSC> enable
OsmoBSC# configure terminal
OsmoBSC(config)# stats interval 10
OsmoBSC(config)# stats reporter statsd ❶
OsmoBSC(config-stats)# prefix BSC1 ❷
OsmoBSC(config-stats)# level subscriber ❸
OsmoBSC(config-stats)# remote-ip 1.2.3.4 ❹
OsmoBSC(config-stats)# remote-port 8125 ❺
OsmoBSC(config-stats)# enable
```

❶ Configure the statsd reporter.

❷ Prefix the reported statistics. This is useful to distinguish statistics from multiple instances of the same service.

❸ Report only `global` statistics or include `peer` or `subscriber` statistics as well.

❹ IP address of the statsd server.

❺ UDP port of the statsd server. Statsd by default listens to port 8125.

You can use Netdata (https://learn.netdata.cloud/) as a statsd server which does not require special configuration to show rate counters. By default all the rate counters will be exposed to the StatsD plugin (listening on 127.0.0.1:8125) and displayed on the Netdata dashboard available via: http://localhost:19999 The list of available charts which includes all the rate counters reported via statsD is available through: http://localhost:19999/api/v1/charts

7.6 Socket stats

libosmocore provides features to monitor the status of TCP connections. This can be a helpful source of information when the links between network components are unreliable (e.g. satellite link between BTS and BSC).

Note
This feature is only available for certain types of TCP connections. At the moment only RSL/OML connections between OsmoBSC and the connected BTSs can be monitored.

7.6.1 Configuration

The gathering of the TCP connection statistics is done via syscalls. This has to be taken into account for the configuration. Since syscalls are rather expensive and time consuming the overall performance of the application may suffer when many TCP connections are present. This may be the case for BSCs with a large number of BTSs connected to it.

The statistics are gathered in batches per interval. A batch size of 5 would mean that only 5 TCP connections per interval are evaluated and the next 5 connections in the next interval and so on.

It is recommended to choose a large reporting interval and a reasonable small batch size to distribute the syscall load as even as possible.

Example 7.3 Report statistics to statsd

```
OsmoBSC> enable
OsmoBSC# configure terminal
stats-tcp interval 10 ❶
stats-tcp batch-size 5 ❷
```

❶ Set the gathering interval (sec.)

❷ Set how many TCP sockets statistics to gather per interval.

7.6.2 Generated stats items

Name	Description
tcp:unacked	unacknowledged packets.
tcp:lost	unacknowledged packets.
tcp:retrans	lost packets.
tcp:rtt	retransmitted packets.
tcp:rcv_rtt	roundtrip-time (receive).
tcp:notsent_bytes	bytes not yet sent.
tcp:rwnd_limited	time (usec) limited by receive window.
tcp:sndbuf_limited	Time (usec) limited by send buffer.
tcp:reord_seen	reordering events seen.

The item group index is the file descriptor number. The item group name consists of a static prefix (e.g. "ipa-rsl"), followed by the IP addresses and ports of both peers.

Example 7.4 VTY output of a stats item group of a TCP connection

```
stats tcp (15)('ipa-rsl,r=10.9.1.143:38455<->l=10.9.1.162:3003'):
 unacknowledged packets:        0
 lost packets:        0
 retransmitted packets:        0
 roundtrip-time:      583
 roundtrip-time (receive):        0
 bytes not yet sent:        0
 time (usec) limited by receive window:        0
 Time (usec) limited by send buffer:        0
 reordering events seen:        0
```

8 Counters

These counters and their description based on OsmoMSC 1.4.0 (OsmoMSC).

8.1 Rate Counters

Table 2: msc - mobile switching center

Name	Reference	Description
loc_update_type:attach	[?]	Received Location Update (IMSI Attach) requests.
loc_update_type:normal	[?]	Received Location Update (LAC change) requests.
loc_update_type:periodic	[?]	Received (periodic) Location Update requests.
loc_update_type:detach	[?]	Received IMSI Detach indications.
loc_update_resp:failed	[?]	Rejected Location Updates requests.
loc_update_resp:completed	[?]	Successful Location Update procedures.
cm_service_request:rejected	[?]	Rejected CM Service Requests.
cm_service_request:accepted	[?]	Accepted CM Service Requests.
paging_resp:rejected	[?]	Rejected Paging Responses.
paging_resp:accepted	[?]	Accepted Paging Responses.
sms:submitted	[?]	Total MO SMS received from the MS.
sms:no_receiver	[?]	Failed MO SMS delivery attempts (no receiver found).
sms:deliver_unknown_error	[?]	Failed MO SMS delivery attempts (other reason).
sms:delivered	[?]	Total MT SMS delivery attempts.
sms:rp_err_mem	[?]	Failed MT SMS delivery attempts (no memory).
sms:rp_err_other	[?]	Failed MT SMS delivery attempts (other reason).
call:mo_setup	[?]	Received MO SETUP messages (MO call establishment).
call:mo_connect_ack	[?]	Received MO CONNECT messages (MO call establishment).
call:mt_setup	[?]	Sent MT SETUP messages (MT call establishment).
call:mt_connect	[?]	Sent MT CONNECT messages (MT call establishment).

Table 2: (continued)

Name	Reference	Description
call:active	[?]	Calls that ever reached the active state.
call:complete	[?]	Calls terminated by DISCONNECT message after reaching the active state.
call:incomplete	[?]	Calls terminated by any other reason after reaching the active state.
nc_ss:mo_requests	[?]	Received MS-initiated call independent SS/USSD requests.
nc_ss:mo_established	[?]	Established MS-initiated call independent SS/USSD sessions.
nc_ss:mt_requests	[?]	Received network-initiated call independent SS/USSD requests.
nc_ss:mt_established	[?]	Established network-initiated call independent SS/USSD sessions.
bssmap:cipher_mode_reject	[?]	Number of CIPHER MODE REJECT messages processed by BSSMAP layer
bssmap:cipher_mode_complete	[?]	Number of CIPHER MODE COMPLETE messages processed by BSSMAP layer

9 Osmo Stat Items

10 Osmo Counters

Table 3: ungrouped osmo counters

Name	Reference	Description
msc.active_calls	[?]	
msc.active_nc_ss	[?]	

11 The Osmocom VTY Interface

All human interaction with Osmocom software is typically performed via an interactive command-line interface called the *VTY*.

Note
Integration of your programs and scripts should **not** be done via the telnet VTY interface, which is intended for human interaction only: the VTY responses may arbitrarily change in ways obvious to humans, while your scripts' parsing will likely break often. For external software to interact with Osmocom programs (besides using the dedicated protocols), it is strongly recommended to use the Control interface instead of the VTY, and to actively request / implement the Control interface commands as required for your use case.

The interactive telnet VTY is used to

- explore the current status of the system, including its configuration parameters, but also to view run-time state and statistics,
- review the currently active (running) configuration,
- perform interactive changes to the configuration (for those items that do not require a program restart),
- store the current running configuration to the config file,
- enable or disable logging; to the VTY itself or to other targets.

The Virtual Tele Type (VTY) has the concept of *nodes* and *commands*. Each command has a name and arguments. The name may contain a space to group several similar commands into a specific group. The arguments can be a single word, a string, numbers, ranges or a list of options. The available commands depend on the current node. there are various keyboard shortcuts to ease finding commands and the possible argument values.

Configuration file parsing during program start is actually performed the VTY's CONFIG node, which is also available in the telnet VTY. Apart from that, the telnet VTY features various interactive commands to query and instruct a running Osmocom program. A main difference is that during config file parsing, consistent indenting of parent vs. child nodes is required, while the interactive VTY ignores indenting and relies on the *exit* command to return to a parent node.

Note
In the *CONFIG* node, it is not well documented which commands take immediate effect without requiring a program restart. To save your current config with changes you may have made, you may use the `write file` command to **overwrite** your config file with the current configuration, after which you should be able to restart the program with all changes taking effect.

This chapter explains most of the common nodes and commands. A more detailed list is available in various programs' VTY reference manuals, e.g. see [vty-ref-osmomsc].

There are common patterns for the parameters, these include IPv4 addresses, number ranges, a word, a line of text and choice. The following will explain the commonly used syntactical patterns:

Table 4: VTY Parameter Patterns

Pattern	Example	Explanation
`A.B.C.D`	`127.0.0.1`	An IPv4 address
`A.B.C.D/M`	`192.168.1.0/24`	An IPv4 address and mask
`X:X::X:X`	`::1`	An IPv6 address
`X:X::X:X/M`	`::1/128`	An IPv6 address and mask
`TEXT`	`example01`	A single string without any spaces, tabs
`.TEXT`	`Some information`	A line of text
`(OptionA|OptionB|OptionC)`	`OptionA`	A choice between a list of available options
`<0-10>`	`5`	A number from a range

11.1 Accessing the telnet VTY

The VTY of a given Osmocom program is implemented as a telnet server, listening to a specific TCP port.

Please see Appendix A to check for the default TCP port number of the VTY interface of the specific Osmocom software you would like to connect to.

As telnet is insecure and offers neither strong authentication nor encryption, the VTY by default only binds to localhost (127.0.0.1) and will thus not be reachable by other hosts on the network.

Warning
By default, any user with access to the machine running the Osmocom software will be able to connect to the VTY. We assume that such systems are single-user systems, and anyone with local access to the system also is authorized to access the VTY. If you require stronger security, you may consider using the packet filter of your operating system to restrict access to the Osmocom VTY ports further.

11.2 VTY Nodes

The VTY by default has the following minimal nodes:

VIEW
When connecting to a telnet VTY, you will be on the *VIEW* node. As its name implies, it can only be used to view the system status, but it does not provide commands to alter the system state or configuration. As long as you are in the non-privileged *VIEW* node, your prompt will end in a > character.

ENABLE
The *ENABLE* node is entered by the `enable` command, from the *VIEW* node. Changing into the *ENABLE* node will unlock all kinds of commands that allow you to alter the system state or perform any other change to it. The *ENABLE* node and its children are signified by a # character at the end of your prompt.
You can change back from the *ENABLE* node to the *VIEW* node by using the `disable` command.

CONFIG
The *CONFIG* node is entered by the `configure terminal` command from the *ENABLE* node. The config node is used to change the run-time configuration parameters of the system. The prompt will indicate that you are in the config node by a `(config)#` prompt suffix.
You can always leave the *CONFIG* node or any of its children by using the `end` command.
This node is also automatically entered at the time the configuration file is read. All configuration file lines are processed as if they were entered from the VTY *CONFIG* node at start-up.

Other
Depending on the specific Osmocom program you are running, there will be few or more other nodes, typically below the *CONFIG* node. For example, the OsmoBSC has nodes for each BTS, and within the BTS node one for each TRX, and within the TRX node one for each Timeslot.

11.3 Interactive help

The VTY features an interactive help system, designed to help you to efficiently navigate is commands.

Note
The VTY is present on most Osmocom GSM/UMTS/GPRS software, thus this chapter is present in all the relevant manuals. The detailed examples below assume you are executing them on the OsmoMSC VTY. They will work in similar fashion on the other VTY interfaces, while the node structure will differ in each program.

11.3.1 The question-mark (?) command

If you type a single `?` at the prompt, the VTY will display possible completions at the exact location of your currently entered command.

If you type `?` at an otherwise empty command (without having entered even only a partial command), you will get a list of the first word of all possible commands available at this node:

Example: Typing ? at start of OsmoMSC prompt

```
OsmoMSC> ❶
  show        Show running system information
  list        Print command list
  exit        Exit current mode and down to previous mode
  help        Description of the interactive help system
  enable      Turn on privileged mode command
  terminal    Set terminal line parameters
  who         Display who is on vty
  logging     Configure logging
  no          Negate a command or set its defaults
  sms         SMS related commands
  subscriber  Operations on a Subscriber
```

❶ Type `?` here at the prompt, the `?` itself will not be printed.

If you have already entered a partial command, `?` will help you to review possible options of how to continue the command. Let's say you remember that `show` is used to investigate the system status, but you don't remember the exact name of the object. Hitting `?` after typing `show` will help out:

Example: Typing ? after a partial command

```
OsmoMSC> show ❶
  version          Displays program version
  online-help      Online help
  history          Display the session command history
  cs7              ITU-T Signaling System 7
  logging          Show current logging configuration
  alarms           Show current logging configuration
  talloc-context   Show talloc memory hierarchy
  stats            Show statistical values
  asciidoc         Asciidoc generation
  rate-counters    Show all rate counters
  fsm              Show information about finite state machines
  fsm-instances    Show information about finite state machine instances
  sgs-connections  Show SGs interface connections / MMEs
  subscriber       Operations on a Subscriber
  bsc              BSC
  connection       Subscriber Connections
  transaction      Transactions
  statistics       Display network statistics
  sms-queue        Display SMSqueue statistics
  smpp          SMPP Interface
```

❶ Type `?` after the `show` command, the `?` itself will not be printed.

You may pick the `bsc` object and type `?` again:

Example: Typing ? after show bsc

```
OsmoMSC> show bsc
  <cr>
```

By presenting `<cr>` as the only option, the VTY tells you that your command is complete without any remaining arguments being available, and that you should hit enter, a.k.a. "carriage return".

11.3.2 TAB completion

The VTY supports tab (tabulator) completion. Simply type any partial command and press `<tab>`, and it will either show you a list of possible expansions, or completes the command if there's only one choice.

Example: Use of <tab> pressed after typing only s as command

```
OsmoMSC> s❶
show       sms        subscriber
```

❶ Type `<tab>` here.

At this point, you may choose `show`, and then press `<tab>` again:

Example: Use of <tab> pressed after typing show command

```
OsmoMSC> show ❶
version    online-help history    cs7        logging    alarms
talloc-context stats      asciidoc   rate-counters fsm        fsm-instances
sgs-connections subscriber bsc        connection transaction statistics
sms-queue smpp
```

❶ Type `<tab>` here.

11.3.3 The `list` command

The `list` command will give you a full list of all commands and their arguments available at the current node:

Example: Typing list at start of OsmoMSC VIEW node prompt

```
OsmoMSC> list
  show version
  show online-help
  list
  exit
  help
  enable
  terminal length <0-512>
  terminal no length
  who
  show history
  show cs7 instance <0-15> users
  show cs7 (sua|m3ua|ipa) [<0-65534>]
  show cs7 instance <0-15> asp
  show cs7 instance <0-15> as (active|all|m3ua|sua)
  show cs7 instance <0-15> sccp addressbook
  show cs7 instance <0-15> sccp users
  show cs7 instance <0-15> sccp ssn <0-65535>
  show cs7 instance <0-15> sccp connections
  show cs7 instance <0-15> sccp timers
  logging enable
  logging disable
  logging filter all (0|1)
  logging color (0|1)
  logging timestamp (0|1)
  logging print extended-timestamp (0|1)
  logging print category (0|1)
  logging print category-hex (0|1)
  logging print level (0|1)
  logging print file (0|1|basename) [last]
```

```
  logging set-log-mask MASK
  logging level (rll|cc|mm|rr|mncc|pag|msc|mgcp|ho|db|ref|ctrl|smpp|ranap|vlr|iucs|bssap| ↩
      sgs|lglobal|llapd|linp|lmux|lmi|lmib|lsms|lctrl|lgtp|lstats|lgsup|loap|lss7|lsccp|lsua ↩
      |lm3ua|lmgcp|ljibuf|lrspro) (debug|info|notice|error|fatal)
  logging level set-all (debug|info|notice|error|fatal)
  logging level force-all (debug|info|notice|error|fatal)
  no logging level force-all
  show logging vty
  show alarms
  show talloc-context (application|all) (full|brief|DEPTH)
  show talloc-context (application|all) (full|brief|DEPTH) tree ADDRESS
  show talloc-context (application|all) (full|brief|DEPTH) filter REGEXP
  show stats
  show stats level (global|peer|subscriber)
  show asciidoc counters
  show rate-counters
  show fsm NAME
  show fsm all
  show fsm-instances NAME
  show fsm-instances all
  show sgs-connections
  show subscriber (msisdn|extension|imsi|tmsi|id) ID
  show subscriber cache
  show bsc
  show connection
  show transaction
  sms send pending
  sms delete expired
  subscriber create imsi ID
  subscriber (msisdn|extension|imsi|tmsi|id) ID sms sender (msisdn|extension|imsi|tmsi|id)  ↩
      SENDER_ID send .LINE
  subscriber (msisdn|extension|imsi|tmsi|id) ID silent-sms sender (msisdn|extension|imsi| ↩
      tmsi|id) SENDER_ID send .LINE
  subscriber (msisdn|extension|imsi|tmsi|id) ID silent-call start (any|tch/f|tch/any|sdcch)
  subscriber (msisdn|extension|imsi|tmsi|id) ID silent-call stop
  subscriber (msisdn|extension|imsi|tmsi|id) ID ussd-notify (0|1|2) .TEXT
  subscriber (msisdn|extension|imsi|tmsi|id) ID ms-test close-loop (a|b|c|d|e|f|i)
  subscriber (msisdn|extension|imsi|tmsi|id) ID ms-test open-loop
  subscriber (msisdn|extension|imsi|tmsi|id) ID paging
  show statistics
  show sms-queue
  logging filter imsi IMSI
  show smpp esme
```

Tip
Remember, the list of available commands will change significantly depending on the Osmocom program you are accessing, its software version and the current node you're at. Compare the above example of the OsmoMSC *VIEW* node with the list of the OsmoMSC *NETWORK* config node:

Example: Typing list at start of OsmoMSC NETWORK config node prompt

```
OsmoMSC(config-net)# list
  help
  list
  write terminal
  write file
  write memory
  write
  show running-config
```

```
  exit
  end
  network country code <1-999>
  mobile network code <0-999>
  short name NAME
  long name NAME
  encryption a5 <0-3> [<0-3>] [<0-3>] [<0-3>]
  authentication (optional|required)
  rrlp mode (none|ms-based|ms-preferred|ass-preferred)
  mm info (0|1)
  timezone <-19-19> (0|15|30|45)
  timezone <-19-19> (0|15|30|45) <0-2>
  no timezone
  periodic location update <6-1530>
  no periodic location update
```

11.3.4 The attribute system

The VTY allows to edit the configuration at runtime. For many VTY commands the configuration change is immediately valid but for some commands a change becomes valid on a certain event only. In some cases it is even necessary to restart the whole process.

To give the user an overview, which configuration change applies when, the VTY implemets a system of attribute flags, which can be displayed using the `show` command with the parameter `vty-attributes`

Example: Typing show vty-attributes at the VTY prompt

```
OsmoBSC> show vty-attributes
  Global attributes:
    ^  This command is hidden (check expert mode)
    !  This command applies immediately
    @  This command applies on VTY node exit
  Library specific attributes:
    A  This command applies on ASP restart
    I  This command applies on IPA link establishment
    L  This command applies on E1 line update
  Application specific attributes:
    o  This command applies on A-bis OML link (re)establishment
    r  This command applies on A-bis RSL link (re)establishment
    l  This command applies for newly created lchans
```

The attributes are symbolized through a single ASCII letter (flag) and do exist in three levels. This is more or less due to the technical aspects of the VTY implementation. For the user, the level of an attribute has only informative purpose.

The global attributes, which can be found under the same attribute letter in every osmocom application, exist on the top level. The Library specific attributes below are used in various osmocom libraries. Like with the global attributes the attribute flag letter stays the same throughout every osmocom application here as well. On the third level one can find the application specific attributes. Those are unique to each osmocom application and the attribute letters may have different meanings in different osmocom applications. To make the user more aware of this, lowercase letters were used as attribute flags.

The `list` command with the parameter `with-flags` displays a list of available commands on the current VTY node, along with attribute columns on the left side. Those columns contain the attribute flag letters to indicate to the user how the command behaves in terms of how and when the configuration change takes effect.

Example: Typing list with-flags at the VTY prompt

```
OsmoBSC(config-net-bts)# list with-flags
  . ...  help
  . ...  list [with-flags]
  . ...  show vty-attributes
  . ...  show vty-attributes (application|library|global)
```

```
. ...  write terminal
. ...  write file [PATH]
. ...  write memory
. ...  write
. ...  show running-config ❶
. ...  exit
. ...  end
. o..  type (unknown|bs11|nanobts|rbs2000|nokia_site|sysmobts) ❷
. ...  description .TEXT
. ...  no description
. o..  band BAND
. .r.  cell_identity <0-65535> ❸
. .r.  dtx uplink [force]
. .r.  dtx downlink
. .r.  no dtx uplink
. .r.  no dtx downlink
. .r.  location_area_code <0-65535>
. o..  base_station_id_code <0-63>
. o..  ipa unit-id <0-65534> <0-255>
. o..  ipa rsl-ip A.B.C.D
. o..  nokia_site skip-reset (0|1)
! ...  nokia_site no-local-rel-conf (0|1) ❹
! ...  nokia_site bts-reset-timer  <15-100> ❺
```

❶ This command has no attributes assigned.

❷ This command applies on A-bis OML link (re)establishment.

❸ This command applies on A-bis RSL link (re)establishment.

❹, ❺ This command applies immediately.

There are multiple columns because a single command may be associated with multiple attributes at the same time. To improve readability each flag letter gets a dedicated column. Empty spaces in the column are marked with a dot (".")

In some cases the listing will contain commands that are associated with no flags at all. Those commands either play an exceptional role (interactive commands outside "configure terminal", vty node navigation commands, commands to show / write the config file) or will require a full restart of the overall process to take effect.

11.3.5 The expert mode

Some VTY commands are considered relatively dangerous if used in production operation, so the general approach is to hide them. This means that they don't show up anywhere but the source code, but can still be executed. On the one hand, this approach reduces the risk of an accidental invocation and potential service degradation; on the other, it complicates intentional use of the hidden commands.

The VTY features so-called *expert* mode, that makes the hidden commands appear in the interactive help, as well as in the XML VTY reference, just like normal ones. This mode can be activated from the *VIEW* node by invoking the `enable` command with the parameter `expert-mode`. It remains active for the individual VTY session, and gets disabled automatically when the user switches back to the *VIEW* node or terminates the session.

A special attribute in the output of the `list with-flags` command indicates whether a given command is hidden in normal mode, or is a regular command:

Example: Hidden commands in the output of the list with-flags command

```
OsmoBSC> enable expert-mode ❶
OsmoBSC# list with-flags
  ...
  ^   bts <0-255> (activate-all-lchan|deactivate-all-lchan) ❷
  ^   bts <0-255> trx <0-255> (activate-all-lchan|deactivate-all-lchan) ❸
```

```
.   bts <0-255> trx <0-255> timeslot <0-7> sub-slot <0-7> mdcx A.B.C.D <0-65535> ❹
^   bts <0-255> trx <0-255> timeslot <0-7> sub-slot <0-7> (borken|unused) ❺
.   bts <0-255> trx <0-255> timeslot <0-7> sub-slot <0-7> handover <0-255> ❻
.   bts <0-255> trx <0-255> timeslot <0-7> sub-slot <0-7> assignment ❼
.   bts <0-255> smscb-command (normal|schedule|default) <1-4> HEXSTRING ❽
...
```

❶ This command enables the *expert* mode.

❷, ❸, ❺ This is a hidden command (only shown in the *expert* mode).

❹, ❻, ❼, ❽ This is a regular command that is always shown regardless of the mode.

12 libosmocore Logging System

In any reasonably complex software it is important to understand how to enable and configure logging in order to get a better insight into what is happening, and to be able to follow the course of action. We therefore ask the reader to bear with us while we explain how the logging subsystem works and how it is configured.

Most Osmocom Software (like `osmo-bts`, `osmo-bsc`, `osmo-nitb`, `osmo-sgsn` and many others) uses the same common logging system.

This chapter describes the architecture and configuration of this common logging system.

The logging system is composed of

- log targets (where to log),
- log categories (who is creating the log line),
- log levels (controlling the verbosity of logging), and
- log filters (filtering or suppressing certain messages).

All logging is done in human-readable ASCII-text. The logging system is configured by means of VTY commands that can either be entered interactively, or read from a configuration file at process start time.

12.1 Log categories

Each sub-system of the program in question typically logs its messages as a different category, allowing fine-grained control over which log messages you will or will not see. For example, in OsmoBSC, there are categories for the protocol layers `rsl`, `rr`, `mm`, `cc` and many others. To get a list of categories interactively on the vty, type: `logging level ?`

12.2 Log levels

For each of the log categories (see Section 12.1), you can set an independent log level, controlling the level of verbosity. Log levels include:

fatal
: Fatal messages, causing abort and/or re-start of a process. This *shouldn't happen*.

error
: An actual error has occurred, its cause should be further investigated by the administrator.

notice
: A noticeable event has occurred, which is not considered to be an error.

info
Some information about normal/regular system activity is provided.

debug
Verbose information about internal processing of the system, used for debugging purpose. This will log the most.

The log levels are inclusive, e.g. if you select *info*, then this really means that all events with a level of at least *info* will be logged, i.e. including events of *notice*, *error* and *fatal*.

So for example, in OsmoBSC, to set the log level of the Mobility Management category to info, you can use the following command: `log level mm info`.

There is also a special command to set all categories as a one-off to a desired log level. For example, to silence all messages but those logged as notice and above issue the command: `log level set-all notice`

Afterwards you can adjust specific categories as usual.

A similar command is `log level force-all <level>` which causes all categories to behave as if set to log level <level> until the command is reverted with `no log level force-all` after which the individually-configured log levels will again take effect. The difference between `set-all` and `force-all` is that `set-all` actually changes the individual category settings while `force-all` is a (temporary) override of those settings and does not change them.

12.3 Log printing options

The logging system has various options to change the information displayed in the log message.

log color 1
With this option each log message will log with the color of its category. The color is hard-coded and can not be changed. As with other options a *0* disables this functionality.

log timestamp 1
Includes the current time in the log message. When logging to syslog this option should not be needed, but may come in handy when debugging an issue while logging to file.

log print extended-timestamp 1
In order to debug time-critical issues this option will print a timestamp with millisecond granularity.

log print category 1
Prefix each log message with the category name.

log print category-hex 1
Prefix each log message with the category number in hex (<*000b*>).

log print level 1
Prefix each log message with the name of the log level.

log print file 1
Prefix each log message with the source file and line number. Append the keyword `last` to append the file information instead of prefixing it.

12.4 Log filters

The default behavior is to filter out everything, i.e. not to log anything. The reason is quite simple: On a busy production setup, logging all events for a given subsystem may very quickly be flooding your console before you have a chance to set a more restrictive filter.

To request no filtering, i.e. see all messages, you may use: `log filter all 1`

In addition to generic filtering, applications can implement special log filters using the same framework to filter on particular context.

For example in OsmoBSC, to only see messages relating to a particular subscriber identified by his IMSI, you may use: `log filter imsi 262020123456789`

12.5 Log targets

Each of the log targets represent certain destination for log messages. It can be configured independently by selecting levels (see Section 12.2) for categories (see Section 12.1) as well as filtering (see Section 12.4) and other options like `logging timestamp` for example.

12.5.1 Logging to the VTY

Logging messages to the interactive command-line interface (VTY) is most useful for occasional investigation by the system administrator.

Logging to the VTY is disabled by default, and needs to be enabled explicitly for each such session. This means that multiple concurrent VTY sessions each have their own logging configuration. Once you close a VTY session, the log target will be destroyed and your log settings be lost. If you re-connect to the VTY, you have to again activate and configure logging, if you wish.

To create a logging target bound to a VTY, you have to use the following command: `logging enable` This doesn't really activate the generation of any output messages yet, it merely creates and attaches a log target to the VTY session. The newly-created target still doesn't have any filter installed, i.e. *all log messages will be suppressed by default*

Next, you can configure the log levels for desired categories in your VTY session. See Section 12.1 for more details on categories and Section 12.2 for the log level details.

For example, to set the log level of the Call Control category to debug, you can use: `log level cc debug`

Finally, after having configured the levels, you still need to set the filter as it's described in Section 12.4.

Tip
If many messages are being logged to a VTY session, it may be hard to impossible to still use the same session for any commands. We therefore recommend to open a second VTY session in parallel, and use one only for logging, while the other is used for interacting with the system. Another option would be to use different log target.

To review the current vty logging configuration, you can use: `show logging vty`

12.5.2 Logging to the ring buffer

To avoid having separate VTY session just for logging output while still having immediate access to them, one can use `alarms` target. It lets you store the log messages inside the ring buffer of a given size which is available with `show alarms` command.

It's configured as follows:

```
OsmoBSC> enable
OsmoBSC# configure terminal
OsmoBSC(config)# log alarms 98
OsmoBSC(config-log)#
```

In the example above 98 is the desired size of the ring buffer (number of messages). Once it's filled, the incoming log messages will push out the oldest messages available in the buffer.

12.5.3 Logging via gsmtap

GSMTAP is normally a pseudo-header format that enables the IP-transport of GSM (or other telecom) protocols that are not normally transported over IP. For example, the most common situation is to enable GSMTAP in OsmoBTS or OsmoPCU to provide GSM-Um air interface capture files over IP, so they can be analyzed in wireshark.

GSMTAP logging is now a method how Osmocom software can also encapsulate its own log output in GSMTAP frames. We're not trying to re-invent rsyslog here, but this is very handy When debugging complex issues. It enables the reader of the pcap file

containing GSMTAP logging together with other protocol traces to reconstruct exact chain of events. A single pcap file can then contain both the log output of any number of Osmocom programs in the same timeline of the messages on various interfaces in and out of said Osmocom programs.

It's configured as follows:

```
OsmoBSC> enable
OsmoBSC# configure terminal
OsmoBSC(config)# log gsmtap 192.168.2.3
OsmoBSC(config-log)#
```

The hostname/ip argument is optional: if omitted the default 127.0.0.1 will be used. The log strings inside GSMTAP are already supported by Wireshark. Capturing for `port 4729` on appropriate interface will reveal log messages including source file name and line number as well as application. This makes it easy to consolidate logs from several different network components alongside the air frames. You can also use Wireshark to quickly filter logs for a given subsystem, severity, file name etc.

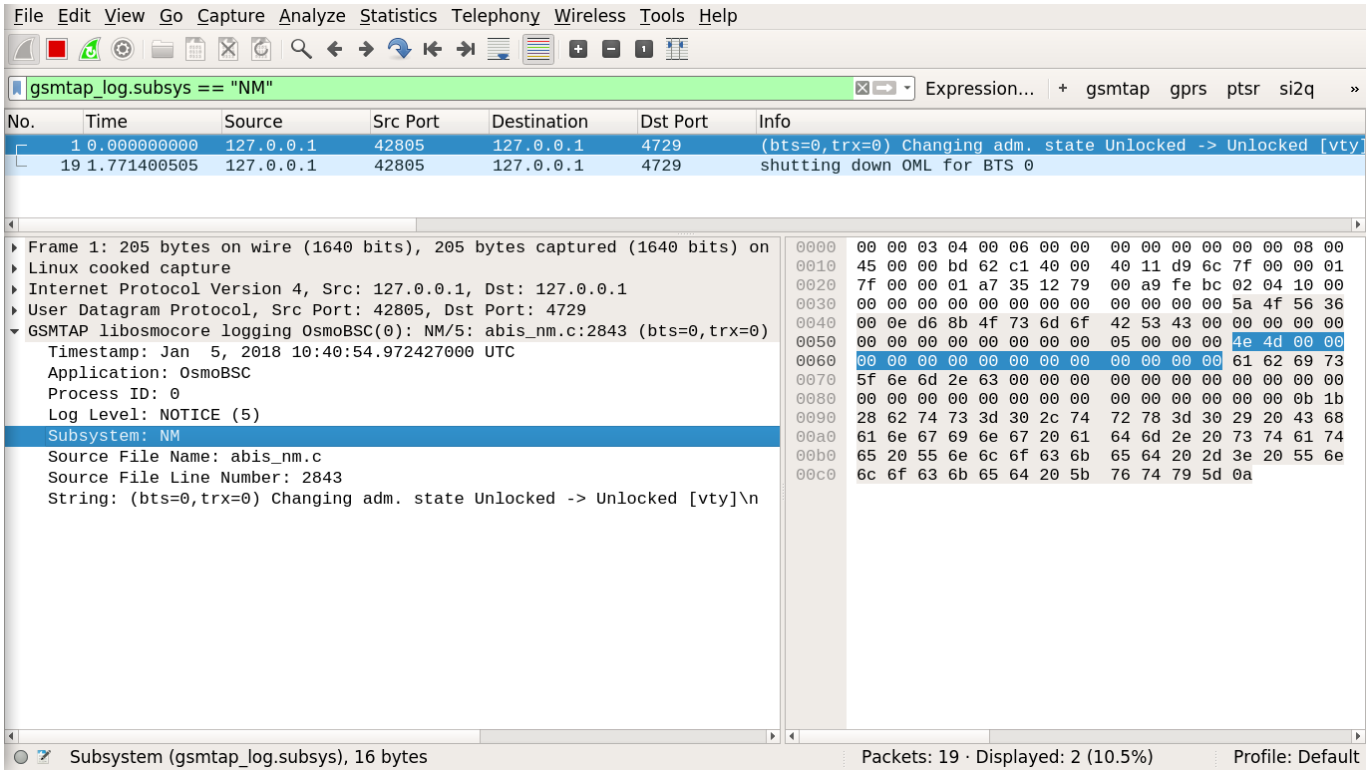


Figure 2: Wireshark with logs delivered over GSMTAP

Note: the logs are also duplicated to stderr when GSMTAP logging is configured because stderr is the default log target which is initialized automatically. To descrease stderr logging to absolute minimum, you can configure it as follows:

```
OsmoBSC> enable
OsmoBSC# configure terminal
OsmoBSC(config)# log stderr
OsmoBSC(config-log)# logging level force-all fatal
```

Note
Every time you generate GSMTAP messages and send it to a unicast (non-broadcast/multicast) IP address, please make sure that the destination IP address actually has a socket open on the specified port, or drops the packets in its packet filter. If unicast GSMTAP messages arrive at a closed destination UDP port, the operating system will likely generate ICMP port unreachable messages. Those ICMP messages in turn will, when arriving at the source (the host on which you run the Osmocom software sending GSMTAP), suppress generation of further GSMTAP messages for some time, resulting in incomplete files. In case of doubt, either send GSMTAP to multicast IP addresses, or run something like `nc -l -u -p 4729 > /dev/null` on the destination host to open the socket at the GSMTAP port and discard anything arriving at it.

12.5.4 Logging to a file

As opposed to Logging to the VTY, logging to files is persistent and stored in the configuration file. As such, it is configured in sub-nodes below the configuration node. There can be any number of log files active, each of them having different settings regarding levels / subsystems.

To configure a new log file, enter the following sequence of commands:

```
OsmoBSC> enable
OsmoBSC# configure terminal
OsmoBSC(config)# log file /path/to/my/file
OsmoBSC(config-log)#
```

This leaves you at the config-log prompt, from where you can set the detailed configuration for this log file. The available commands at this point are identical to configuring logging on the VTY, they include `logging filter`, `logging level` as well as `logging color` and `logging timestamp`.

Tip
Don't forget to use the `copy running-config startup-config` (or its short-hand `write file`) command to make your logging configuration persistent across application re-start.

Note
libosmocore provides file close-and-reopen support by SIGHUP, as used by popular log file rotating solutions such as https://github.com/logrotate/logrotate found in most GNU/Linux distributions.

12.5.5 Logging to syslog

syslog is a standard for computer data logging maintained by the IETF. Unix-like operating systems like GNU/Linux provide several syslog compatible log daemons that receive log messages generated by application programs.

libosmocore based applications can log messages to syslog by using the syslog log target. You can configure syslog logging by issuing the following commands on the VTY:

```
OsmoBSC> enable
OsmoBSC# configure terminal
OsmoBSC(config)# log syslog daemon
OsmoBSC(config-log)#
```

This leaves you at the config-log prompt, from where you can set the detailed configuration for this log file. The available commands at this point are identical to configuring logging on the VTY, they include `logging filter`, `logging level` as well as `logging color` and `logging timestamp`.

Note
Syslog daemons will normally automatically prefix every message with a time-stamp, so you should disable the libosmocore time-stamping by issuing the `logging timestamp 0` command.

12.5.6 Logging to systemd-journal

systemd has been adopted by the majority of modern GNU/Linux distributions. Along with various daemons and utilities it provides systemd-journald [1] - a daemon responsible for event logging (syslog replacement). libosmocore based applications can log messages directly to systemd-journald.

The key difference from other logging targets is that systemd based logging allows to offload rendering of the meta information, such as location (file name, line number), subsystem, and logging level, to systemd-journald. Furthermore, systemd allows to attach arbitrary meta fields to the logging messages [2], which can be used for advanced log filtering.

[1] https://www.freedesktop.org/software/systemd/man/systemd-journald.service.html [2] https://www.freedesktop.org/software/-systemd/man/systemd.journal-fields.html

It was decided to introduce libsystemd as an optional dependency, so it needs to be enabled explicitly at configure/build time:

```
$ ./configure --enable-systemd-logging
```

Note
Recent libosmocore packages provided by Osmocom for Debian and CentOS are compiled **with** libsystemd (https://gerrit.osmocom.org/c/libosmocore/+/22651).

You can configure systemd based logging in two ways:

Example: systemd-journal target with offloaded rendering

```
log systemd-journal raw ❶
 logging filter all 1
 logging level set-all notice
```

❶ `raw` logging handler, rendering offloaded to systemd.

In this example, logging messages will be passed to systemd without any meta information (time, location, level, category) in the text itself, so all the printing parameters like `logging print file` will be ignored. Instead, the meta information is passed separately as *fields* which can be retrieved from the journal and rendered in any preferred way.

```
# Show Osmocom specific fields
$ journalctl --fields | grep OSMO

# Filter messages by logging subsystem at run-time
$ journalctl OSMO_SUBSYS=DMSC -f

# Render specific fields only
$ journalctl --output=verbose \
        --output-fields=SYSLOG_IDENTIFIER,OSMO_SUBSYS,CODE_FILE,CODE_LINE,MESSAGE
```

See `man 7 systemd.journal-fields` for a list of default fields, and `man 1 journalctl` for general information and available formatters.

Example: systemd-journal target with libosmocore based rendering

```
log systemd-journal ❶
 logging filter all 1
 logging print file basename
 logging print category-hex 0
 logging print category 1
 logging print level 1
 logging timestamp 0 ❷
 logging color 1 ❸
 logging level set-all notice
```

1. Generic logging handler, rendering is done by libosmocore.
2. Disable timestamping, systemd will timestamp every message anyway.
3. Colored messages can be rendered with `journalctl --output=cat`.

In this example, logging messages will be pre-processed by libosmocore before being passed to systemd. No additional fields will be attached, except the logging level (`PRIORITY`). This mode is similar to *syslog* and *stderr*.

12.5.7 Logging to stderr

If you're not running the respective application as a daemon in the background, you can also use the stderr log target in order to log to the standard error file descriptor of the process.

In order to configure logging to stderr, you can use the following commands:

```
OsmoBSC> enable
OsmoBSC# configure terminal
OsmoBSC(config)# log stderr
OsmoBSC(config-log)#
```

13 Osmocom SS7 + SIGTRAN support

13.1 History / Background

If you're upgrading from earlier releases of the Osmocom stack, this section will give you some background about the evolution.

13.1.1 The Past (before 2017)

In the original implementation of the GSM BSC inside Osmocom (the OsmoBSC program, part of OpenBSC), no SS7 support was included.

This is despite the fact that ETSI/3GPP mandated the use of SCCP over MTP over E1/T1 TDM lines for the A interface at that time.

Instead of going down to the TDM based legacy physical layers, OsmoBSC implemented something called an IPA multiplex, which apparently some people also refer to as SCCPlite. We have never seen any specifications for this interface, but implemented it from scratch using protocol traces.

The IPA protocol stack is based on a minimal sub-set of SCCP (including connection oriented SCCP) wrapped into a 3-byte header to packetize a TCP stream.

The IPA/SCCPlite based A interface existed at a time when the ETSI/3GPP specifications did not offer any IP based transport for the A interface. An official as added only in Release FIXME of the 3GPP specifications.

The A interface BSSMAP protocol refers to voice circuits (E1/T1 timeslots) using circuit identity codes (CICs). As there are no physical timeslots on a TCP/IP based transport layer, the CICs get mapped to RTP streams for circuit-switched data using out-of-band signaling via MGCP, the IETF-standardized Media Gateway Control Protocol.

13.1.2 The present (2017)

In 2017, sysmocom was tasked with implementing a 3GPP AoIP compliant A interface. This meant that a lot of things had to change in the existing code:

- removal of the existing hard-wired SCCPlite/IPA code from OsmoBSC
- introduction of a formal SCCP User SAP at the lower boundary of BSSMAP

- introduction of libosmo-sigtran, a comprehensive SS7 and SIGTRAN library which includes a SCCP implementation for connectionless and connection-oriented procedures, offering the SCCP User SAP towards BSSAP
- introduction of an A interface in OsmoMSC (which so far offered Iu only)
- port of the existing SUA-based IuCS and IuPS over to the SCCP User SAP of libosmo-sigtran.
- Implementation of ETSI M3UA as preferred/primary transport layer for SCCP
- Implementation of an IPA transport layer inside libosmo-sigtran, in order to keep backwards-compatibility.

This work enables the Osmocom universe to become more compliant with modern Releases of 3GPP specifications, which enables interoperability with other MSCs or even BSCs. However, this comes at a price: Increased complexity in set-up and configuration.

Using SS7 or SIGTRAN based transport of the A interface adds an entirely new domain that needs to be understood by system and network administrators setting up cellular networks based on Osmocom.

One of the key advantages of the Osmocom architecture with OsmoNITB was exactly this simplification and reduction of complexity, enabling more people to set-up and operate cellular networks.

So we have put some thought into how we can achieve compatibility with SS7/SIGTRAN and the 3GPP specifications, while at the same time enabling some degree of auto-configuration where a small network can be set up without too many configuration related to the signaling network. We have achieved this by "abusing" (or extending) the M3UA Routing Key Management slightly.

13.2 Osmocom extensions to SIGTRAN

Osmocom has implemented some extensions to the SIGTRAN protocol suite. Those extensions will be documented below.

13.2.1 Osmocom M3UA Routing Key Management Extensions

In classic M3UA, a peer identifies its remote peer based on IP address and port details. So once an ASP connects to an SG, the SG will check if there is any configuration that matches the source IP (and possibly source port) of that connection in order to understand which routing context is used - and subsequently which traffic is to be routed to this M3UA peer.

This is quite inflexible, as it means that every BSC in a GSM network needs to be manually pre-configured at the SG/STP, and that configuration on the BSC and MSC must match to enable communication.

M3UA specifies an optional Routing Key Management (RKM) sub-protocol. Using RKM, an ASP can dynamically tell the SG/STP, which traffic it wants to receive. However, the idea is still that the SG has some matching configuration.

In OsmoSTP based on libosmo-sigtran, we decided to (optionally) enable fully dynamic registration. This means that any ASP can simply connect to the SG and request the dynamic creation of an ASP and AS with a corresponding routing key for a given point code. As long as the SG doesn't already have a route to this requested point code, The SG will simply trust any ASP and set a corresponding route.

To enable dynamic creation of ASPs within an AS from any source IP/port, the corresponding xUA Server (Section 13.5) must be configured with `accept-asp-connections dynamic-permitted`.

To enable dynamic registration of routing keys via RKM, the corresponding SS7 Instance (Section 13.4) must be configured with `xua rkm routing-key-allocation dynamic-permitted`.

This is of course highly insecure and can only be used in trusted, internal networks. However, it is quite elegant in reducing the amount of configuration complexity. All that is needed, is that an unique point code is configured at each of the ASPs (application programs) that connect to the STP.

To put things more concretely: Each BSC and MSC connecting to OsmoSTP simply needs to be configured to have a different point code, and to know to which IP/port of the STP to connect. There's no other configuration required for a small, autonomous, self-contained network. OsmoSTP will automatically install ASP, AS and route definitions on demand, and route messages between all connected entities.

The same above of course also applies to HNB-GW and OsmoSGSN in the case of Iu interfaces.

13.2.2 IPA / SCCPlite backwards compatibility

The fundamental problem with IPA/SCCPlite is that there's no MTP routing label surrounding the SCCP message. This is generally problematic in the context of connection-oriented SCCP, as there is no addressing information inside the SCCP messages after the connection has been established. Instead, the messages are routed based on the MTP label, containing point codes established during connection set-up time.

This means that even if the SCCP messages did contain Called/Calling Party Addresses with point codes or global titles, it would only help us for routing connectionless SCCP. The A interface, however, is connection-oriented.

So in order to integrate IPA/SCCPlite with a new full-blown SS7/SIGTRAN stack, there are the following options:

1. implement SCCP connection coupling. This is something like a proxy for connection-oriented SCCP, and is what is used in SS7 to route beyond a given MTP network (e.g. at gateways between different MTP networks).
2. consider all SCCP messages to be destined for the local point code of the receiver. This then means that the SG functionality must be included inside the MSC, and the MSC be bound to the SSN on the local point code.
3. hard-code some DPC when receiving a message from an IPA connection. It could be any remote PC and we'd simply route the message towards that point code.

But then we also have the return direction:

1. We could "assign" a unique SPC to each connected IPA client (BSC), and then announce that PC towards the SS7 side. Return packets would then end up at our IPA-server-bearing STP, which forwards them to the respective IPA connection and thus BSC. On the transmit side, we'd simply strip the MTP routing label and send the raw SCCP message over IPA.
2. If the IPA server / SGW resides within the MSC, one could also have some kind of handle/reference to the specific TCP connection through which the BSC connected. All responses for a given peer would then have to be routed back to the same connection. This is quite ugly as it completely breaks the concepts of the SCCP User SAP, where a user has no information (nor to worry about) any "physical" signaling links.

13.3 Minimal Osmocom SIGTRAN configurations for small networks

If you're not an SS7 expert, and all you want is to run your own small self-contained cellular network, this section explains what you need to do.

In general, you can consider OsmoSTP as something like an IP router. On the application layer (in our case the BSSAP/BSSMAP or RANAP protocols between Radio Access Network and Core Network), it is completely invisible/transparent. The BSC connects via SCCP to the MSC. It doesn't know that there's an STP in between, and that this STP is performing some routing function. Compares this to your web browser not knowing about IP routers, it just establishes an http connection to a web server.

This is also why most GSM network architecture diagrams will not explicitly show an STP. It is not part of the cellular network. Rather, one or many STPs are part of the underlying SS7 signaling transport network, on top of which the cellular network elements are built.

13.3.1 A minimal 2G configuration to get started

You will be running the following programs:

- OsmoBSC as the base-station controller between your BTS (possibly running OsmoBTS) and the MSC
- OsmoMSC as the mobile switching center providing SMS and telephony service to your subscribers
- OsmoSTP as the signal transfer point, routing messages between one or more BSCs and the MSC

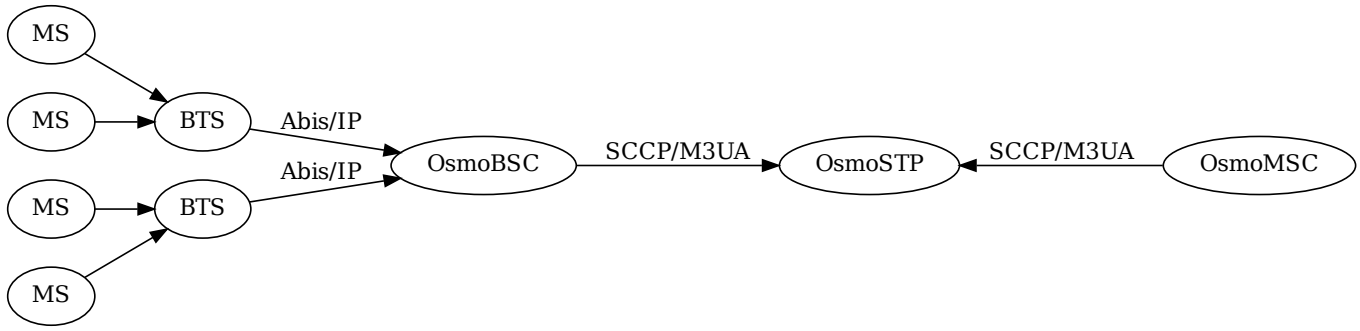


Figure 3: Simple signaling network for 2G (GSM)

You can use the OsmoSTP fully dynamic registration feature, so the BSCs and the MSC will simply register with their point codes to the STP, and the STP will create most configuration on the fly.

All you need to make sure is:

- to assign one unique point code to each BSC and MSC
- to point all BSCs and the MSC to connect to the IP+Port of the STP
- to configure the point code of the MSC in the BSCs

13.3.2 A minimal 3G configuration to get started

You will be running the following programs:

- OsmoHNBGW as the homeNodeB Gateway between your femtocells / small cells and the MSC+SGSN
- OsmoMSC as the mobile switching center providing SMS and telephony service to your subscribers
- OsmoSGSN as the Serving GPRS Support Node, providing packet data (internet) services to your subscribers
- OsmoSTP as the signal transfer point, routing messages between one or more HNBGWs and the MSC and SGSN

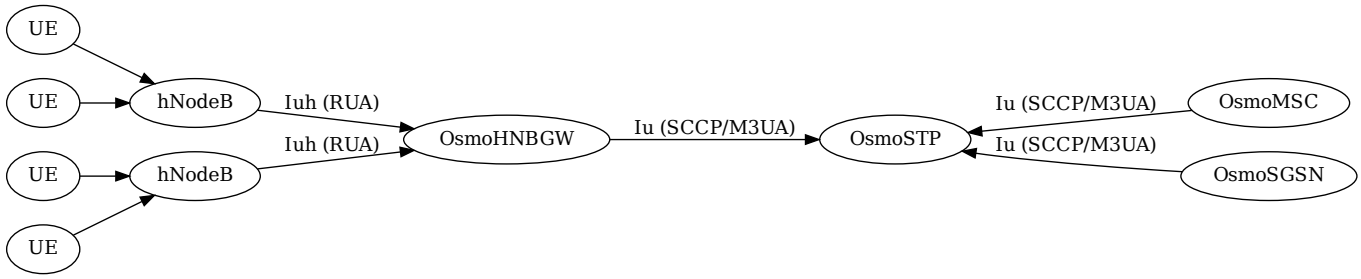


Figure 4: Simple signaling network for 3G (UMTS)

You can use the OsmoSTP fully dynamic registration feature, so the HNBGWs, the MSC and the SGSN will simply register with their point codes to the STP, and the STP will create most configuration on the fly.

All you need to make sure is:

- to assign one unique point code to each HNBGW, MSC and SGSN
- to point all HNBGWs and the MSC and SGSN to connect to the IP+Port of STP
- to configure the point code of the MSC in the HNBGWs
- to configure the point code of the SGSN in the HNBGWs

13.4 Osmocom SS7 Instances

The entire SS7 stack can be operated multiple times within one application/program by means of so-called SS7 Instances.

There can be any number of SS7 Instances, and each instance has its own set of XUA Servers, ASPs, ASs, Routes, etc.

Each SS7 Instance can have different point code formats / lengths.

Table 5: Major Attributes of an Osmocom SS7 Instance

Name	VTY Command	Description
ID	(config)# cs7 instance ID	The numeric identifier of this instance
Name	(config-cs7)# name NAME	A human-readable name for this instance
Description	(config-cs7)# description DESC	More verbose description
Primary PC	(config-cs7)# point-code PC	Primary local point code
Network Indicator	(config-cs7)# network-indicator	Network Indicator used in MTP3 Routing Label
Point Code Format	(config-cs7)# point-code format	Point Code Format (Default: 3.8.3)
Point Code Delimiter	(config-cs7)# point-code delimiter	Point Code Delimiter: . or -

13.5 Osmocom SS7 xUA Server

A **xUA Server** is a server that binds + listens to a given SCTP (SIGTRAN) or TCP (IPA) port and accepts connections from remote peers (ASPs).

There can be any number of xUA Servers within one SS7 Instance, as long as they all run on a different combination of IP address and port.

Table 6: Major Attributes of an Osmocom SS7 xUA Server

Name	Description
Local IP	Local Port Number to which the server shall bind/listen
Local Port	Local IP Address to which the server shall bind/listen
Protocol	Protocol (M3UA, SUA, IPA) to be operated by this server
Accept Dynamic ASPs	Should we accept connections from ASPs that are not explicitly pre-configured with their source IP and port?

13.6 Osmocom SS7 Users

A SS7 User is part of a program that binds to a given MTP-Layer Service Indicator (SI). The Osmocom SS7 stack offers an API to register SS7 Users, as well as the VTY command `show cs7 instance <0-15> users` to list the currently registered users.

13.7 Osmocom SS7 Links

Conceptually, SS7 links are on the same level as SIGTRAN ASPs. The details of SS7 Links in the Osmocom implementation are TBD.

13.8 Osmocom SS7 Linksets

Conceptually, SS7 Linksets are on the same level as SIGTRAN ASs. The details of SS7 Links in the Osmocom implementation are TBD.

13.9 Osmocom SS7 Application Servers

This corresponds 1:1 to the SIGTRAN concept of an Application Server, i.e. a given external Application that interfaces the SS7 network via a SS7 protocol variant such as M3UA.

In the context of Osmocom, for each program connecting to a STP (like a BSC or MSC), you will have one Application Server definition.

An AS has the following properties:

Table 7: Major Attributes of an Osmocom SS7 Application Server

Name	Description
Name	A human-readable name for this instance
Description	More verbose description (for human user only)
Protocol	Protocol (M3UA, SUA, IPA) to be operated by this server
Routing Key	Routing Key (mostly Point Code) routed to this AS
Traffic Mode	Broadcast, Loadshare or Override
Recovery Timeout	Duration of the AS T(r) recovery timer. During this time, outgoing messages are queued. If the AS is ACTIVE before timer expiration, the queue is drained. At expiration, the queue is flushed.
State	Application Server State (Down, Inactive, Active, Pending)
ASPs	Which ASPs are permitted to transfer traffic for this AS

13.10 Osmocom SS7 Application Server Processes

An Application Server Process corresponds to a given SCTP (or TCP) connection. From the STP/SG (Server) point-of-view, those are incoming connections from Application Servers such as the BSCs. From the ASP (Client) Point of view, it has one `osmo_ss7_asp` object for each outbound SIGTRAN connection.

An ASP has the following properties:

Table 8: Major Attributes of an Osmocom SS7 Application Server Process

Name	Description
Name	A human-readable name for this instance
Description	More verbose description (for human user only)
Protocol	Protocol (M3UA, SUA, IPA) to be operated by this server
Role	Server (SG) or Client (ASP)?
Local Port	Port Number of the local end of the connection
Local IP	IP Address of the local end of the connection
Remote Port	Port Number of the remote end of the connection
Remote IP	IP Address of the remote end of the connection
State	ASP State (Down, Inactive, Active)

13.11 Osmocom SS7 Routes

An Osmocom SS7 Route holds information to route traffic with a matching destination point code and point code mask (similar to IP Address + Netmask) towards a specified SS7 Linkset or Application Server. The Linkset or Application Servers are identified by their name.

Routes are inserted in a routing table, where they are ordered first based on prefix mask length (longest prefix first) and then on priority (lowest value first). As a result, more specific routes (eg. a single Point Code) will match and be used before more generic ones (eg. a subnet).

Table 9: Major Attributes of an Osmocom SS7 Application Server Process

Name	Description
Point Code	Destination Point Code for this route
Mask	Destination Mask for this route (like an IP netmask)
Linkset/AS Name	Destination Linkset or AS, identified by name
Priority	Priority of the route in the routing table; lower value means higher priority

The `show cs7 instance <0-15> route` command can be used on the VTY to obtain the list of routes inserted in the system routing table.

The `show cs7 instance <0-15> route-lookup <DPC> from <OPC> sls <SLS>` command can be used on the VTY to test and debug routing of specific traffic on the currently running routing table, similar to what `ip route get` linux command does in the IP world.

13.12 Osmocom SCCP Instances

An Osmocom SCCP Instance can be bound to an Osmocom SS7 Instance. It will register/bind for the ITU-standard Service Indicator (SI).

13.13 Osmocom SCCP User

An Program (like a BSC) will *bind* itself to a given well-known sub-system number (SSN) in order to receive SCCP messages destined for this SSN.

There is an API to bind a program to a SSN, which implicitly generates an SCCP User object.

The `show cs7 instance <0-15> sccp users` command can be used on the VTY to obtain a list of currently bound SCCP users, as well as their corresponding SSNs.

13.14 Osmocom SCCP Connection

This is how Osmocom represents each individual connection of connection-oriented SCCP.

To illustrate the practical application: For the common use case of the A or Iu interfaces, this means that every dedicated radio channel that is currently active to any UE/MS has one SCCP connection to the MSC and/or SGSN.

The `show cs7 instance <0-15> sccp connections` command can be used on the VTY to obtain a list of currently active SCCP connections, as well as their source/destination and current state.

13.15 Osmocom SCCP User SAP

The Osmocom SCCP User SAP (Service Access Point) is the programming interface between the SCCP Provider (libosmo-sigtran) and the SCCP User (such as osmo-bsc, osmo-msc, osmo-hnbgw, etc.). It follows primitives as laid out in [itu-t-q711], encapsulated in `osmo_prim` structures.

13.16 Osmocom MTP User SAP

The Osmocom MTP User SAP (Service Access Point) is the programming interface between the MTP Provider and the MTP User (e.g. SCCP). It follows primitives as laid out in [itu-t-q711], encapsulated in `osmo_prim` structures.

14 Configure SCCP/M3UA

All CNI programs using SCCP/M3UA act as M3UA ASP role and SCTP client, expecting to connect to a Signalling Gateway (STP/SG) implementing the M3UA SG role as SCTP server. The STP/SG then routes M3UA messages between its ASPs, typically by point-codes.

For an introduction about SCCP/M3UA/SS7/SIGTRAN technology, please see the chapter *Signaling Networks: SS7 and SIGTRAN* in the OsmoSTP user manual.

In an all-Osmocom CNI, the typical simple/minimal usage is:

- OsmoSTP acts as the STP/SG (server role) and routes between the ASP,
- All other Osmocom CNI programs act as SCTP client and provide ASP implementations.

For example, in an all-Osmocom minimal setup,

- OsmoMSC contacts an OsmoSTP and subscribes its point-code 0.23.1;
- then OsmoBSC also contacts the same OsmoSTP, subscribes with its own point-code 1.23.3.
- Using these established links, OsmoBSC initiates an A-interface link by directing a BSSAP RESET message to the MSC's point-code 0.23.1,
- and the RESET ACK response from the MSC is routed back to the BSC's point-code 1.23.3.

The details of SCCP/M3UA are configured in the *cs7* section of the VTY configuration.

Osmocom programs automatically configure missing SCCP/M3UA configuration, by assuming sane defaults for small/minimal all-Osmocom installations, which may not be what you want in larger networks integrating with non-Osmocom core network elements.

If no explicit `routing-key` is set, it may be determined at runtime by negotiation with OsmoSTP — see OsmoSTP manual chapter "Osmocom M3UA Routing Key Management Extensions", regarding config option `accept-asp-connections dynamic-permitted`.

The complete active configuration of an Osmocom program can be obtained by the VTY command `show cs7 config` (the usual `show running-config` omits automatically configured items). Here is an example of OsmoMSC's default configuration:

```
OsmoMSC> show cs7 config
cs7 instance 0
 point-code 0.23.1
 asp asp-clnt-OsmoMSC-A-Iu 2905 0 m3ua
  remote-ip 127.0.0.1
  role asp
  sctp-role client
 as as-clnt-OsmoMSC-A-Iu m3ua
  asp asp-clnt-OsmoMSC-A-Iu
  routing-key 2 0.23.1
```

At the time of writing, SCCP/M3UA links involving Osmocom program are:

- A-interface: OsmoBSC to OsmoMSC

- IuCS-interface: OsmoHNBGW to OsmoMSC
- IuPS-interface: OsmoHNBGW to OsmoSGSN
- Lb-interface: OsmoSMLC to OsmoBSC

On the SCTP/IP level, those connections are actually all established from the respective program (BSC, MSC, HNBGW, SGSN, SMLC) to OsmoSTP. Hence, if you look at the traffic in a protocol analyzer like wireshark, at IP level, you will see each of those programs establishing an SCTP association from a random local IP to the well-known SCTP port for M3UA (2905) at the OsmoSTP.

Those star-connections for M3UA/SCTP then are the transport network for higher level protocols like SCCP. OsmoSTP then acts as central router for SCCP-level message exchange between all the connected programs.

14.1 Connect to STP Instance

Establishing an SCCP/M3UA link towards a remote STP instance can be configured as:

```
cs7 instance 0
 asp my-asp 2905 0 m3ua
  # IP address of the remote STP:
  remote-ip 10.23.24.1
  # optional: local bind to a specific IP
  local-ip 10.9.8.7
  role asp
  sctp-role client
```

Be aware that such an `asp` needs to be linked to an `as`, see Section 15.5.

By default, an STP instance is assumed to listen on the default M3UA port (2905) on the local host. That means in general `127.0.0.1` will be used as default remote SCTP address, and `::1` will be added to the SCTP association if IPv6 support is available on the system.

Note
OsmoSTP listens by default on `::` if IPv6 is enabled on the system, and on `0.0.0.0` otherwise. Address `::` actually superseeds `0.0.0.0`, meaning it will listen on all IPv4 and IPv6 addresses available on the system.

Caution
Some applications overwrite the default target remote address to be `localhost`. If IPv6 support is available on the system, `localhost` will usually resolve to `::1`, otherwise it will usually resolve to `127.0.0.1`.

14.2 Local Point-Code

Each CNI program on an SCCP/M3UA link typically has a local point-code, configurable by:

```
cs7 instance 0
 point-code 7.65.4
```

If an explicit routing context is configured, this point-code is repeated in the `routing-key` configuration:

```
cs7 instance 0
 point-code 0.23.1
 as my-as m3ua
  routing-key 2 0.23.1
```

See also Section 15.4.

14.3 Remote Point-Code

Programs establishing communication across SCCP links need a remote SCCP address, typically by point-code, to contact.

For example,

- OsmoBSC needs to know the MSC's point-code, to be able to establish the A-interface.
- OsmoHNBGW needs to know the MSC's point-code, to be able to establish the IuCS-interface.
- OsmoHNBGW needs to know the SGSN's point-code, to be able to establish the IuPS-interface.

To maintain remote SCCP addresses, each `cs7` instance maintains an SCCP address book:

```
cs7 instance 0
 sccp-address remote-pc-example
  point-code 1.23.1
```

This address book entry on its own has no effect. It is typically referenced by specific configuration items depending on the individual programs.

Examples:

- An OsmoBSC configures the MSC's remote SCCP address:

```
cs7 instance 0
 sccp-address my-remote-msc
  point-code 1.23.1
msc 0
 msc-addr my-remote-msc
```

- An HNBGW configures both the remote MSC's and SGSN's SCCP addresses:

```
cs7 instance 0
 sccp-address my-msc
  point-code 0.23.1
 sccp-address my-sgsn
  point-code 0.23.2
hnbgw
 iucs
  remote-addr my-msc
 iups
  remote-addr my-sgsn
```

Besides a point-code, an SCCP address can have several routing indicators:

- PC: routing by point-code is the default for Osmocom.
- GT: routing by Global Title is configurable by `routing-indicator GT`.
- IP: routing by IP address is configurable by `routing-indicator IP`.

In OsmoSTP, only routing by point-code is currently implemented.

14.4 Point-Code Format

Point-codes can be represented in various formats. For details, see OsmoSTP manual, chapter "Point Codes".

By default, Osmocom uses a point-code representation of 3.8.3, i.e. first digit of 3 bit, second digit of 8 bit, and third digit of 3 bit.

```
cs7 instance 0
 point-code format 3 8 3
 point-code 0.23.1
```

Often, point-codes are also represented as a single decimal number:

```
cs7 instance 0
 point-code format 24
 point-code 185
```

It is also possible to use a dash as delimiter.

```
cs7 instance 0
 point-code delimiter dash
 point-code 0-23-1
```

14.5 AS and ASP

Each CNI program needs at least one Application Server `as` and one Application Server Process `asp` configured on its `cs7` to be able to communicate on SCCP/M3UA. An `asp` needs to be part of at least one `as`. For details, see the OsmoSTP manual, chapters "Application Server" and "Application Server Process".

In Osmocom's `cs7`, any amount of `as` and `asp` can be configured by name, and an `as` references the `asp` entries belonging to it by their names.

In a simple/minimal Osmocom setup, an Osmocom CNI program would have exactly one `as` with one `asp`.

For example:

```
cs7 instance 0
 asp my-asp 2905 0 m3ua
  # where to reach the STP:
  remote-ip 127.0.0.1
  role asp
  sctp-role client
 as my-as m3ua
  asp my-asp
```

In Osmocom CNI programs, it is possible to omit the `as` and/or `asp` entries, which the program will then attempt to configure automatically.

When configuring both `as` and `asp` manually, make sure to link them by name. For example, the following configuration will **fail**, because `as` and `asp` are not linked:

```
cs7 instance 0
 asp my-asp 2905 0 m3ua
  remote-ip 127.0.0.1
  role asp
  sctp-role client
 as my-as m3ua
  routing-key 2 0.23.1
```

To **fix** above config, link the `asp` to an `as` by adding `asp my-asp`:

```
cs7 instance 0
 asp my-asp 2905 0 m3ua
  remote-ip 127.0.0.1
  role asp
  sctp-role client
 as my-as m3ua
  asp my-asp
  routing-key 2 0.23.1
```

14.6 Subsystem Number (SSN)

Osmocom CNI programs typically route SCCP/M3UA messages by PC+SSN: each ASP, having a given SCCP address, receives messages for one or more specific subsystems, identified by a Subsystem Number (SSN).

For example, the A-interface between BSC and MSC uses SSN = BSSAP (254). In Osmocom programs, SSNs do not need to be configured; they implicitly, naturally relate to the interfaces that a program implements.

For example, OsmoBSC takes the configured remote MSC's SCCP address and adds the SSN = BSSAP to it in order to contact the MSC's A-interface. To receive A-interface messages from the MSC, OsmoBSC subscribes a local user for this SSN on the ASP.

14.7 Routing Context / Routing Key

In SCCP/M3UA, messages can be routed by various Routing Indicators (PC+SSN, PC, GT, ...). Osmocom CNI programs typically use PC+SSN as Routing Indicator.

On the SG (for example OsmoSTP), each ASP's distinct Routing Indicator needs to be indexed by a distinct Routing Context (a simple index number scoped per SG), to forward M3UA to the correct peer.

The Osmocom SG implementation employs Routing Key Management (RKM, see OsmoSTP manual) to automatically determine a distinct Routing Context index for each connected ASP. Routing Contexts can also be configured manually — some non-Osmocom SG implementations require this.

Each Routing Context is associated with a Routing Indicator and address; this association is called a Routing Key.

For example, to configure an OsmoBSC with a local point-code of 1.23.3 to receive M3UA with Routing Context of 2 and RI=PC:

```
cs7 instance 0
 point-code 1.23.3
 as my-as m3ua
  routing-key 2 1.23.3
```

Osmocom programs so far implement Routing Keys by Destination Point Code (DPC), plus optional Subsystem Number (SSN) and/or Service Indicator (SI):

```
routing-key RCONTEXT DPC
routing-key RCONTEXT DPC si (aal2|bicc|b-isup|h248|isup|sat-isup|sccp|tup)
routing-key RCONTEXT DPC ssn SSN
routing-key RCONTEXT DPC si (aal2|bicc|b-isup|h248|isup|sat-isup|sccp|tup) ssn SSN
```

14.7.1 M3UA without Routing Context IE / Routing Context *0*

As per the M3UA specification, the use of the routing context IE is optional as long as there is only one AS within an ASP. As soon as there are multiple different AS within one ASP, the routing context IE is mandatory, as it is the only clue to differentiate which of the ASs a given message belongs to.

In the Osmocom M3UA implementation, it is generally assumed that a routing context IE is always used, for the sake of clarity.

However, the routing context ID of *0* has the special meaning of *do not encode a routing context IE on transmit*.

So if you configure an application like OsmoBSC to use routing context 0, then no routing context IE will be included in outbound M3UA messages.

This special interpretation of *0* within the Osmocom M3UA implementation however means that we can not represent M3UA with a routing context IE that actually contains *0* as a numeric identifier.

So you only have the following options: * Using M3UA with routing context (1..N) * Using M3UA without routing context (0)

14.7.2 Example: Static Routing

Osmocom SS7 supports dynamic routing key registration via M3UA Routing Key Management (RKM), allowing minimal SS7 configuration. If all of your components support dynamic RKM, you should probably use it: see `accept-asp-connections dynamic-permitted` in osmo-stp.cfg.

This chapter explains how to configure osmo-stp if dynamic RKM is not an option.

In this example, let's connect osmo-bsc via osmo-stp to osmo-msc using only static SS7 routing.

```
BSC        <--RK-1-->  STP         <--RK-3--> MSC
IP 1.1.1.1             IP 2.2.2.2             IP 3.3.3.3
M3UA 2905              M3UA 2905              M3UA 2905
PC 1.1.1                                      PC 3.3.3
```

Every one static route fanning out from STP gets assigned a distinct Routing Key — a simple integer number. Above, the BSC's link has RK 1, the MSC's link has RK 3.

For static routing, the M3UA port numbers must be fixed, i.e. there must be no `0` for a client's local port as in `asp foo 2905 0 m3ua`. Instead, you may use `asp foo 2905 2905 m3ua`.

The BSC needs to configure:

- its own point-code — has to match the PC configured for the BSC in osmo-stp.cfg
- the routing key — has to match the RK assigned to BSC's PC in osmo-stp.cfg
- the MSC's point-code — has to match the PC in osmo-stp.cfg and osmo-msc.cfg
- local and remote IP:ports for M3UA — have to match the IP:ports in osmo-stp.cfg

The MSC needs to configure:

- its own point-code — has to match the PC configured for the MSC in osmo-stp.cfg
- the routing key — has to match the RK assigned to MSC's PC in osmo-stp.cfg
- local and remote IP:ports for M3UA — have to match the IP:ports in osmo-stp.cfg

The STP needs to configure:

- all point-codes — they have to match the PCs in osmo-bsc.cfg and osmo-msc.cfg
- all routing keys — they have to match the RKs used in osmo-bsc.cfg and osmo-msc.cfg
- local and remote IP:ports for M3UA — have to match the IP:ports in osmo-bsc.cfg and osmo-msc.cfg

osmo-bsc.cfg

```
cs7 instance 0
 point-code 1.1.1

 asp mybsc-0 2905 2905 m3ua
  remote-ip 2.2.2.2
  local-ip 1.1.1.1
  sctp-role client
 as mybsc0 m3ua
  asp mybsc0-0
  routing-key 1 1.1.1

 sccp-address mymsc
  routing-indicator PC
  point-code 3.3.3

msc 0
 msc-addr mymsc
```

osmo-stp.cfg

```
cs7 instance 0
 xua rkm routing-key-allocation static-only
 listen m3ua 2905
  accept-asp-connections pre-configured
  local-ip 2.2.2.2

 # asp <name> <remote-port> <local-port|0> m3ua
 asp mybsc-0 2905 2905 m3ua
  remote-ip 1.1.1.1
  local-ip 2.2.2.2
 as mybsc m3ua
  asp bsc-0
  routing-key 1 1.1.1

 asp mymsc-0 2905 2905 m3ua
  remote-ip 3.3.3.3
  local-ip 2.2.2.2
 as mymsc m3ua
  asp mymsc-0
  routing-key 3 3.3.3

 route-table system
  update route 1.1.1 7.255.7 linkset mybsc
  update route 3.3.3 7.255.7 linkset mymsc
```

osmo-msc.cfg

```
cs7 instance 0
 point-code 3.3.3

 asp mymsc-0 2905 2905 m3ua
  remote-ip 2.2.2.2
  local-ip 3.3.3.3
  sctp-role client
 as mymsc0 m3ua
  asp mymsc0-0
  routing-key 3 3.3.3
```

For comparison, the same setup with dynamic routing key management is a lot shorter, especially at osmo-stp.cfg, and there is no need to manually configure point-codes and routing keys between STP and {BSC,MSC}:

osmo-bsc.cfg

```
cs7 instance 0
 point-code 1.1.1

 asp mybsc-0 2905 0 m3ua
  remote-ip 2.2.2.2
  local-ip 1.1.1.1
  sctp-role client
 as mybsc0 m3ua
  asp mybsc0-0

 sccp-address mymsc
  routing-indicator PC
  point-code 3.3.3

msc 0
 msc-addr mymsc
```

osmo-stp.cfg

```
cs7 instance 0
 xua rkm routing-key-allocation dynamic-permitted
 listen m3ua 2905
  accept-asp-connections dynamic-permitted
  local-ip 2.2.2.2
```

osmo-msc.cfg

```
cs7 instance 0
 point-code 3.3.3

 asp mymsc-0 2905 0 m3ua
  remote-ip 2.2.2.2
  local-ip 3.3.3.3
  sctp-role client
 as mymsc0 m3ua
  asp mymsc0-0
```

15 Configure SCCP/M3UA

All CNI programs using SCCP/M3UA act as M3UA ASP role and SCTP client, expecting to connect to a Signalling Gateway (STP/SG) implementing the M3UA SG role as SCTP server. The STP/SG then routes M3UA messages between its ASPs, typically by point-codes.

For an introduction about SCCP/M3UA/SS7/SIGTRAN technology, please see the chapter *Signaling Networks: SS7 and SIGTRAN* in the OsmoSTP user manual.

In an all-Osmocom CNI, the typical simple/minimal usage is:

- OsmoSTP acts as the STP/SG (server role) and routes between the ASP,
- All other Osmocom CNI programs act as SCTP client and provide ASP implementations.

For example, in an all-Osmocom minimal setup,

- OsmoMSC contacts an OsmoSTP and subscribes its point-code 0.23.1;
- then OsmoBSC also contacts the same OsmoSTP, subscribes with its own point-code 1.23.3.

- Using these established links, OsmoBSC initiates an A-interface link by directing a BSSAP RESET message to the MSC's point-code 0.23.1,
- and the RESET ACK response from the MSC is routed back to the BSC's point-code 1.23.3.

The details of SCCP/M3UA are configured in the *cs7* section of the VTY configuration.

Osmocom programs automatically configure missing SCCP/M3UA configuration, by assuming sane defaults for small/minimal all-Osmocom installations, which may not be what you want in larger networks integrating with non-Osmocom core network elements.

If no explicit `routing-key` is set, it may be determined at runtime by negotiation with OsmoSTP — see OsmoSTP manual chapter "Osmocom M3UA Routing Key Management Extensions", regarding config option `accept-asp-connections dynamic-permitted`.

The complete active configuration of an Osmocom program can be obtained by the VTY command `show cs7 config` (the usual `show running-config` omits automatically configured items). Here is an example of OsmoMSC's default configuration:

```
OsmoMSC> show cs7 config
cs7 instance 0
 point-code 0.23.1
 asp asp-clnt-OsmoMSC-A-Iu 2905 0 m3ua
  remote-ip 127.0.0.1
  role asp
  sctp-role client
 as as-clnt-OsmoMSC-A-Iu m3ua
  asp asp-clnt-OsmoMSC-A-Iu
  routing-key 2 0.23.1
```

At the time of writing, SCCP/M3UA links involving Osmocom program are:

- A-interface: OsmoBSC to OsmoMSC
- IuCS-interface: OsmoHNBGW to OsmoMSC
- IuPS-interface: OsmoHNBGW to OsmoSGSN
- Lb-interface: OsmoSMLC to OsmoBSC

On the SCTP/IP level, those connections are actually all established from the respective program (BSC, MSC, HNBGW, SGSN, SMLC) to OsmoSTP. Hence, if you look at the traffic in a protocol analyzer like wireshark, at IP level, you will see each of those programs establishing an SCTP association from a random local IP to the well-known SCTP port for M3UA (2905) at the OsmoSTP.

Those star-connections for M3UA/SCTP then are the transport network for higher level protocols like SCCP. OsmoSTP then acts as central router for SCCP-level message exchange between all the connected programs.

15.1 Connect to STP Instance

Establishing an SCCP/M3UA link towards a remote STP instance can be configured as:

```
cs7 instance 0
 asp my-asp 2905 0 m3ua
  # IP address of the remote STP:
  remote-ip 10.23.24.1
  # optional: local bind to a specific IP
  local-ip 10.9.8.7
  role asp
  sctp-role client
```

Be aware that such an `asp` needs to be linked to an `as`, see Section 15.5.

By default, an STP instance is assumed to listen on the default M3UA port (2905) on the local host. That means in general `127.0.0.1` will be used as default remote SCTP address, and `::1` will be added to the SCTP association if IPv6 support is available on the system.

Note
OsmoSTP listens by default on `::` if IPv6 is enabled on the system, and on `0.0.0.0` otherwise. Address `::` actually superseeds `0.0.0.0`, meaning it will listen on all IPv4 and IPv6 addresses available on the system.

Caution
Some applications overwrite the default target remote address to be `localhost`. If IPv6 support is available on the system, `localhost` will usually resolve to `::1`, otherwise it will usually resolve to `127.0.0.1`.

15.2 Local Point-Code

Each CNI program on an SCCP/M3UA link typically has a local point-code, configurable by:

```
cs7 instance 0
 point-code 7.65.4
```

If an explicit routing context is configured, this point-code is repeated in the `routing-key` configuration:

```
cs7 instance 0
 point-code 0.23.1
 as my-as m3ua
  routing-key 2 0.23.1
```

See also Section 15.4.

15.3 Remote Point-Code

Programs establishing communication across SCCP links need a remote SCCP address, typically by point-code, to contact.

For example,

- OsmoBSC needs to know the MSC's point-code, to be able to establish the A-interface.
- OsmoHNBGW needs to know the MSC's point-code, to be able to establish the IuCS-interface.
- OsmoHNBGW needs to know the SGSN's point-code, to be able to establish the IuPS-interface.

To maintain remote SCCP addresses, each `cs7` instance maintains an SCCP address book:

```
cs7 instance 0
 sccp-address remote-pc-example
  point-code 1.23.1
```

This address book entry on its own has no effect. It is typically referenced by specific configuration items depending on the individual programs.

Examples:

- An OsmoBSC configures the MSC's remote SCCP address:

```
cs7 instance 0
 sccp-address my-remote-msc
  point-code 1.23.1
msc 0
 msc-addr my-remote-msc
```

- An HNBGW configures both the remote MSC's and SGSN's SCCP addresses:

```
cs7 instance 0
 sccp-address my-msc
  point-code 0.23.1
 sccp-address my-sgsn
  point-code 0.23.2
hnbgw
 iucs
  remote-addr my-msc
 iups
  remote-addr my-sgsn
```

Besides a point-code, an SCCP address can have several routing indicators:

- PC: routing by point-code is the default for Osmocom.
- GT: routing by Global Title is configurable by `routing-indicator GT`.
- IP: routing by IP address is configurable by `routing-indicator IP`.

In OsmoSTP, only routing by point-code is currently implemented.

15.4 Point-Code Format

Point-codes can be represented in various formats. For details, see OsmoSTP manual, chapter "Point Codes".

By default, Osmocom uses a point-code representation of 3.8.3, i.e. first digit of 3 bit, second digit of 8 bit, and third digit of 3 bit.

```
cs7 instance 0
 point-code format 3 8 3
 point-code 0.23.1
```

Often, point-codes are also represented as a single decimal number:

```
cs7 instance 0
 point-code format 24
 point-code 185
```

It is also possible to use a dash as delimiter.

```
cs7 instance 0
 point-code delimiter dash
 point-code 0-23-1
```

15.5 AS and ASP

Each CNI program needs at least one Application Server `as` and one Application Server Process `asp` configured on its `cs7` to be able to communicate on SCCP/M3UA. An `asp` needs to be part of at least one `as`. For details, see the OsmoSTP manual, chapters "Application Server" and "Application Server Process".

In Osmocom's `cs7`, any amount of `as` and `asp` can be configured by name, and an `as` references the `asp` entries belonging to it by their names.

In a simple/minimal Osmocom setup, an Osmocom CNI program would have exactly one `as` with one `asp`.

For example:

```
cs7 instance 0
 asp my-asp 2905 0 m3ua
  # where to reach the STP:
  remote-ip 127.0.0.1
  role asp
  sctp-role client
 as my-as m3ua
  asp my-asp
```

In Osmocom CNI programs, it is possible to omit the `as` and/or `asp` entries, which the program will then attempt to configure automatically.

When configuring both `as` and `asp` manually, make sure to link them by name. For example, the following configuration will **fail**, because `as` and `asp` are not linked:

```
cs7 instance 0
 asp my-asp 2905 0 m3ua
  remote-ip 127.0.0.1
  role asp
  sctp-role client
 as my-as m3ua
  routing-key 2 0.23.1
```

To **fix** above config, link the `asp` to an `as` by adding `asp my-asp`:

```
cs7 instance 0
 asp my-asp 2905 0 m3ua
  remote-ip 127.0.0.1
  role asp
  sctp-role client
 as my-as m3ua
  asp my-asp
  routing-key 2 0.23.1
```

15.6 Subsystem Number (SSN)

Osmocom CNI programs typically route SCCP/M3UA messages by PC+SSN: each ASP, having a given SCCP address, receives messages for one or more specific subsystems, identified by a Subsystem Number (SSN).

For example, the A-interface between BSC and MSC uses SSN = BSSAP (254). In Osmocom programs, SSNs do not need to be configured; they implicitly, naturally relate to the interfaces that a program implements.

For example, OsmoBSC takes the configured remote MSC's SCCP address and adds the SSN = BSSAP to it in order to contact the MSC's A-interface. To receive A-interface messages from the MSC, OsmoBSC subscribes a local user for this SSN on the ASP.

15.7 Routing Context / Routing Key

In SCCP/M3UA, messages can be routed by various Routing Indicators (PC+SSN, PC, GT, ...). Osmocom CNI programs typically use PC+SSN as Routing Indicator.

On the SG (for example OsmoSTP), each ASP's distinct Routing Indicator needs to be indexed by a distinct Routing Context (a simple index number scoped per SG), to forward M3UA to the correct peer.

The Osmocom SG implementation employs Routing Key Management (RKM, see OsmoSTP manual) to automatically determine a distinct Routing Context index for each connected ASP. Routing Contexts can also be configured manually — some non-Osmocom SG implementations require this.

Each Routing Context is associated with a Routing Indicator and address; this association is called a Routing Key.

For example, to configure an OsmoBSC with a local point-code of 1.23.3 to receive M3UA with Routing Context of 2 and RI=PC:

```
cs7 instance 0
 point-code 1.23.3
 as my-as m3ua
  routing-key 2 1.23.3
```

Osmocom programs so far implement Routing Keys by Destination Point Code (DPC), plus optional Subsystem Number (SSN) and/or Service Indicator (SI):

```
routing-key RCONTEXT DPC
routing-key RCONTEXT DPC si (aal2|bicc|b-isup|h248|isup|sat-isup|sccp|tup)
routing-key RCONTEXT DPC ssn SSN
routing-key RCONTEXT DPC si (aal2|bicc|b-isup|h248|isup|sat-isup|sccp|tup) ssn SSN
```

15.7.1 M3UA without Routing Context IE / Routing Context *0*

As per the M3UA specification, the use of the routing context IE is optional as long as there is only one AS within an ASP. As soon as there are multiple different AS within one ASP, the routing context IE is mandatory, as it is the only clue to differentiate which of the ASs a given message belongs to.

In the Osmocom M3UA implementation, it is generally assumed that a routing context IE is always used, for the sake of clarity.

However, the routing context ID of *0* has the special meaning of *do not encode a routing context IE on transmit*.

So if you configure an application like OsmoBSC to use routing context 0, then no routing context IE will be included in outbound M3UA messages.

This special interpretation of *0* within the Osmocom M3UA implementation however means that we can not represent M3UA with a routing context IE that actually contains *0* as a numeric identifier.

So you only have the following options: * Using M3UA with routing context (1..N) * Using M3UA without routing context (0)

15.7.2 Example: Static Routing

Osmocom SS7 supports dynamic routing key registration via M3UA Routing Key Management (RKM), allowing minimal SS7 configuration. If all of your components support dynamic RKM, you should probably use it: see `accept-asp-connections dynamic-permitted` in osmo-stp.cfg.

This chapter explains how to configure osmo-stp if dynamic RKM is not an option.

In this example, let's connect osmo-bsc via osmo-stp to osmo-msc using only static SS7 routing.

```
 BSC        <--RK-1-->  STP         <--RK-3--> MSC
 IP 1.1.1.1             IP 2.2.2.2             IP 3.3.3.3
 M3UA 2905              M3UA 2905              M3UA 2905
 PC 1.1.1                                      PC 3.3.3
```

Every one static route fanning out from STP gets assigned a distinct Routing Key — a simple integer number. Above, the BSC's link has RK 1, the MSC's link has RK 3.

For static routing, the M3UA port numbers must be fixed, i.e. there must be no `0` for a client's local port as in `asp foo 2905 0 m3ua`. Instead, you may use `asp foo 2905 2905 m3ua`.

The BSC needs to configure:

- its own point-code — has to match the PC configured for the BSC in osmo-stp.cfg
- the routing key — has to match the RK assigned to BSC's PC in osmo-stp.cfg
- the MSC's point-code — has to match the PC in osmo-stp.cfg and osmo-msc.cfg
- local and remote IP:ports for M3UA — have to match the IP:ports in osmo-stp.cfg

The MSC needs to configure:

- its own point-code — has to match the PC configured for the MSC in osmo-stp.cfg
- the routing key — has to match the RK assigned to MSC's PC in osmo-stp.cfg
- local and remote IP:ports for M3UA — have to match the IP:ports in osmo-stp.cfg

The STP needs to configure:

- all point-codes — they have to match the PCs in osmo-bsc.cfg and osmo-msc.cfg
- all routing keys — they have to match the RKs used in osmo-bsc.cfg and osmo-msc.cfg
- local and remote IP:ports for M3UA — have to match the IP:ports in osmo-bsc.cfg and osmo-msc.cfg

osmo-bsc.cfg

```
cs7 instance 0
 point-code 1.1.1

 asp mybsc-0 2905 2905 m3ua
  remote-ip 2.2.2.2
  local-ip 1.1.1.1
  sctp-role client
 as mybsc0 m3ua
  asp mybsc0-0
  routing-key 1 1.1.1

 sccp-address mymsc
  routing-indicator PC
  point-code 3.3.3

msc 0
 msc-addr mymsc
```

osmo-stp.cfg

```
cs7 instance 0
 xua rkm routing-key-allocation static-only
 listen m3ua 2905
  accept-asp-connections pre-configured
  local-ip 2.2.2.2

 # asp <name> <remote-port> <local-port|0> m3ua
 asp mybsc-0 2905 2905 m3ua
  remote-ip 1.1.1.1
```

```
  local-ip 2.2.2.2
 as mybsc m3ua
  asp bsc-0
  routing-key 1 1.1.1

 asp mymsc-0 2905 2905 m3ua
  remote-ip 3.3.3.3
  local-ip 2.2.2.2
 as mymsc m3ua
  asp mymsc-0
  routing-key 3 3.3.3

 route-table system
  update route 1.1.1 7.255.7 linkset mybsc
  update route 3.3.3 7.255.7 linkset mymsc
```

osmo-msc.cfg

```
cs7 instance 0
 point-code 3.3.3

 asp mymsc-0 2905 2905 m3ua
  remote-ip 2.2.2.2
  local-ip 3.3.3.3
  sctp-role client
 as mymsc0 m3ua
  asp mymsc0-0
  routing-key 3 3.3.3
```

For comparison, the same setup with dynamic routing key management is a lot shorter, especially at osmo-stp.cfg, and there is no need to manually configure point-codes and routing keys between STP and {BSC,MSC}:

osmo-bsc.cfg

```
cs7 instance 0
 point-code 1.1.1

 asp mybsc-0 2905 0 m3ua
  remote-ip 2.2.2.2
  local-ip 1.1.1.1
  sctp-role client
 as mybsc0 m3ua
  asp mybsc0-0

 sccp-address mymsc
  routing-indicator PC
  point-code 3.3.3

msc 0
 msc-addr mymsc
```

osmo-stp.cfg

```
cs7 instance 0
 xua rkm routing-key-allocation dynamic-permitted
 listen m3ua 2905
  accept-asp-connections dynamic-permitted
  local-ip 2.2.2.2
```

osmo-msc.cfg

```
cs7 instance 0
 point-code 3.3.3

 asp mymsc-0 2905 0 m3ua
  remote-ip 2.2.2.2
  local-ip 3.3.3.3
  sctp-role client
 as mymsc0 m3ua
  asp mymsc0-0
```

16 Configuring the Core Network

The core network parameters are configured by the config file (as in `osmo-msc -c osmo-msc.cfg`). The config file is parsed by the VTY, which is also available via telnet in the running `osmo-msc` instance. Be aware that even though you may be able to change these parameters without restarting `osmo-msc`, some may not take immediate effect, and it is safest to use the config file to have these parameters set at startup time.

The core network parameters are found in the `config` / `network`.

A full reference to the available commands can be found in the *OsmoMSC VTY reference manual* [vty-ref-osmomsc]. This section describes only the most commonly used settings.

Here is an overview of the config items, described in more detail below:

```
network
 network country code 262
 mobile network code 89
 mm info 1
 short name OsmoMSC
 long name OsmoMSC
 authentication required
 encryption a5 3
```

Tip
Use the telnet VTY interface to query the current configuration of a running `osmo-msc` process:

```
$ telnet localhost 4254
OsmoMSC> enable
OsmoMSC# show running-config
```

Some parameters may be changed without restarting `osmo-msc`. To reach the `network` node, enter:

```
OsmoMSC> enable
OsmoMSC# configure terminal
OsmoMSC(config)# network
OsmoMSC(config-net)# short name Example-Name
OsmoMSC(config-net)# exit
OsmoMSC(config)#
```

The telnet VTY features tab-completion as well as context sensitive help shown when entering a `?` question mark.
You can always use the `list` VTY command or enter `?` on the blank prompt to get a list of all possible commands at the current node.

16.1 MCC/MNC

The key identities of every GSM PLMN is the Mobile Country Code and the Mobile Network Code. They are identical over the entire network. In most cases, the MCC/MNC will be allocated to the operator by the respective local regulatory authority. For example, to set the MCC/MNC of 262-89, have this in your osmo-msc.cfg:

```
network
 network country code 262
 mobile network code 89
```

16.2 Configuring MM INFO

The *MM INFO* procedure can be used after a successful *LOCATION UPDATE* in order to transmit the human-readable network name as well as local time zone information to the MS. By default, *MM INFO* is not active, i.e. `0`. Set to `1` to activate this feature:

```
network
 mm info 1
 short name OsmoMSC
 long name OsmoMSC
```

Note
Not all phones support the MM INFO procedure. If a phone is not factory-programmed to contain the name for your MCC/MNC, it will likely only provide a numeric display of the network name, such as *262-89*, or show the country code transformed into a letter, such as *D 89*.

The time information transmitted is determined by the local system time of the operating system on which OsmoMSC is running.

16.3 Authentication

A subscriber's IMSI must be entered in the HLR database to be able to attach. A subscriber-create-on-demand feature is also available, see the *OsmoHLR reference manual* [userman-osmohlr].

A known IMSI in the HLR may or may not have authentication keys associated, which profoundly affects the ability to attach and the algorithms used to negotiate authentication, as the following sections explain for 2G and 3G.

16.3.1 Authentication on 2G

If authentication tokens (such as KI for 2G, or K and OP/OPC for UMTS) are present in the HLR, OsmoMSC will only attach a subscriber after successful authentication. Note that the 3G authentication keys are also used on 2G when the MS indicates UMTS AKA capability, in which case the full UMTS style mutual authentication may indeed take place on 2G (GERAN).

On 2G, if no authentication keys are present in the HLR for a given subscriber, OsmoMSC will attach the subscriber *without* authentication. Subscribers that lack authentication keys can always be rejected with this setting:

```
network
 authentication required
```

16.3.2 Authentication on 3G

3G (UTRAN) always requires authentication (a.k.a. Integrity Protection) by specification, and hence authentication keys must be present in the HLR for a subscriber to be able to attach on 3G.

OsmoMSC always indicates UIA1 and UIA2 as permitted Integrity Protection algorithms on 3G.

16.4 Ciphering

To enable ciphering on the radio link, authentication must take place first: the Kc resulting from authentication is the key used for ciphering. Hence, to be able to use ciphering, a subscriber must have authentication tokens available in the HLR.

16.4.1 Ciphering on 2G

The MS, BTS and MSC must agree on a ciphering algorithm to use.

- The MS sends its supported ciphering algorithms via Classmark IEs during Location Updating.
- Typically the BSC needs to know which A5 ciphers are supported by connected BTSes, see the `network / encryption a5` configuration item for OsmoBSC [vty-ref-osmobsc].
- Finally, OsmoMSC may impose that specific A5 ciphers shall not be considered.

It is the responsibility of the BSC to then pick an A5 cipher that satisfies all requirements.

- In OsmoMSC, A5/0 means that ciphering is turned off.

```
network
 encryption a5 0
```

- A5/1 and A5/3 are currently supported by Osmocom.

```
network
 encryption a5 1 3
```

- Never use A5/2: it is an "export grade cipher" and has been deprecated for its low ciphering strength.
- To allow either no encryption or any of A5/1 or A5/3 based on the presence of authentication keys and abilities of the MS, SIM and BSC configuration, it is recommended to enable all ciphers in OsmoMSC. The highest available A5 cipher will be used; the order in which the A5 options are configured does not affect the choice.

```
network
 encryption a5 0 1 3
```

16.4.2 Ciphering on 3G

While authentication is always required on 3G, ciphering is optional.

So far OsmoMSC allows switching ciphering on 3G either on or off — the default behavior is to enable ciphering. (Individual choice of algorithms may be added in the future.)

Disable 3G ciphering:

```
network
 encryption uea 0
```

Enable 3G ciphering (default):

```
network
 encryption uea 1 2
```

OsmoMSC indicates UEA1 and UEA2 as permitted encryption algorithms on 3G.

17 Short Message Peer to Peer (SMPP)

The *Short Message Peer to Peer (SMPP) Protocol* [smpp-34] has been used for the communication with SMSCs. Osmocom implements version 3.4 of the protocol. Using this interface one can send MT-SMS to an attached subscriber or receive unrouted MO-SMS.

SMPP is served by the Osmocom MSC layer (both in the old OsmoNITB as well as the new OsmoMSC.

SMPP describes a situation where multiple ESMEs (External SMS Entities) interact with a SMSC (SMS Service Center) via the SMPP protocol. Each entity is identified by its System Id. The System ID is a character string which is configured by the system administrator.

OsmoMSC implements the SMSC side of SMPP and subsequently acts as a TCP server accepting incoming connections from ESME client programs.

Each ESME identifies itself to the SMSC with its system-id and an optional shared password.

17.1 Global SMPP configuration

There is a `smpp` vty node at the top level of the OsmoMSC configuration. Under this node, the global SMPP configuration is performed.

Use the `local-tcp-ip` command to define the TCP IP and port at which the OsmoMSC internal SMSC should listen for incoming SMPP connections. The default behaviour is to listen on all IPs (0.0.0.0), and the default port assigned to SMPP is 2775.

Use the `system-id` command to define the System ID of the SMSC.

Use the `policy` parameter to define whether only explicitly configured ESMEs are permitted to access the SMSC (`closed`), or whether any ESME should be accepted (`accept-all`).

Use the `smpp-first` command to define if SMPP routes have higher precedence than MSISDNs contained in the HLR (`smpp-first`), or if only MSISDNs found not in the HLR should be considered for routing to SMPP (`no smpp-first`).

17.2 ESME configuration

Under the `smpp` vty node, you can add any number of `esme` nodes, one for each ESME that you wish to configure.

Use the `esme NAME` command (where NAME corresponds to the system-id of the ESME to be configured) under the SMPP vty node to enter the configuration node for this given ESME.

Use the `password` command to specify the password (if any) for the ESME.

Use the `default-route` command to indicate that any MO-SMS without a more specific route should be routed to this ESME.

Use the `deliver-src-imsi` command to indicate that the SMPP DELIVER messages for MO SMS and the SMPP ALERT should state the IMSI (rather than the MSISDN) as source address.

Use the `osmocom-extensions` command to request that Osmocom specific extension TLVs shall be included in the SMPP PDUs. Those extensions include the ARFCN of the cell, the L1 transmit power of the MS, the timing advance, the uplink and dwnlink RxLev and RxQual, as well as the IMEI of the terminal at the time of generating the SMPP DELIVER PDU.

Use the `dcs-transparent` command to transparently pass the DCS value from the SMS Layer3 protocols to SMPP, instead of converting them to the SMPP-specific values.

Use the `route prefix` command to specify a route towards this ESME. Using routes, you specify which destination MSISDNs should be routed towards your ESME.

17.3 Example configuration snippet

The following example configuration snippet shows a single ESME *galactica* with a prefix-route of all national numbers stating with 2342:

```
smpp
 local-tcp-port 2775
 policy closed
 no smpp-first
 esme galactica
  password SoSayWeAll
  deliver-src-imsi
  osmocom-extensions
   route prefix national isdn 2342
```

17.4 Osmocom SMPP protocol extensions

Osmocom has implemented some extensions to the SMPP v3.4 protocol.

These extensions can be enabled using the `osmocom-extensions` VTY command at `esme` level.

The TLV definitions can be found in the `<osmocom/gsm/protocol/smpp34_osmocom.h>` header file provided by libosmocore.

17.4.1 RF channel measuremets

When the Osmocom SMPP extensions are enabled, we add the following TLVs to each SMPP DELIVER PDU:

TLV	IEI	Length (Octets)	Purpose
TLVID_osmo_arfcn	0x2300	2	GSM ARFCN of the radio interface
TLVID_osmo_ta	0x2301	1	Timing Advance on the radio interface
TLVID_osmo_ms_l1_txpwr	0x2307	1	Transmit Power of the MS in uplink direction
TLVID_osmo_rxlev_ul	0x2302	2	Uplink receive level as measured by BTS in dBm (int16_t)
TLVID_osmo_rxqual_ul	0x2303	1	Uplink RxQual value as measured by BTS
TLVID_osmo_rxlev_dl	0x2304	2	Downlink receive level as measured by MS in dBm (int16_t)
TLVID_osmo_rxqual_dl	0x2305	1	Downlink RxQual value as measured by MS

All of the above values reflect the **last measurement report** as received vi A-bis RSL from the BTS. It is thus a snapshot value (of the average within one 480ms SACCH period), and not an average over all the SACCH periods during which the channel was open or the SMS was received. Not all measurement reports contain all the values. So you might not get an TLVID_osmo_rxlev_dl IE, as that particular uplink frame might habe benn lost for the given snapshot we report.

17.4.2 Equipment IMEI

If we know the IMEI of the subscribers phone, we add the following TLV to each SMPP DELIVER PDU:

TLV	IEI	Length	Purpose
TLVID_osmo_imei	0x2306	variable	IMEI of the subscribers phone (ME)

18 MNCC for External Call Control

The 3GPP GSM specifications define an interface point (service access point) inside the MSC between the call-control part and the rest of the system. This service access point is called the MNCC-SAP. It is described in *3GPP TS 24.007* [3gpp-ts-24-007] Chapter 7.1.

However, like for all internal interfaces, 3GPP does not give any specific encoding for the primitives passed at this SAP.

The MNCC protocol has been created by the Osmocom community and allows to control the call handling and audio processing by an external application. The interface is currently exposed using Unix Domain Sockets. The protocol is defined in the `mncc.h` header file.

It is exposed by the Osmocom MSC layer (both in the old OsmoNITB as well as the new OsmoMSC.

OsmoMSC can run in two different modes:

1. with internal MNCC handler
2. with external MNCC handler

18.1 Internal MNCC handler

When the internal MNCC handler is enabled, OsmoMSC will switch voice calls between GSM subscribers internally and automatically based on the the subscribers *extension* number. No external software is required.

Note
Internal MNCC is the default behavior.

18.1.1 Internal MNCC Configuration

The internal MNCC handler offers some configuration parameters under the `mncc-int` VTY configuration node.

18.1.1.1 `default-codec tch-f (fr|efr|amr)`

Using this command, you can configure the default voice codec to be used by voice calls on TCH/F channels.

18.1.1.2 `default-codec tch-h (hr|amr)`

Using this command, you can configure the default voice codec to be used by voice calls on TCH/H channels.

18.2 External MNCC handler

When the external MNCC handler is enabled, OsmoMSC will not perform any internal call switching, but delegate all call-control handling towards the external MNCC program connected via the MNCC socket.

If you intend to operate OsmoMSC with external MNCC handler, you have to disable the internal MNCC handler and specify the MNCC socket path in the configuration file.

At the time of this writing, there are only two known open source applications implementing the MNCC interface compatible with the Osmocom MNCC socket:

- historically `lcr`, the Linux Call Router (support for modern MNCC protocol versions may be missing)
- `osmo-sip-connector`, the more up-to-date integration of external call routing by translating MNCC into a SIP trunk towards an external SIP PBX / switch.

18.3 DTMF considerations

In mobile networks, the signaling of DTMF tones is implemented differently, depending on the signaling direction. A mobile originated DTMF tone is signaled using START/STOP DTMF messages which are hauled through various protocols upwards into the core network.

Contrary to that, a mobile terminated DTMF tone is not transferred as an out of band message. Instead, in-band signaling is used, which means a tone is injected early inside a PBX or MGW.

When using OsmoMSC with its built in MNCC functionality a mobile originated DTMF message will not be translated into an in-band tone. Therefore, sending DTMF will not work when internal MNCC is used.

For external MNCC, the network integrator must make sure that the back-end components are configured properly in order to handle the two different signaling schemes depending on the signaling direction.

Note
osmo-sip-connector will translate MNCC DTMF signaling into sip-info messages. DTMF signaling in the opposite direction is not possible. osmo-sip-connector will reject sip-info messages that attempt to signal a DTMF tone.

18.4 MNCC protocol description

The protocol follows the primitives specified in 3GPP TS 04.07 Chapter 7.1. The encoding of the primitives is provided in the `mncc.h` header file in OsmoMSC's source tree, which uses some common definitions from `osmocom/gsm/mncc.h` (part of libosmocore.git).

However, Osmocom's MNCC specifies a number of additional primitives beyond those listed in the 3GPP specification.

The different calls in the network are distinguished by their callref (call reference), which is a unique unsigned 32bit integer.

18.4.1 MNCC_HOLD_IND

Direction: OsmoMSC → Handler

A *CC HOLD* message was received from the MS.

18.4.2 MNCC_HOLD_CNF

Direction: Handler → OsmoMSC

Acknowledge a previously-received *CC HOLD* message, causes the transmission of a *CC HOLD ACK* message to the MS.

18.4.3 MNCC_HOLD_REJ

Direction: Handler → OsmoMSC

Reject a previously-received *CC HOLD* message, causes the transmission of a *CC HOLD REJ* message to the MS.

18.4.4 MNCC_RETRIEVE_IND

Direction: OsmoMSC → Handler

A *CC RETRIEVE* message was received from the MS.

18.4.5 MNCC_RETRIEVE_CNF

Direction: Handler → OsmoMSC

Acknowledge a previously-received *CC RETRIEVE* message, causes the transmission of a *CC RETRIEVE ACK* message to the MS.

18.4.6 MNCC_RETRIEVE_REJ

Direction: Handler → OsmoMSC

Reject a previously-received *CC RETRIEVE* message, causes the transmission of a *CC RETRIEVE REJ* message to the MS.

18.4.7 MNCC_USERINFO_REQ

Direction: OsmoMSC → Handler

Causes a *CC USER INFO* message to be sent to the MS.

18.4.8 MNCC_USERINFO_IND

Direction: OsmoMSC → Handler

Indicates that a *CC USER-USER* message has been received from the MS.

18.4.9 MNCC_BRIDGE

Direction: Handler → OsmoMSC

Requests that the TCH (voice) channels of two calls shall be inter-connected. This is the old-fashioned way of using MNCC, historically required for circuit-switched BTSs whose TRAU frames are received via an E1 interface card, and works only when the TCH channel types match.

Note
Internal MNCC uses MNCC_BRIDGE to connect calls directly between connected BTSs or RNCs, in effect disallowing calls between mismatching TCH types and forcing all BTSs to be configured with exactly one TCH type and codec. This is a limitation that will probably remain for the old OsmoNITB. For the new OsmoMSC, the MNCC_BRIDGE command will instruct the separate OsmoMGW to bridge calls, which will be able to handle transcoding between different TCH as well as 3G (IuUP) payloads (but note: not yet implemented at the time of writing this). Hence an external MNCC may decide to bridge calls directly between BTSs or RNCs that both are internal to the OsmoMSC, for optimization reasons.

18.4.10 MNCC_FRAME_RECV

Direction: Handler → OsmoMSC

Enable the forwarding of TCH voice frames via the MNCC interface in OsmoMSC→Handler direction for the specified call.

18.4.11 MNCC_FRAME_DROP

Direction: Handler → OsmoMSC

Disable the forwarding of TCH voice frames via the MNCC interface in OsmoMSC→Handler direction for the specified call.

18.4.12 MNCC_LCHAN_MODIFY

Direction: Handler → OsmoMSC

Modify the current dedicated radio channel from signalling to voice, or if it is a signalling-only channel (SDCCH), assign a TCH to the MS.

18.4.13 MNCC_RTP_CREATE

Direction: Handler → OsmoMSC

Create a RTP socket for this call at the BTS/TRAU that serves this BTS.

18.4.14 MNCC_RTP_CONNECT

Direction: Handler → OsmoMSC

Connect the RTP socket of this call to the given remote IP address and port.

18.4.15 MNCC_RTP_FREE

Direction: Handler → OsmoMSC

Release a RTP connection for one given call.

18.4.16 GSM_TCHF_FRAME

Direction: both

Transfer the payload of a GSM Full-Rate (FR) voice frame between the OsmoMSC and an external MNCC handler.

18.4.17 GSM_TCHF_FRAME_EFR

Direction: both

Transfer the payload of a GSM Enhanced Full-Rate (EFR) voice frame between the OsmoMSC and an external MNCC handler.

18.4.18 GSM_TCHH_FRAME

Direction: both

Transfer the payload of a GSM Half-Rate (HR) voice frame between the OsmoMSC and an external MNCC handler.

18.4.19 GSM_TCH_FRAE_AMR

Direction: both

Transfer the payload of a GSM Adaptive-Multi-Rate (AMR) voice frame between the OsmoMSC and an external MNCC handler.

18.4.20 GSM_BAD_FRAME

Direction: OsmoMSC → Handler

Indicate that no valid voice frame, but a *bad frame* was received over the radio link from the MS.

18.4.21 MNCC_START_DTMF_IND

Direction: OsmoMSC → Handler

Indicate the beginning of a DTMF tone playback.

18.4.22 MNCC_START_DTMF_RSP

Direction: Handler → OsmoMSC

Acknowledge that the DTMF tone playback has been started.

18.4.23 MNCC_START_DTMF_REJ

Direction: both

Indicate that starting a DTMF tone playback was not possible.

18.4.24 MNCC_STOP_DTMF_IND

Direction: OsmoMSC → Handler

Indicate the ending of a DTMF tone playback.

18.4.25 MNCC_STOP_DTMF_RSP

Direction: Handler → OsmoMSC

Acknowledge that the DTMF tone playback has been stopped.

19 Osmux

`Osmux` is a protocol aimed at multiplexing and transmitting voice and signalling traffic from multiple sources in order to reduce the overall bandwidth consumption. This feature becomes specially meaningful in case of satellite based GSM systems, where the transmission cost on the back-haul is relatively expensive. In such environment, even seemingly small protocol optimizations, eg. message batching and trunking, can result in significant cost reduction.

Full reference document for the osmux protocol can be found here: https://ftp.osmocom.org/docs/latest/osmux-reference.pdf

In Osmocom satellite based GSM networks, the satellite link is envisioned to be in between the BSS and the core network, that is, between the BSC and the MSC (or BSC-NAT). Hence, Osmocom components can make use of `Osmux` protocol to multiplex payload audio streams from call legs between OsmoBSC and OsmoMSC (or OsmoBSCNAT). The MGW attached those components need of course also be aware of `Osmux` existence in order to properly set up the audio data plane.

Under some specific circumstances, the operator may decide to set up the network with a bandwidth-limited (e.g. satellite) link between BTS and BSC. Hence, use of the `Osmux` protocol is also possible between an Osmux capable BTS (like OsmoBTS) and OsmoBSC (and its co-located MGW).

19.1 Osmux and NAT

It is quite usual for satellite based links to use NATs, which means any or both of the two components at each side of the satellite link (BSC and MSC/BSC-NAT) may end up being behind a NAT and being unable to provide the real public address to its peer on the other side of the satellite.

As a result, upon call parameter negotiation (RTP/Osmux IP address and port), those parameters won't be entirely useful and some specific logic needs to be introduced into the network components to circumvent the NAT under those cases.

For instance, if the BSC and its co-located MGW (BSC/MGW from now on) is under a NAT, it may end up providing its private address and port as RTP/Osmux parameters to the MSC/MGW through GSM protocols, but MSC will fail to send any message to that tuple because of the NAT or routing issues (due to IP address being a private address). In that scenario, MSC/MGW needs to be aware that there's a NAT and wait until an RTP/Osmux message arrives from the BSC/MGW host. It then can, from that message source IP address and port (and CID in case of Osmux), discover the real public IP address and port of the peer

(BSC/MGW). From that point on, the BSC/MGW punched a hole in the NAT (its connection table is updated) and MSC/MGW is able to send data back to it on that connection.

In order to make use of the features above, OsmoMGW must be made aware explicitly through VTY configuration that its peers are located behind a NAT. This is done through the `osmux peer-behind-nat (on|off)` VTY commands.

If OsmoMGW itself is behind a NAT, it must use the VTY config `rtp keep-alive` (used for both RTP and Osmux) to at least the value `once`, in order for it to punch the hole in its NAT so that its peer can know where to send packets back to it.

Another characteristic of NATs is that they tend to drop connections from their connection tables after some inactivity time, meaning a peer may receive notice about the other end not being available while it actually is. This means the GSM network needs to be configured in a way to ensure inactivity periods are short enough that this cannot occur.

Hence, if OsmoMGW is behind a NAT, it is actually desirable to have the VTY config `rtp keep-alive` configured with the `<1-120>` value in order to force transmission of dummy packets ever few seconds.

Osmux implementations such as OsmoMGW also come with the `osmux dummy` VTY command to enable sending dummy AMR payloads to the peer even if no real data was received (for instance if DTX is used). This is useful under some specific satellite links which were proven to work unreliably if the total throughput in use over the link changes over time. This way throughput resources are kept pre-allocated until they are needed again (audio is received again).

19.2 CID allocation

Each peer (BSC/MGW and MSC/MGW) allocates its own *local CID*, and receives from its peer a *remote CID* (aka the peer's *local CID*) through the used GSM protocol. This *remote CID* is then used to send Osmux frames to that peer.

```
BSC/MGW(localCID=Y,remoteCID=?)<-X--MSC/MGW(localCID=X,remoteCID=?)
BSC/MGW(localCID=Y,remoteCID=X)--Y->MSC/MGW(localCID=X,remoteCID=Y)
```

This way each peer is responsible for allocating and managing their own local address (CID) space. This is basically the same that happens with regular IP address and port in the RTP case (and those also apply when Osmux is used, but an extra identifier, the CID, is allocated).

In an ideal scenario, without NAT, each BSC/MGW would have a public, differentiated and unique IP and port set tuple, And MSC/MGW should be able to identify messages coming from them by easily matching source IP address, port (and CID in Osmux case) against the parameters negotiated during call set up.

In this kind of scenario, MSC/MGW could easily open and manage one Osmux socket per BSC (based on SDP IPaddr and port parameters), with same `<localIPaddr, localPort>` tuple, allowing for 256 Osmux CIDs per BSC and hence call legs per BSC. Each of the peers could actually have more than one Osmux socket towards the other peer, by using a pool of ports or IP addresses, so there's really not limit if required as long as there's a way to infer the initially negotiated `<srcIP, srcPort, dstIP, dstPort, remoteCID>` tuple from the received audio packets.

However, due to some constrains from in between NATs explained in section above, BSC/MGW IP address and port are not a priory known, and could change between different connections coming from it. As a result, it is difficult to infer the entire tuple, so for now MGW needs to allocate its Osmux *local CID* in a clever way, in order to be able to identify the full tuple from it.

Hence, currently OsmoMGW CID allocation implementation shares CID between all connections, which means it can only handle up to 256 concurrent Osmux connections (call legs).

Future work in OsmoMGW (OS#4092) plans to use a set of local ports for Osmux sockets instead of only 1 currently used. This way local ports can be matched against specific `<remoteIP, remotePort>` tuples and have an entire 256 Osmux CID space per `<remoteIP, remotePort>` (aka per peer).

19.3 3GPP AoIP network setup with Osmux

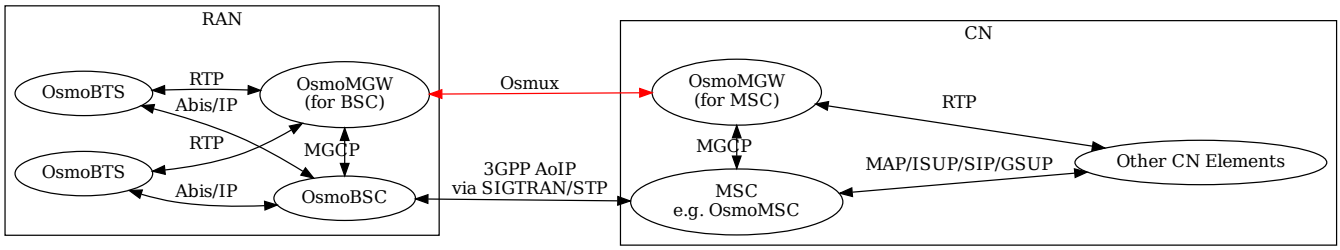


Figure 5: Sample node diagram of a 3GPP AoIP network with Osmux enabled

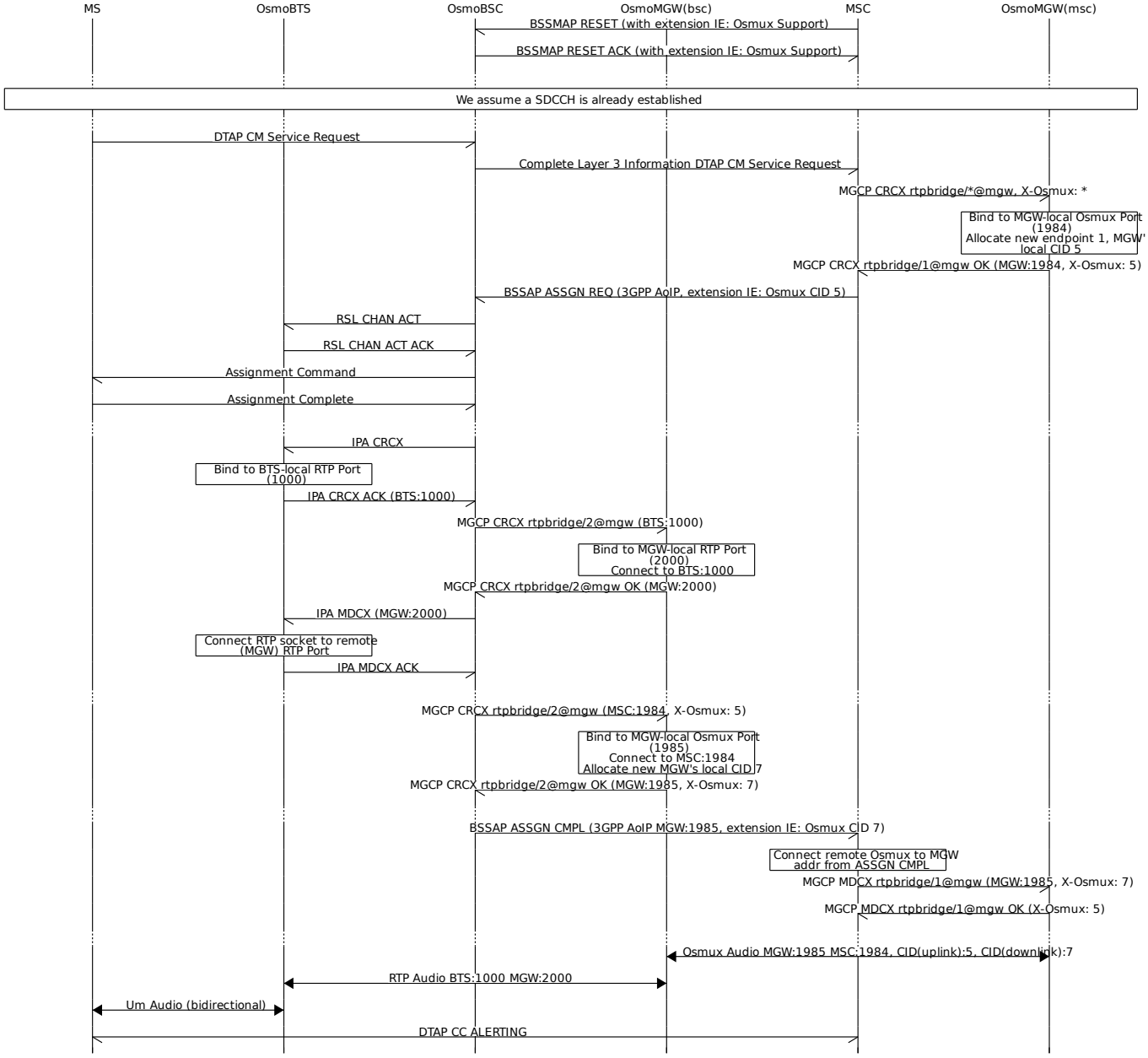


Figure 6: MO-call with Osmux enable on 3GPP AoIP

19.4 SCCPLite network setup with Osmux

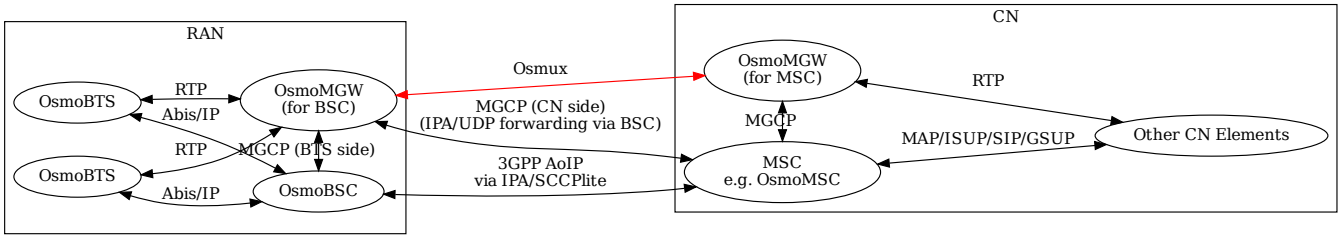


Figure 7: Sample node diagram of a 3GPP AoIP using A/IP with IPA/SCCPlite network with Osmux enabled

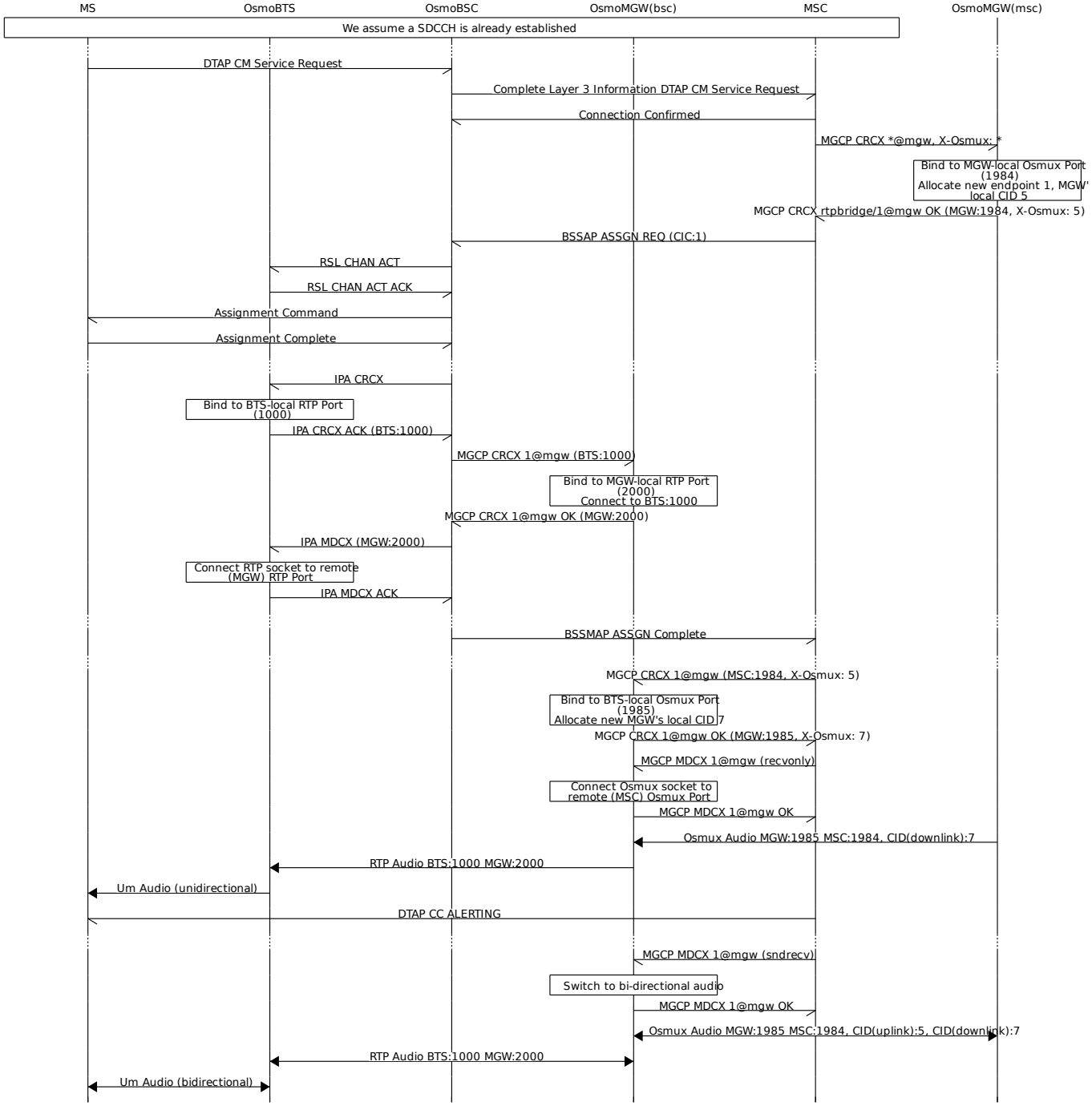


Figure 8: MO-call with Osmux enable on 3GPP AoIP using A/IP with IPA/SCCPlite

19.5 SCCPLite network setup with Osmux + BSC-NAT

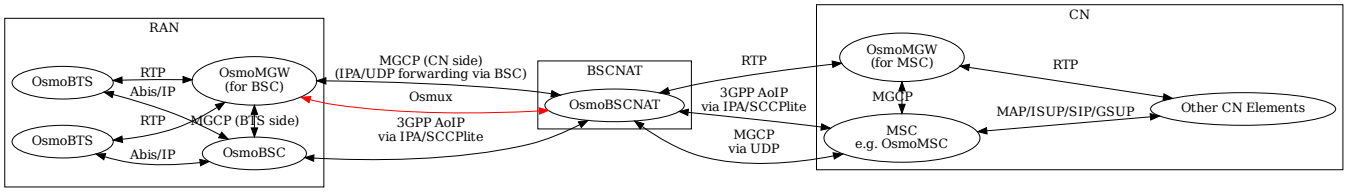


Figure 9: Sample node diagram of a 3GPP AoIP using A/IP with IPA/SCCPlite network and BSC-NAT with Osmux enabled

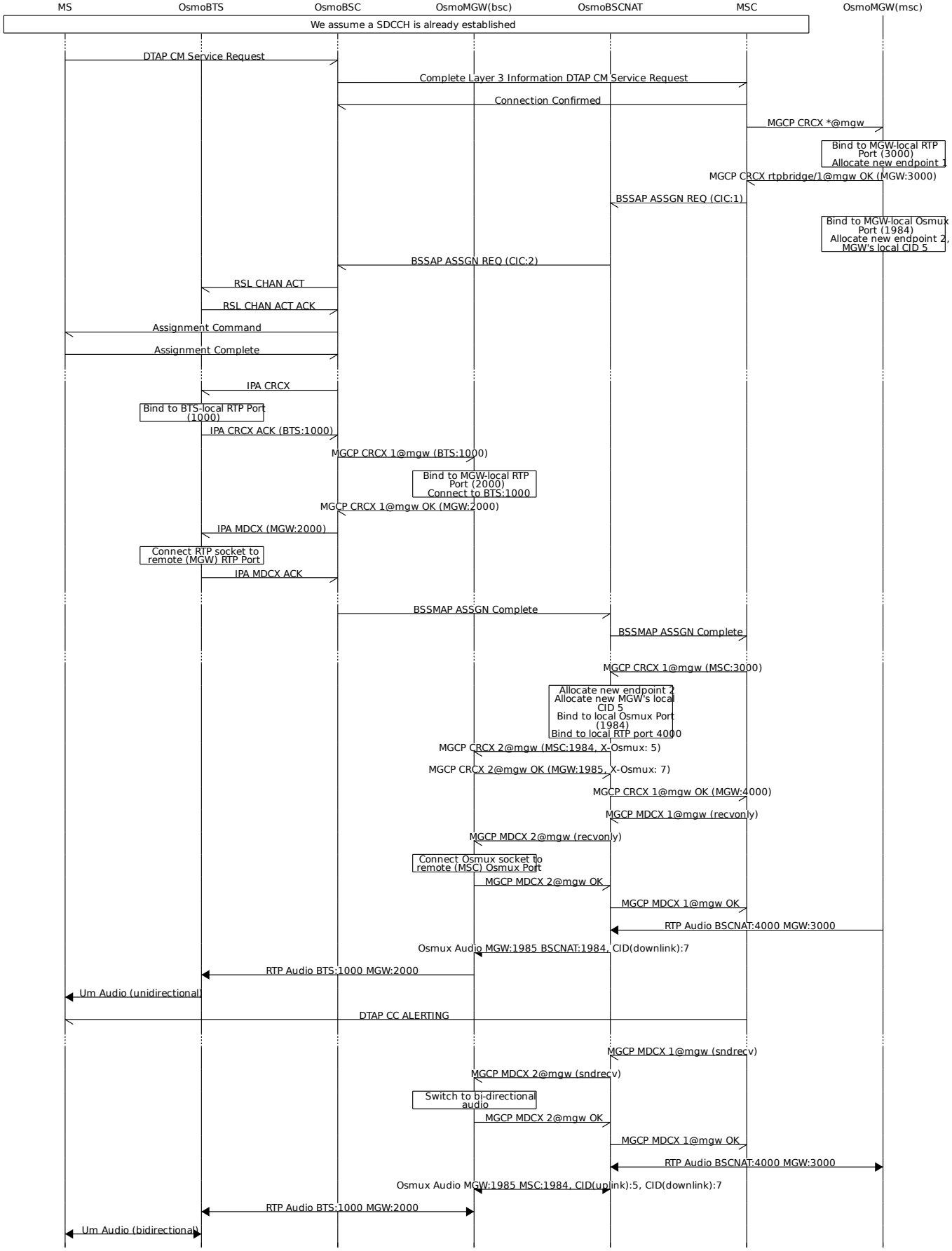


Figure 10: MO-call with Osmux enable on 3GPP AoIP using A/IP with IPA/SCCPlite with a BSC-NAT between BSC and MSC

19.6 Osmux and MGCP

`X-Osmux` indicates to OsmoMGW that a given connection of an `rtpbridge` endpoint has to be configured in order to handle Osmux frames instead of RTP messages on the data plane.

19.6.1 `X-Osmux` Format

The value part of `X-Osmux` must be one integer in range [0..255], or alternatively only on request messages, an asterisk (*) if the value is not yet known.

`X-Osmux` must be issued in the MGCP header section (typically as its last item), before the SDP section starts.

`X-Osmux` can be included inside `CRCX` and `MDCX` request messages, as well as their respective response messages.

In request messages, the value part of `X-Osmux` specifies the CID to be used by OsmoMGW to *send* Osmux frames to the remote peer for that connection, also known as the MGW's *remote CID* or the peer's *local CID*.

In response messages, the value part of `X-Osmux` specifies the CID where OsmoMGW expect to *receive* Osmux frames from the remote peer for that connection, also known as the MGW's *local CID* or the peer's *remote CID*.

Example: X-Osmux format with a known CID 3.

```
X-Osmux: 3
```

Example: X-Osmux format with a wildcard (not yet known) CID.

```
X-Osmux: *
```

19.6.2 `X-Osmux` Considerations

If the MGCP client is willing to use Osmux for a given connection, it shall specify so during `CRCX` time, and not later. If at `CRCX` time the MGCP client doesn't yet know the MGW's *remote CID*, it can use an astersik (*) and provide *remote CID* later within `MDCX` messages.

All subsequent `MDCX` messages sent towards an Osmux connection must contain the original MGW's *remote CID* sent during `CRCX`. The same way, all `MDCX` response shall contain the *local CID* sent during `CRCX`.

The other required connection address parameters, such as IP address, port, and codecs, are negotiated through MGCP and SDP as usual, but in this case the IP address and port specific the Osmux socket IP address and port to use, that together with the Osmux CID conform the entire tuple identifying a Osmux stream.

Since Osmux only supports AMR codec payloads, the SDP must specify use of AMR codec.

Example: CRCX message that instructs OsmoMGW to create an Osmux connection

```
CRCX 189 rtpbridge/1@mgw MGCP 1.0
C: 36
M: sendrecv
X-Osmux: 2

v=0
o=- 36 23 IN IP4 172.18.2.20
s=-
c=IN IP4 1.2.3.4
t=0 0
m=audio 2342 RTP/AVP 112
a=fmtp:112
a=rtpmap:112 AMR/8000/1
a=ptime:20
```

Example: response to CRCX containing the MGW's local CID

```
200 189 OK
I: 07E41584
X-Osmux: 2
Z: rtpbridge/1@mgw

v=0
o=- foo 21 IN IP4 172.18.1.20
s=-
c=IN IP4 172.18.1.20
t=0 0
m=audio 11002 RTP/AVP 112
a=rtpmap:112 AMR/8000
a=ptime:20
```

19.6.3 `X-Osmux` Support

`X-Osmux` is known to be supported by OsmoMGW on the MGCP server side, and by OsmoBSC as well as OsmoMSC on the MGCP client side (through libosmo-mgcp-cli). No other programs supporting this feature are known or envisioned at the time of writing this document.

In OmoMGW, Osmux support is managed through VTY.

Example: Sample config file section with Osmux configuration

```
mgcp
 ...
 osmux on ❶
 osmux bind-ip 172.18.1.20 ❷
 osmux port 1984 ❸
 osmux batch-factor 4 ❹
 osmux dummy on ❺
```

❶ Allow clients to set allocate Osmux connections in `rtpbridge` endpoints, while still allowing RTP connections

❷ Bind the Osmux socket to the provided IP address

❸ Bind the Osmux socket to the provided UDP port

❹ Batch up to 4 RTP payloads of the same stream on each Osmux frame

❺ Periodically send Osmux dummy frames, useful to punch a hole in NATs and maintain connections opened.

19.7 Abis setup with Osmux

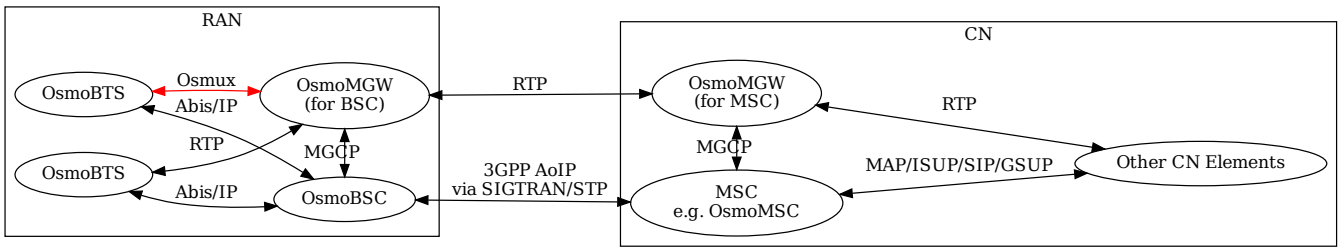


Figure 11: Sample node diagram of Osmux enabled in the Abis interface

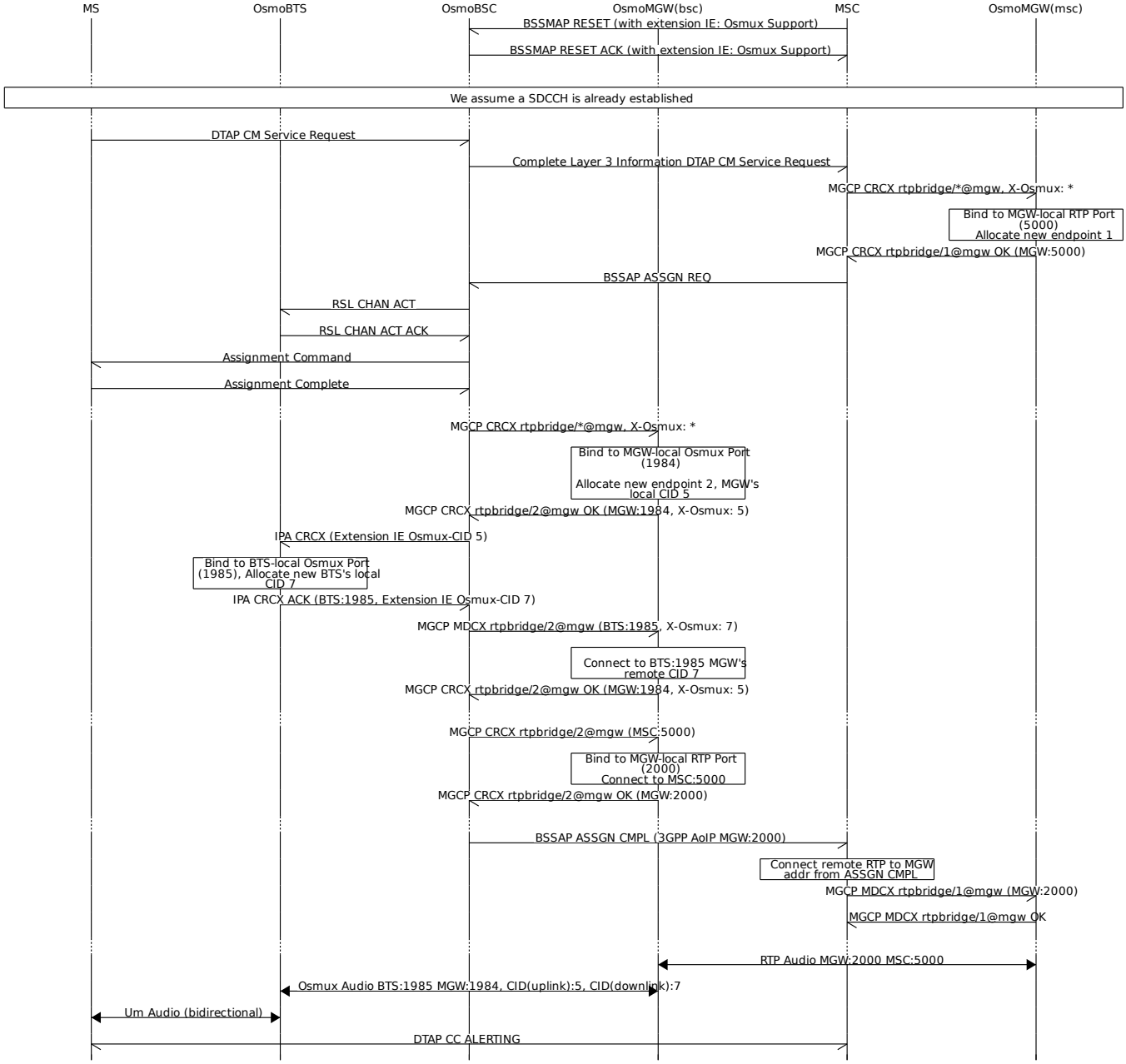


Figure 12: MO-call with Osmux enabled on Abis

19.8 Osmux Support in OsmoMSC

19.8.1 OsmoMSC in a A/IP with IPA/SCCPlite network setup

In this kind of setup, the CN side of BSC co-located MGW is managed by the MSC, meaning the use of Osmux is transparent to BSC since MSC takes care of both peer MGW connections. Moreover, in this case the MSC has no dynamic information on Osmux support in the BSC co-located MGW until `CRCX` time, which means configuration on both nodes need to be carefully set up so they can work together.

Osmux usage in OsmoMSC in managed through the VTY command `osmux (on|off|only)`. Since there's no dynamic information on Osmux support, it may be required in the future to have an extra VTY command which can be set per BSC to fine-tune which ones should use Osmux and which shouldn't.

OsmoMSC will behave differently during call set up based on the VTY command presented above:

- `off`: OsmoMSC won't include an `X-Osmux` extension to `CRCX` sent to the BSC co-located MGW when configuring the CN side of the MGW endpoint. If the MGW answers with a `CRCX ACK` containing an `X-Osmux`, OsmoMSC will cancel the call establishment.
- `on`: OsmoMSC will initially configure its co-located MGW to use Osmux, then similarly send a `CRCX` with an `X-Osmux` extension towards the BSC co-located MGW. Under this configuration, if the BSC co-located MGW didn't support Osmux, it could send a `CRCX ACK` without `X-Osmux` extension or fail (depending on its own configuration), and OsmoMSC could choose to re-create its local connection as non-Osmux (RTP) (and possibly try again against BSC co-located MGW), but this behavior is currently not implemented. As a result, currently `on` behaves the same as `only`.
- `only`: OsmoMSC will configure its co-located MGW as well as the BSC co-located MGW to use Osmux by including the `X-Osmux` MGCP extension. If MGW rejects to use Osmux, OsmoMSC will reject the call and the call establishment will fail.

19.8.2 OsmoMSC in a 3GPP AoIP network setup

Osmux usage in OsmoMSC in managed through the VTY command `osmux (on|off|only)`. Once enabled (`on` or `only`), OsmoMSC will start appending the vendor specific *Osmux Support* IE in *BSSMAP RESET* and *BSSMAP RESET-ACK* message towards the BSC in order to announce it supports Osmux, and BSC will do the same. This way, OsmoMSC can decide whether to use Osmux or not based on this information when setting up a call (this time using *Osmux CID* IE). It should be noted that this option should not be enabled unless BSCs managed by OsmoMSC support handling this extension IE (like OsmoBSC), 3rd-party BSCs might otherwise refuse the related *RESET/RESET-ACK* messages.

OsmoMSC will behave differently during call set up based on the VTY command presented above:

- `off`: OsmoMSC won't use Osmux. That is, it will send a *BSSMAP Assign Request* without the *Osmux CID* IE, and will send a `CRCX` without `X-Osmux` extension towards its co-located MGW.
- `on`: If BSC announced Osmux support to OsmoMSC during *BSSMAP RESET* time, then OsmoMSC will set up the call to use Osmux (by adding `X-Osmux` to MGCP `CRCX` and *Osmux CID* IE to *BSSMAP Assign Request*). If the BSC didn't announce Osmux support to OsmoMSC, then OsmoMSC will use RTP to set up the call (by avoiding addition of previously described bits).
- `only`: Same as per `on`, except that OsmoMSC will allow to set up only Osmux calls on the CN-side, this is, it will reject to set up voice calls for BSC which didn't announce Osmux support.

20 MGW Pooling

OsmoMSC is able to use a pool of media gateway (MGW) instances. The aim of MGW pooling is to evenly distribute the RTP voice stream load across multiple MGW instances. This can help to scale out over multiple VMs or physical machines. Until osmo-mgw includes multithreading support, it may also be used to scale-out to multiple cores on a single host.

The load distribution is managed in such a way that when a new call is placed, the pool will automatically assign the call to the available MGW with the lowest load.

MGW pooling is recommended for larger RAN or CN installations. For small networks and lab installations the classic method with one MGW per OsmoMSC offers sufficient performance.

20.1 MGW pool VTY configuration

In OsmoMSC the MGW is controlled via an MGCP-Client. The VTY commands to configure the MGCP-Client are part of the several *mgw* nodes, one per MGCP-Client to set up.

To setup an MGW pool, the user must first install multiple OsmoMGW instances, so that they won't interfere with each other. This can be done using different local host IP addresses or different ports. When OsmoMGW is installed from packages, the systemd configuration may also require adjustment.

By default, MGCP-Client will use whatever source IP address is resolved by the kernel routing subsystem, and will also ask the kernel to pick a free UDP port.

Example configuration with two MGCP-Client instances in a pool:

```
mgw 0
 description media-gw-0 ❶
 remote-ip 127.0.0.1
 remote-port 2432
 local-ip 127.0.0.1
 local-port 2431
 endpoint-domain mgw0 ❷
mgw 1
 description media-gw-1 ❸
 remote-ip 127.0.0.1
 remote-port 2430
 local-ip 127.0.0.1
 local-port 2429
 endpoint-domain mgw1 ❹
```

❷, ❹ When working with multiple MGW / MGCP-Client instances, the domain name for each MGW should be different. Otherwise it won't be possible to distinguish the endpoint names in the log. It should also be noted that the domain name must match the configuration of the related OsmoMGW instance.

❶, ❸ It is also possible to assign a descriptive name to each MGW instance. The MGCP client specific log lines will then use this name as logging context. If no description is set, the domain name will be used.

20.2 MGW pool management

The MGW pool is fully runtime-manageable. The only limitation is that an MGCP-Client can not be restarted or removed as long as it is serving calls (see also: Section 20.2.5).

20.2.1 MGW pool status

The VTY implements a *show mgw-pool* command that lists the currently configured MGW pool members, their status and call utilization. The following snippet shows an output example run on OsmoBSC, but it should be also available on other applications supporting the MGW pooling VTY features:

```
OsmoBSC> show mgw-pool
% MGW-Pool:
%  MGW 0:media-gw-0
%   MGCP link:     connected,UP
%   service:       unblocked
%   ongoing calls: 1
%  MGW 1:media-gw-1
%   MGCP link:     connected,UP
%   service:       unblocked
%   ongoing calls: 0
```

20.2.2 Adding an MGW / MGCP-Client to the MGW pool

To add a new MGCP-Client to the pool, the *mgw* node is used. Like with the *bts* or the *msc* node a reference number is used that usually starts at 0. However it is still possible to assign any number from 0-255. The enumeration also may contain gaps. The following snippet shows an output example run on OsmoBSC, but it should be also available on other applications supporting the MGW pooling VTY features:

```
OsmoBSC> enable
OsmoBSC# configure terminal
OsmoBSC(config)# network
OsmoBSC(config-net)# mgw 2
OsmoBSC(config-mgw)# ?
  keepalive        Monitor if the MGCP link against MGW is still usable
  local-ip         local bind to connect to MGW from
  local-port       local port to connect to MGW from
  remote-ip        remote IP address to reach the MGW at
  remote-port      remote port to reach the MGW at
  endpoint-domain  Set the domain name to send in MGCP messages, e.g. the part 'foo' in ' ↩
      rtpbridge/*@foo'.
  reset-endpoint   Add an endpoint name that should be reset (DLCX) on connect to the reset ↩
      -endpoint list,e.g. 'rtpbridge/*'
```

The newly added MGW will immediately appear in the mgw-pool list but it won't be used until its configuration finished by reconnecting it.

```
% MGW-Pool:
%  MGW 0:media-gw-0
%   MGCP link:     connected,UP
%   service:       unblocked
%   ongoing calls: 2
%  MGW 1:media-gw-1
%   MGCP link:     connected,UP
%   service:       unblocked
%   ongoing calls: 3
%  MGW 2:mgw ❶
%   MGCP link:     disconnected,DOWN
%   service:       unblocked
%   ongoing calls: 0
```

❶ In this example a description is not set yet, so the domain name ("mgw") is displayed.

20.2.3 Reconnecting an MGW / MGCP-Client

It may become necessary to reconnect an MGCP-Client. This is the case when the VTY configuration was changed at runtime. In order to make the changes effective the MGW configuration must be reloaded by reconnecting the MGW connection. Also newly created MGW instances require a reconnect once their configuration is done.

To reconnect an MGCP-Client use the *reconnect* VTY command:

```
OsmoBSC# mgw 2 reconnect
```

The mgcp-client status should immediately change to *connected*. The MGW is now ready to be used for new calls.

```
OsmoBSC# show mgw-pool
% MGW-Pool:
%  MGW 0:media-gw-0
%   MGCP link:     connected,UP
%   service:       unblocked
%   ongoing calls: 2
%  MGW 1:media-gw-1
%   MGCP link:     connected,UP
%   service:       unblocked
%   ongoing calls: 3
%  MGW 2:mgw
%   MGCP link:     connected,UP
%   service:       unblocked
%   ongoing calls: 0
```

It should be noted that MGCP a protocol is used via UDP, the connect only happens locally to forward the UDP datagrams properly (state printed in `mgcp-client: (dis)connected` above). Also (unless a reset endpoint is configured like in the example config above) there will be no immediate interaction with the MGW. However, the log should at least confirm the connect worked and the MGCP client has been created successfully.

```
Mon Aug  2 17:15:00 2021 DLMGCP mgcp_client.c:788 MGCP client: using endpoint domain '@mgw'
Mon Aug  2 17:15:00 2021 DLMGCP mgcp_client.c:908 MGCP GW connection: r=127.0.0.1:2427<->l ↩
    =127.0.0.1:2727
```

For that reason, it is strongly advised to enable the `keepalive` feature in OsmoMSC to schedule periodical MGCP queries against the MGW, in order to make sure it is reachable (the state `MGCP link: UP|DOWN` printed above). See section Section 20.2.4 below for more information.

20.2.4 Monitor MGCP link (keepalive) / MGCP-Client

The `keepalive` feature in OsmoMSC allows scheduling periodical queries on the MGCP layer in order to make sure it is reachable and hence obtain information on the state of the MGCP link. This is in turn used by the MGW Pool when picking an MGW from the pool: MGWs whose link is considered to be DOWN are skipped.

The feature consists of: - A `keepalive request-interval` which will trigger a transmission of an MGCP AuditEndpoint command targeting endpoint with name `keepalive request-endpoint`. This interval is updated every time any message is transmitted in the MGCP link, meaning the MGCP AuditEndpoint message is only triggered if no message has been transmitted since `keepalive request-interval` seconds ago. - A `keepalive timeout` which upon triggering (because no message was received over that amount of time) will then consider the MGW to be non-reachable (link DOWN).

The `keepalive` parameters are to be preferrably configured so that `"keepalive request-interval" * 2 < "keepalive timeout"`.

Example VTY configuration of `keepalive` feature in OsmoMSC:

```
mgw 0
 ...
 keepalive request-interval 20 ❶
 keepalive request-endpoint null ❷
 keepalive timeout 50 ❸
```

❶ Transmit an MGCP AuditEndpoint message to the MGW if no message has been sent to it over last 20 seconds

❷ The MGCP AuditEndpoint targets the `null` endpoint. This is a special endpoint available at OsmoMGW for those purposes, but any available endpoint can be configured and used instead.

❸ Consider the MGCP link to be DOWN if no message is received from the MGW over the last 50 seconds

Note
The `keepalive` feature is disabled by default, and must be explicitly configured in order to enable it.

20.2.5 Blocking an MGW / MGCP-Client

If it becomes apparent that an MGCP-Client must be restarted or removed from the config (maintenance) the operator can put that MGCP-Client into a blocked mode. A blocked MGCP-Client will still serve the ongoing calls but it will not be picked for the assignment of new calls.

To block an MGCP-Client use the *block* VTY command:

```
OsmoBSC# mgw 2 block
OsmoBSC# show mgw-pool
% MGW-Pool:
%  MGW 0:media-gw-0
```

```
%   MGCP link:     connected,UP
%   service:       unblocked
%   ongoing calls: 11
%  MGW 1:media-gw-1
%   MGCP link:     connected,UP
%   service:       unblocked
%   ongoing calls: 12
%  MGW 2:mgw
%   MGCP link:     connected,UP
%   service:       blocked
%   ongoing calls: 10
```

When the number of ongoing calls has tapered off, the MGW / MGCP-Client can be restarted or removed if necessary.

```
OsmoBSC# show mgw-pool
% MGW-Pool:
%  MGW 0:media-gw-0
%   MGCP link:     connected,UP
%   service:       unblocked
%   ongoing calls: 15
%  MGW 1:media-gw-1
%   MGCP link:     connected,UP
%   service:       unblocked
%   ongoing calls: 14
%  MGW 2:mgw
%   MGCP link:     connected,UP
%   service:       blocked
%   ongoing calls: 0
```

If the blockade should be reverted, the *unblock* VTY command can be used in the same way to remove the blockade. (Reconnecting will not remove the blockade.)

20.2.6 Removing an MGW / MGCP-Client

An MGCP-Client is removed from the pool using the *no mgw* command from the configure terminal. The MGCP-Client instance will automatically be terminated and the related resources are freed. The only requirement is that there are no ongoing calls on the selected instance.

```
OsmoBSC# configure terminal
OsmoBSC(config)# network
OsmoBSC(config-net)# no mgw 2
```

21 SGs interface

OsmoMSC offers an SGs interface using the SGsAP protocol. The SGs interface is an optional interface between a 2G (GERAN) / 3G (UTRAN) MSC and an 4G (EUTRAN) MME. Its purpose is to facilitate both CSFB (Circuit-Switched Fall Back) and SMSoS (SMS over SGs). It is used for Mobility management (MM) and paging procedures between the EPS (Evolved Packet Services) and CS (Circuit Switched) domain.

21.1 VTY configuration

The SGs interface implementation in OsmoMSC is automatically active and requires only minimal configuration. When no specific configuration is provided OsmoMSC will listen on 0.0.0.0:29118 (SCTP) for incoming connections.

This is sufficient in the most configurations, but in larger installations, where services are either tied to specific interfaces and/or more instances of OsmoMSC run in parallel, a custom configuration is necessary.

The user has the option to configure the IP address (`local-ip`) and the SCTP port (`local-port`) and also the `vlr-name` that OsmoMSC uses to identify itself towards the MME. It should be noted that the `vlr-name` is usually the DNS name for the IP address of the VLR/MSC, so IP address used and the `vlr-name` should match the DNS server entry.

```
sgs
 local-ip 127.0.0.1
 local-port 29118
 vlr-name vlr.example.net
```

In order to fine tune the behavior of the SGs interface the user also has control over the relevant timers (`ts5`, `ts6-2`, `ts7`, `ts11`, `ts14`, `ts15`) and counters (`ns7`, `ns11`). Each timer and counter has to be configured separately. In the following example we change the value of ts and ns11.

```
sgs
 timer ts7 23
 counter ns11 100
```

Note
In case multiple instances of OsmoMSC run in parallel, it is advised to use a different `vlr-name` for each instance. In any case it must be ensured that the SGs interface of each instance is bound to a different IP address and/or port.

21.2 Connection monitoring

The user can use the VTY command `show sgs-connections` to list the MMEs that are currently connected to OsmoMSC.

22 Osmocom Control Interface

The VTY interface as described in Section 11 is aimed at human interaction with the respective Osmocom program.

Other programs **should not** use the VTY interface to interact with the Osmocom software, as parsing the textual representation is cumbersome, inefficient, and will break every time the formatting is changed by the Osmocom developers.

Instead, the *Control Interface* was introduced as a programmatic interface that can be used to interact with the respective program.

22.1 Control Interface Protocol

The control interface protocol is a mixture of binary framing with text based payload.

The protocol for the control interface is wrapped inside the IPA multiplex header with the stream identifier set to IPAC_PROTO_OSMO (0xEE).

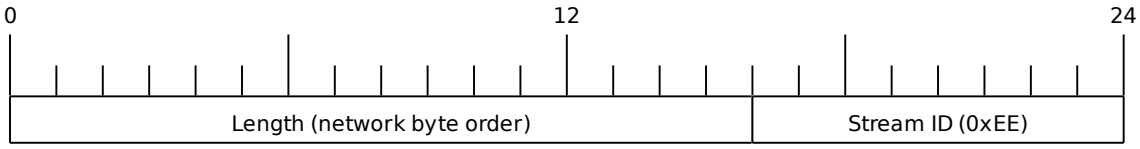


Figure 13: IPA header for control protocol

Inside the IPA header is a single byte of extension header with protocol ID 0x00 which indicates the control interface.

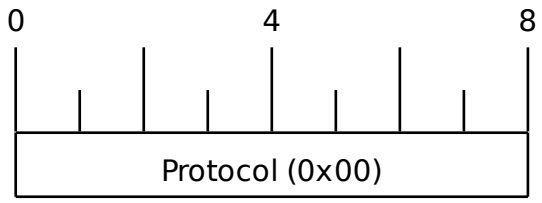


Figure 14: IPA extension header for control protocol

After the concatenation of the two above headers, the plain-text payload message starts. The format of that plain text is illustrated for each operation in the respective message sequence chart in the chapters below.

The fields specified below follow the following meaning:

<id>
A numeric identifier, uniquely identifying this particular operation. Value `0` is not allowed unless it's a TRAP message. It will be echoed back in any response to a particular request.

<var>
The name of the variable / field affected by the GET / SET / TRAP operation. Which variables/fields are available is dependent on the specific application under control.

<val>
The value of the variable / field

<reason>
A text formatted, human-readable reason why the operation resulted in an error.

22.1.1 GET operation

The GET operation is performed by an external application to get a certain value from inside the Osmocom application.

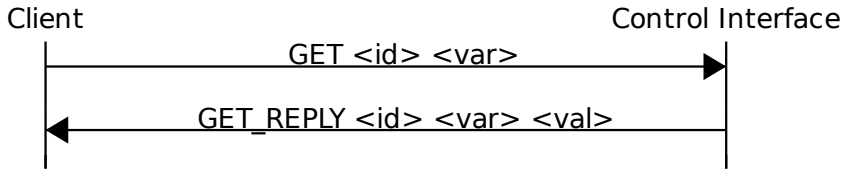


Figure 15: Control Interface GET operation (successful outcome)

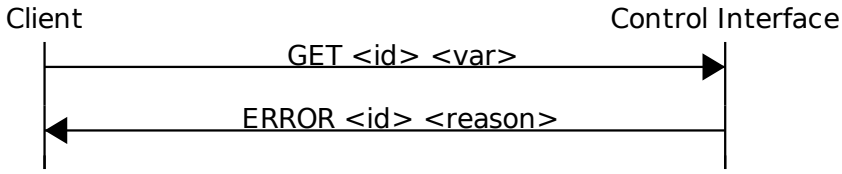


Figure 16: Control Interface GET operation (unsuccessful outcome)

22.1.2 SET operation

The SET operation is performed by an external application to set a value inside the Osmocom application.

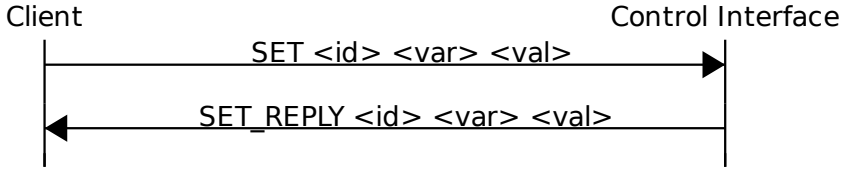


Figure 17: Control Interface SET operation (successful outcome)

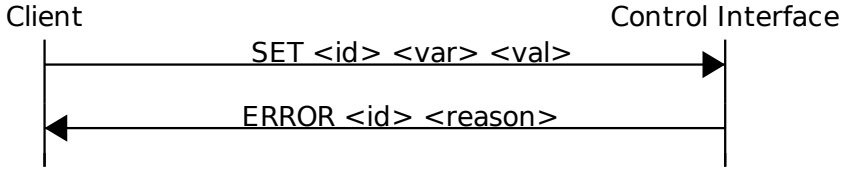


Figure 18: Control Interface SET operation (unsuccessful outcome)

22.1.3 TRAP operation

The program can at any time issue a trap. The term is used in the spirit of SNMP.

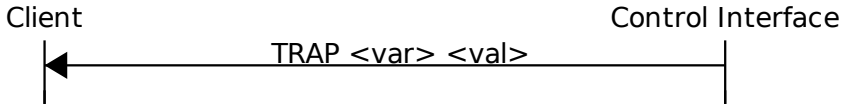


Figure 19: Control Interface TRAP operation

22.2 Common variables

There are several variables which are common to all the programs using control interface. They are described in the following table.

Table 10: Variables available over control interface

Name	Access	Value	Comment
counter.*	RO		Get counter value.
rate_ctr.*	RO		Get list of rate counter groups.
rate_ctr.IN.GN.GI.name	RO		Get value for interval IN of rate counter name which belong to group named GN with index GI.

Those read-only variables allow to get value of arbitrary counter using its name.

For example "`rate_ctr.per_hour.bsc.0.handover:timeout`" is the number of handover timeouts per hour.

Of course for that to work the program in question have to register corresponding counter names and groups using libosmocore functions.

In the example above, "`bsc`" is the rate counter group name and "`0`" is its index. It is possible to obtain all the rate counters in a given group by requesting "`rate_ctr.per_sec.bsc.*`" variable.

The list of available groups can be obtained by requesting "`rate_ctr.*`" variable.

The rate counter group name have to be prefixed with interval specification which can be any of "**per_sec**", "**per_min**", "**per_hour**", "**per_day**" or "**abs**" for absolute value.

The old-style counters available via "`counter.*`" variables are superseded by "`rate_ctr.abs`" so its use is discouraged. There might still be some applications not yet converted to rate_ctr.

22.3 Control Interface python examples

In the `osmo-python-tests` repository, there is an example python script called `scripts/osmo_ctrl.py` which implements the Osmocom control interface protocol.

You can use this tool either stand-alone to perform control interface operations against an Osmocom program, or you can use it as a reference for developing your own python software talking to the control interface.

Another implementation is in `scripts/osmo_rate_ctr2csv.py` which will retrieve performance counters for a given Osmocom program and output it in csv format. This can be used to periodically (using systemd timer for example) retrieve data to build KPI and evaluate how it changes over time.

Internally it uses "`rate_ctr.*`" variable described in Section 22.2 to get the list of counter groups and than request all the counters in each group. Applications interested in individual metrics can request it directly using `rate_ctr2csv.py` as an example.

22.3.1 Getting rate counters

Example: Use rate_ctr2csv.py to get rate counters from OsmoBSC

```
$ ./scripts/osmo_rate_ctr2csv.py --header
Connecting to localhost:4249...
Getting rate counter groups info...
"group","counter","absolute","second","minute","hour","day"
"e1inp.0","hdlc:abort","0","0","0","0","0"
"e1inp.0","hdlc:bad_fcs","0","0","0","0","0"
"e1inp.0","hdlc:overrun","0","0","0","0","0"
"e1inp.0","alarm","0","0","0","0","0"
"e1inp.0","removed","0","0","0","0","0"
```

```
"bsc.0","chreq:total","0","0","0","0","0"
"bsc.0","chreq:no_channel","0","0","0","0","0"
...
"msc.0","call:active","0","0","0","0","0"
"msc.0","call:complete","0","0","0","0","0"
"msc.0","call:incomplete","0","0","0","0","0"
Completed: 44 counters from 3 groups received.
```

22.3.2 Setting a value

Example: Use osmo_ctrl.py to set the short network name of OsmoBSC

```
$ ./osmo_ctrl.py -d localhost -s short-name 32C3
Got message: SET_REPLY 1 short-name 32C3
```

22.3.3 Getting a value

Example: Use osmo_ctrl.py to get the mnc of OsmoBSC

```
$ ./osmo_ctrl.py -d localhost -g mnc
Got message: GET_REPLY 1 mnc 262
```

22.3.4 Listening for traps

You can use `osmo_ctrl.py` to listen for traps the following way:

Example: Using osmo_ctrl.py to listen for traps:

```
$ ./osmo_ctrl.py -d localhost -m
❶
```

❶ the command will not return and wait for any TRAP messages to arrive

23 Generic Subscriber Update Protocol

23.1 General

This chapter describes the remote protocol that is used by OsmoSGSN and OsmoMSC to update and manage the local subscriber list in OsmoHLR. Functionally, it resembles the interface between the SGSN/VLR on the one hand side, and HLR/AUC on the other side.

For more information, see the specification of the Gr interface (3GPP TS 03.60).

Traditionally, the GSM MAP (Mobile Application Part) protocol is used for this purpose, running on top of a full telecom signalling protocol stack of MTP2/MTP3/SCCP/TCAP, or any of the SIGTRAN alternatives.

In order to avoid many of the complexities of MAP, which are difficult to implement in the plain C language environment of the Osmocom cellular network elements like the SGSN, we introduce the GSUP protocol.

The GSUP protocol and the messages are designed after the corresponding MAP messages (see 3GPP TS 09.02) with the following main differences:

- The encoding uses TLV structures instead of ASN.1 BER
- Segmentation is not used, i.e. we rely on the fact that the underlying transport protocol can transport signalling messages of any size.

23.2 Connection

The protocol expects that a reliable, ordered, packet boundaries preserving connection is used (e.g. IPA over TCP). The remote peer is either a service that understands the protocol natively or a wrapper service that maps the messages to/from real MAP messages that can be used to directly communicate with an HLR.

23.3 Using IPA

By default, the following identifiers should be used:

- IPA Stream ID: 0xEE (OSMO)
- IPA OSMO protocol extension: 0x05

For more information about the IPA multiplex, please see the *OsmoBTS Abis/IP Specification*.

23.4 Procedures

23.4.1 Authentication management

The SGSN or VLR sends a SEND_AUTHENTICATION_INFO_REQ message containing the MS's IMSI to the peer. On errors, especially if authentication info is not available for that IMSI, the peer returns a SEND_AUTHENTICATION_INFO_ERR message. Otherwise the peer returns a SEND_AUTHENTICATION_INFO_RES message. If this message contains at least one authentication tuple, the SGSN or VLR replaces all tuples that are assigned to the subscriber. If the message doesn't contain any tuple the SGSN or VLR may reject the Attach Request. (see 3GPP TS 09.02, 25.5.6)

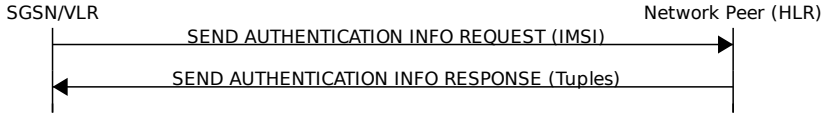


Figure 20: Send Authentication Info (Normal Case)

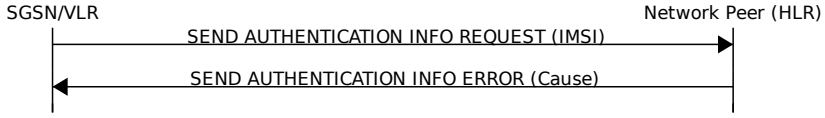


Figure 21: Send Authentication Info (Erroneous Case)

23.4.2 Reporting of Authentication Failure

Using this procedure, the SGSN or VLR reports authentication failures to the HLR.

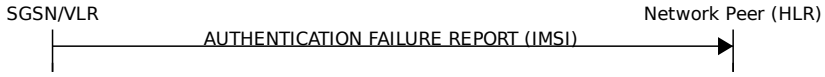


Figure 22: Authentication Failure Report (Normal Case)

23.4.3 Location Updating

The SGSN or VLR sends a UPDATE_LOCATION_REQ to the peer. If the request is denied by the network, the peer returns an UPDATE_LOCATION_ERR message to the SGSN or VLR. Otherwise the peer returns an UPDATE_LOCATION_RES message containing all information fields that shall be inserted into the subscriber record. If the *PDP info complete* information element is set in the message, the SGSN or VLR clears existing PDP information fields in the subscriber record first. (see 3GPP TS 09.02, 19.1.1.8)

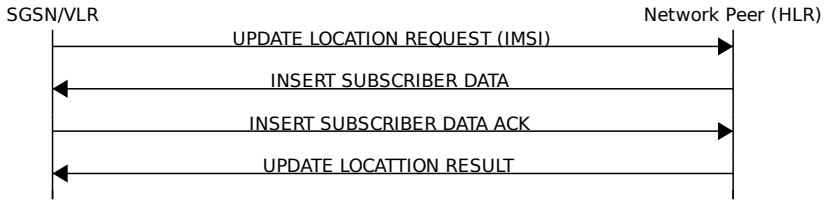


Figure 23: Update Location (Normal Case)

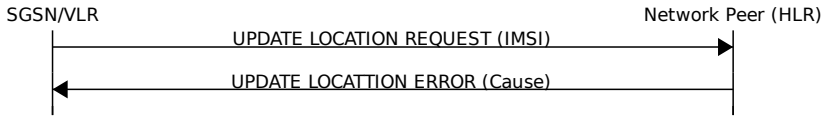


Figure 24: Update Location (Error Case)

23.4.4 Location Cancellation

Using the Location Cancellation procedure, the Network Peer (HLR) can request the SGSN or VLR to remove a subscriber record.

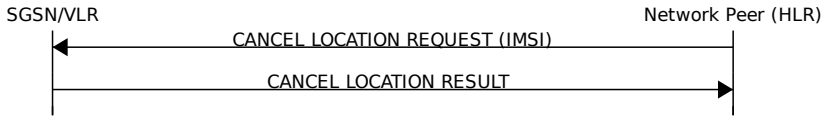


Figure 25: Cancel Location (Normal Case)

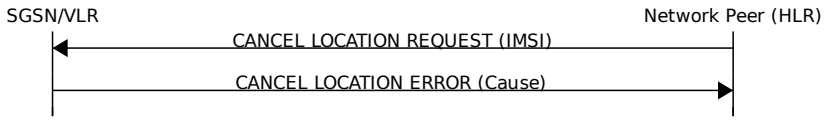


Figure 26: Cancel Location (Error Case)

23.4.5 Purge MS

Using the Purge MS procedure, the SGSN or VLR can request purging of MS related state from the HLR. It is used after the SGSN or VLR detects that no radio contact has been established for a prolonged duration (i.e. longer than the periodic LU timeout). See 3GPP TS 23.012 Section 3.6.1.4 for a description of this procedure.

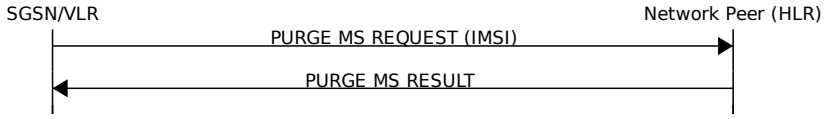


Figure 27: Purge MS (Normal Case)

23.4.6 Delete Subscriber Data

Using the Delete Subscriber Data procedure, the Peer (HLR) can remove some of the subscriber data from the SGSN or VLR. This is used in case the subscription details (e.g. PDP Contexts / APNs) change while the subscriber is registered to that SGSN VLR.

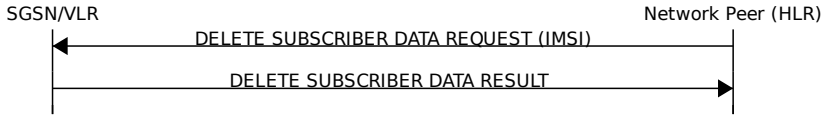


Figure 28: Delete Subscriber Data (Normal Case)

23.4.7 Check IMEI

The VLR asks the EIR to check if a new ME's IMEI is acceptable or not. The EIR may implement a blacklist or whitelist and reject the IMEI based on that. Against the original purpose of the Check IMEI Procedure, this could also be used to save the IMEI in the HLR DB.

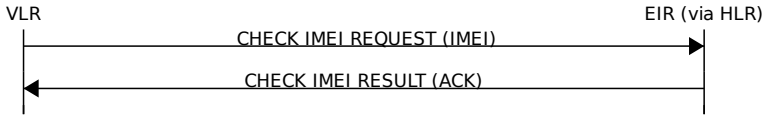


Figure 29: Check IMEI (Normal Case)

23.5 Procedures (E Interface)

The E interface connects two MSCs in the traditional GSM MAP world. It is used for the inter-MSC handover. In GSUP, we don't need that extra connection, as we route the messages over the GSUP server (OsmoHLR) instead.

Whenever MSC-A is sending to MSC-B, and vice-versa, the message needs to pass through the GSUP server. In order to make the following message sequence charts easier to read, this step has been omitted.

23.5.1 E Handover

MSC-A has an active RAN connection and hands it over to MSC-B.

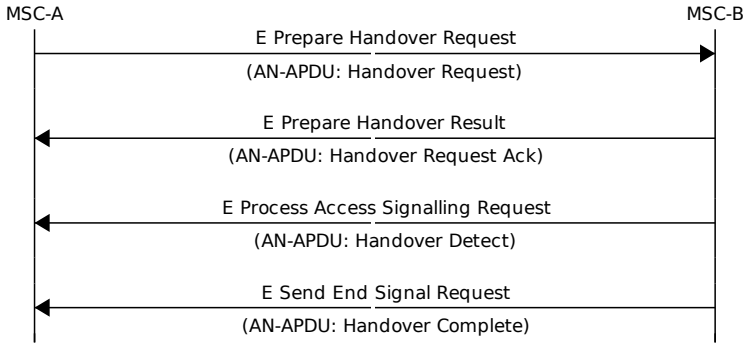


Figure 30: E Handover (Normal Case)

23.5.2 E Subsequent Handover

MSC-B has an active RAN connection, and asks MSC-A to hand it over to MSC-B'.

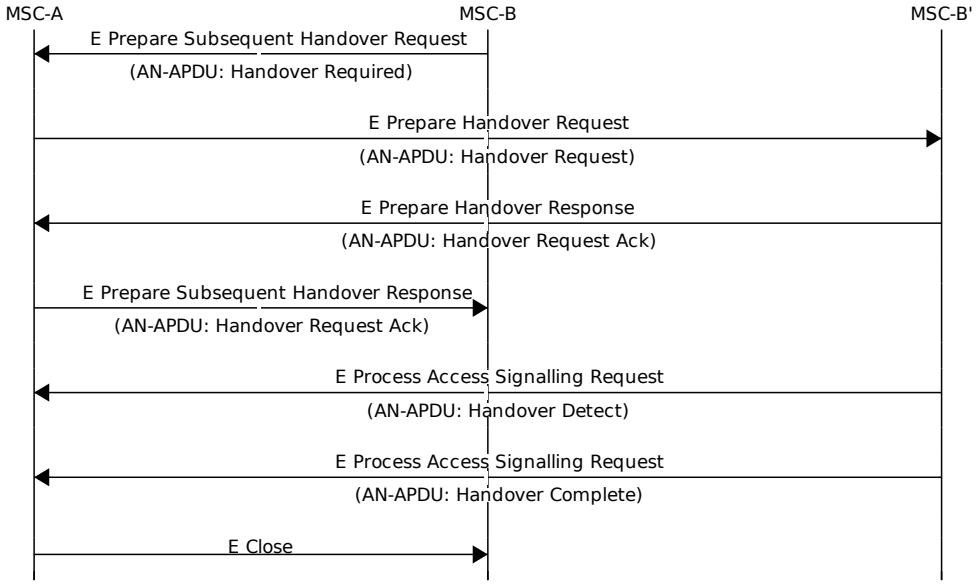


Figure 31: E Subsequent Handover (Normal Case)

23.5.3 E Forward and Process Access Signalling

MSC-A is forwarding a message from its BSS (Base Station Subsystem) to MSC-B. MSC-B forwards the message to its BSS, and answers to MSC-A with a Process Access Signalling Request.

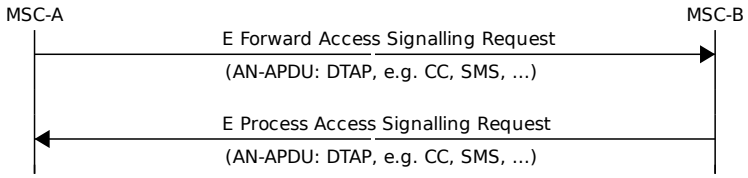


Figure 32: E Process and Forward Access Signalling (Normal Case)

23.5.4 E Routing Error

The GSUP server can not route any of the requests above, and responds with an E Routing Error. Possible reasons for not being able to route the message are missing routing IEs, a mismatching source name IE (Section 23.7.31), the destination not being connected to the GSUP server or a failed attempt to send the message from the GSUP sever to the destination. To figure out, what went wrong in detail, refer to the GSUP server's logs.

In the traditional GSM MAP world, the participants of an E procedure are directly connected, hence this routing error message does not exist in MAP.

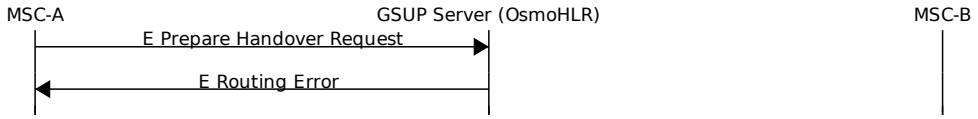


Figure 33: E Routing Error example

23.6 Message Format

23.6.1 General

Every message is based on the following message format

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10

If a numeric range is indicated in the *presence* column, multiple information elements with the same tag may be used in sequence. The information elements shall be sent in the given order. Nevertheless after the generic part the receiver shall be able to received them in any order. Unknown IE shall be ignored.

Besides a numeric range, the *presence* column may have *M* (Mandatory), *O* (Optional) or *C* (Conditional). The *format* column holds either *V* (Value) or *TLV* (Tag Length Value).

23.6.2 Send Authentication Info Request

Direction: SGSN / VLR ⇒ HLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
28	CN Domain	Section 23.7.15	O	TLV	3
26	AUTS	Section 23.7.13	C	TLV	18
20	RAND	Section 23.7.7	C	TLV	18
05	PDP info	Section 23.7.3	C	TLV	2-N
2a	Current RAT Type	Section 23.7.36	O	TLV	3

The conditional *AUTS* and *RAND* IEs are both present in case the SIM (via UE) requests an UMTS AKA re-synchronization procedure. Either both optional IEs are present, or none of them.

The conditional *PDP Info* IE is only present in the CEAI interface used by the ePDG. It should contain the *PDP Context ID*, *PDP Address* (dynamic addressing) and *Access Point Name* IEs.

23.6.3 Send Authentication Info Error

Direction: HLR ⇒ SGSN / VLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
02	Cause	Section 23.7.26	M	TLV	3

23.6.4 Send Authentication Info Response

Direction: HLR ⇒ SGSN / VLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
03	Auth Tuple	Section 23.7.6	0-5	TLV	36

23.6.5 Authentication Failure Report

Direction: SGSN / VLR ⇒ HLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
28	CN Domain	Section 23.7.15	O	TLV	3

23.6.6 Update Location Request

Direction: SGSN / VLR ⇒ HLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
28	CN Domain	Section 23.7.15	O	TLV	3

23.6.7 Update Location Error

Direction: HLR ⇒ SGSN / VLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
02	Cause	Section 23.7.26	M	TLV	3

23.6.8 Update Location Result

Direction: HLR ⇒ SGSN / VLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
08	MSISDN	Section 23.7.20	O	TLV	0-9

IEI	IE	Type	Presence	Format	Length
09	HLR Number	Section 23.7.25	O	TLV	0-9
04	PDP info complete	Section 23.7.18	O	TLV	2
05	PDP info	Section 23.7.3	O	TLV	2-N

If the PDP info complete IE is present, the old PDP info list shall be cleared.

23.6.9 Location Cancellation Request

Direction: HLR ⇒ SGSN / VLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
28	CN Domain	Section 23.7.15	O	TLV	3
06	Cancellation type	Section 23.7.16	O	TLV	3

23.6.10 Location Cancellation Error

Direction: SGSN / VLR ⇒ HLR

TODO

23.6.11 Location Cancellation Result

Direction: SGSN / VLR ⇒ HLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
28	CN Domain	Section 23.7.15	O	TLV	3

23.6.12 Purge MS Request

Direction: SGSN / VLR ⇒ HLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
28	CN Domain	Section 23.7.15	O	TLV	3
09	HLR Number	Section 23.7.25	M	TLV	0-9

23.6.13 Purge MS Error

Direction: HLR ⇒ SGSN / VLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
02	Cause	Section 23.7.26	M	TLV	3

23.6.14 Purge MS Result

Direction: HLR ⇒ SGSN / VLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
07	Freeze P-TMSI	Section 23.7.18	M	TLV	2

23.6.15 Insert Subscriber Data Request

Direction: HLR ⇒ SGSN / VLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
28	CN Domain	Section 23.7.15	O	TLV	3
08	MSISDN	Section 23.7.20	O	TLV	0-9
09	HLR Number	Section 23.7.25	O	TLV	0-9
04	PDP info complete	Section 23.7.18	M	TLV	2
05	PDP info	Section 23.7.3	C	TLV	0-10
14	PDP-Charging Characteristics	Section 23.7.23	O	TLV	4

If the PDP info complete IE is present, the old PDP info list shall be cleared.

23.6.16 Insert Subscriber Data Error

Direction: SGSN / VLR ⇒ HLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
02	Cause	Section 23.7.26	M	TLV	3

23.6.17 Insert Subscriber Data Result

Direction: SGSN / VLR ⇒ HLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10

23.6.18 Delete Subscriber Data Request

Direction: HLR ⇒ SGSN / VLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
28	CN Domain	Section 23.7.15	O	TLV	3
10	PDP Context ID	Section 23.7.5	C	TLV	3

23.6.19 Delete Subscriber Data Error

Direction: SGSN / VLR ⇒ HLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
02	Cause	Section 23.7.26	M	TLV	3

23.6.20 Delete Subscriber Data Result

Direction: HLR ⇒ SGSN / VLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10

23.6.21 Process Supplementary Service Request

Direction: bidirectional

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
30	Session ID	Section 23.8.1	M	TLV	6
31	Session State	Section 23.8.2	M	TLV	3
35	Supplementary Service Info	Section 23.7.27	O	TLV	2-...

This message is used in both directions in case of USSD, because it is not known is it request or response without parsing the GSM 04.80 payload.

23.6.22 Process Supplementary Service Error

Direction: EUSE / HLR ⇒ MSC

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
30	Session ID	Section 23.8.1	M	TLV	6
31	Session State	Section 23.8.2	M	TLV	3
02	Cause	Section 23.7.26	M	TLV	3

23.6.23 Process Supplementary Service Response

Direction: EUSE / HLR ⇒ MSC

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
30	Session ID	Section 23.8.1	M	TLV	6
31	Session State	Section 23.8.2	M	TLV	3
35	Supplementary Service Info	Section 23.7.27	O	TLV	2-...

The purpose of this message is not clear yet. Probably, it can be used to notify the MSC that a structured supplementary service is successfully activated or deactivated, etc.

23.6.24 MO-forwardSM Request

Direction: MSC / SGSN ⇒ SMSC (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
40	SM-RP-MR (Message Reference)	Section 23.8.3	M	TLV	1
41	SM-RP-DA (Destination Address)	Section 23.8.4	M	TLV	2-...
42	SM-RP-OA (Originating Address)	Section 23.8.5	M	TLV	2-...
43	SM-RP-UI (SM TPDU)	Section 23.8.7	M	TLV	1-...

This message is used to forward MO short messages from MSC / SGSN to an SMSC. The corresponding MAP service is MAP-MO-FORWARD-SHORT-MESSAGE, see 3GPP TS 29.002, section 12.2.

23.6.25 MO-forwardSM Error

Direction: SMSC (via HLR) ⇒ MSC / SGSN

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
40	SM-RP-MR (Message Reference)	Section 23.8.3	M	TLV	1
44	SM-RP-Cause (Cause value)	Section 23.8.8	M	TLV	1
43	SM-RP-UI (diagnostic field)	Section 23.8.7	O	TLV	1-...

This message is used to indicate a negative result of an earlier MO short message delivery. The corresponding MAP service is MAP-MO-FORWARD-SHORT-MESSAGE, see 3GPP TS 29.002, section 12.2.

23.6.26 MO-forwardSM Result

Direction: SMSC (via HLR) ⇒ MSC / SGSN

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
40	SM-RP-MR (Message Reference)	Section 23.8.3	M	TLV	1

This message is used to indicate a successful result of an earlier MO short message delivery. The corresponding MAP service is MAP-MO-FORWARD-SHORT-MESSAGE, see 3GPP TS 29.002, section 12.2.

23.6.27 MT-forwardSM Request

Direction: SMSC (via HLR) ⇒ MSC / SGSN

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
40	SM-RP-MR (Message Reference)	Section 23.8.3	M	TLV	1

IEI	IE	Type	Presence	Format	Length
41	SM-RP-DA (Destination Address)	Section 23.8.4	M	TLV	2-...
42	SM-RP-OA (Originating Address)	Section 23.8.5	M	TLV	2-...
43	SM-RP-UI (SM TPDU)	Section 23.8.7	M	TLV	1-...
45	SM-RP-MMS (More Messages to Send)	Section 23.8.9	O	TLV	1

This message is used to forward MT short messages from an SMSC to MSC / SGSN. The corresponding MAP service is MAP-MT-FORWARD-SHORT-MESSAGE, see 3GPP TS 29.002, section 12.9.

23.6.28 MT-forwardSM Error

Direction: MSC / SGSN ⇒ SMSC (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
40	SM-RP-MR (Message Reference)	Section 23.8.3	M	TLV	1
44	SM-RP-Cause (Cause value)	Section 23.8.8	M	TLV	1
43	SM-RP-UI (diagnostic field)	Section 23.8.7	O	TLV	1-...

This message is used to indicate a negative result of an earlier MT short message delivery. The corresponding MAP service is MAP-MT-FORWARD-SHORT-MESSAGE, see 3GPP TS 29.002, section 12.9.

23.6.29 MT-forwardSM Result

Direction: MSC / SGSN ⇒ SMSC (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
40	SM-RP-MR (Message Reference)	Section 23.8.3	M	TLV	1

This message is used to indicate a successful result of an earlier MT short message delivery. The corresponding MAP service is MAP-MT-FORWARD-SHORT-MESSAGE, see 3GPP TS 29.002, section 12.9.

23.6.30 READY-FOR-SM Request

Direction: MSC / SGSN ⇒ SMSC (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
40	SM-RP-MR (Message Reference)	Section 23.8.3	M	TLV	1
46	SM Alert Reason	Section 23.8.10	M	TLV	1-...

This message is used between the MSC / SGSN and an SMSC when a subscriber indicates memory available situation (see TS GSM 04.11, section 7.3.2). The corresponding MAP service is MAP-READY-FOR-SM, see 3GPP TS 29.002, section 12.4.

23.6.31 READY-FOR-SM Error

Direction: SMSC (via HLR) ⇒ MSC / SGSN

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
40	SM-RP-MR (Message Reference)	Section 23.8.3	M	TLV	1
44	SM-RP-Cause (Cause value)	Section 23.8.8	M	TLV	1
43	SM-RP-UI (diagnostic field)	Section 23.8.7	O	TLV	1-...

This message is used to indicate a negative result of an earlier MO SMMA (Memory Available) indication. The corresponding MAP service is MAP-READY-FOR-SM, see 3GPP TS 29.002, section 12.4.

23.6.32 READY-FOR-SM Result

Direction: SMSC (via HLR) ⇒ MSC / SGSN

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
40	SM-RP-MR (Message Reference)	Section 23.8.3	M	TLV	1

This message is used to indicate a successful result of an earlier MO SMMA (Memory Available) indication. The corresponding MAP service is MAP-READY-FOR-SM, see 3GPP TS 29.002, section 12.4.

23.6.33 CHECK-IMEI Request

Direction: VLR ⇒ EIR (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
50	IMEI	Section 23.7.28	M	TLV	11

23.6.34 CHECK-IMEI Error

Direction: EIR (via HLR) ⇒ VLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
02	Cause	Section 23.7.26	M	TLV	3

23.6.35 CHECK-IMEI Result

Direction: EIR (via HLR) ⇒ VLR

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
51	IMEI Check Result	Section 23.7.29	M	TLV	3

23.6.36 E Prepare Handover Request

Direction: MSC-A=MSC-I ⇒ MSC-B=MSC-T (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
0a	Message Class	Section 23.7.30	M	TLV	3
60	Source Name	Section 23.7.31	M	TLV	2-...
61	Destination Name	Section 23.7.32	M	TLV	2-...
62	AN-APDU	Section 23.7.33	M	TLV	2-...

23.6.37 E Prepare Handover Error

Direction: MSC-B=MSC-T ⇒ MSC-A=MSC-I (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
0a	Message Class	Section 23.7.30	M	TLV	3
60	Source Name	Section 23.7.31	M	TLV	2-...
61	Destination Name	Section 23.7.32	M	TLV	2-...
62	AN-APDU	Section 23.7.33	M	TLV	2-...

23.6.38 E Prepare Handover Result

Direction: MSC-B=MSC-T ⇒ MSC-A=MSC-I (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
0a	Message Class	Section 23.7.30	M	TLV	3
60	Source Name	Section 23.7.31	M	TLV	2-...
61	Destination Name	Section 23.7.32	M	TLV	2-...
62	AN-APDU	Section 23.7.33	M	TLV	2-...

23.6.39 E Prepare Subsequent Handover Request

Direction: MSC-B=MSC-I ⇒ MSC-A (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
0a	Message Class	Section 23.7.30	M	TLV	3
60	Source Name	Section 23.7.31	M	TLV	2-...
61	Destination Name	Section 23.7.32	M	TLV	2-...
62	AN-APDU	Section 23.7.33	M	TLV	2-...

23.6.40 E Prepare Subsequent Handover Error

Direction: MSC-A ⇒ MSC-B=MSC-I (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
0a	Message Class	Section 23.7.30	M	TLV	3
60	Source Name	Section 23.7.31	M	TLV	2-...
61	Destination Name	Section 23.7.32	M	TLV	2-...
62	AN-APDU	Section 23.7.33	M	TLV	2-...

23.6.41 E Prepare Subsequent Handover Result

Direction: MSC-A ⇒ MSC-B=MSC-I (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
0a	Message Class	Section 23.7.30	M	TLV	3
60	Source Name	Section 23.7.31	M	TLV	2-...
61	Destination Name	Section 23.7.32	M	TLV	2-...
62	AN-APDU	Section 23.7.33	M	TLV	2-...

23.6.42 E Send End Signal Request

Direction: MSC-B=MSC-T ⇒ MSC-A=MSC-I (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
0a	Message Class	Section 23.7.30	M	TLV	3
60	Source Name	Section 23.7.31	M	TLV	2-...
61	Destination Name	Section 23.7.32	M	TLV	2-...
62	AN-APDU	Section 23.7.33	M	TLV	2-...

23.6.43 E Send End Signal Error

Direction: MSC-A=MSC-I ⇒ MSC-B=MSC-T (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
0a	Message Class	Section 23.7.30	M	TLV	3
60	Source Name	Section 23.7.31	M	TLV	2-...
61	Destination Name	Section 23.7.32	M	TLV	2-...
62	AN-APDU	Section 23.7.33	M	TLV	2-...

23.6.44 E Send End Signal Result

Direction: MSC-A ⇒ MSC-B=MSC-I (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
0a	Message Class	Section 23.7.30	M	TLV	3
60	Source Name	Section 23.7.31	M	TLV	2-...

IEI	IE	Type	Presence	Format	Length
61	Destination Name	Section 23.7.32	M	TLV	2-...
62	AN-APDU	Section 23.7.33	M	TLV	2-...

23.6.45 E Process Access Signalling Request

Direction: MSC-B=MSC-T ⇒ MSC-A=MSC-I (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
0a	Message Class	Section 23.7.30	M	TLV	3
60	Source Name	Section 23.7.31	M	TLV	2-...
61	Destination Name	Section 23.7.32	M	TLV	2-...
62	AN-APDU	Section 23.7.33	M	TLV	2-...

23.6.46 E Forward Access Signalling Request

Direction: MSC-A ⇒ MSC-B=MSC-I (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
0a	Message Class	Section 23.7.30	M	TLV	3
60	Source Name	Section 23.7.31	M	TLV	2-...
61	Destination Name	Section 23.7.32	M	TLV	2-...
62	AN-APDU	Section 23.7.33	M	TLV	2-...

23.6.47 E Close

Direction: MSC-A ⇒ MSC-B (via HLR)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
0a	Message Class	Section 23.7.30	M	TLV	3
60	Source Name	Section 23.7.31	M	TLV	2-...
61	Destination Name	Section 23.7.32	M	TLV	2-...

23.6.48 E Abort

This message was added to GSUP for the inter-MSC handover. But so far it is not used yet.

23.6.49 E Routing Error

Direction: GSUP Server (HLR) ⇒ GSUP Client (MSC)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
0a	Message Class	Section 23.7.30	M	TLV	3
60	Source Name	Section 23.7.31	M	TLV	2-...
61	Destination Name	Section 23.7.32	M	TLV	2-...

IEI	IE	Type	Presence	Format	Length
30	Session ID	Section 23.8.1	O	TLV	6
31	Session State	Section 23.8.2	O	TLV	3

23.6.50 ePDG Tunnel Request

Direction: GSUP Client (strongswan) ⇒ GSUP Server (ePDG)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
0a	Message Class	Section 23.7.30	M	TLV	3
15	PCO	Section 23.7.24	O	TLV	1-...
28	CN Domain	Section 23.7.15	O	TLV	3
60	Source Name	Section 23.7.31	O	TLV	2-...

23.6.51 ePDG Tunnel Error

Direction: GSUP Server (ePDG) ⇒ GSUP Client (strongswan)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
0a	Message Class	Section 23.7.30	M	TLV	3
02	Cause	Section 23.7.26	M	TLV	3

23.6.52 ePDG Tunnel Result

Direction: GSUP Server (ePDG) ⇒ GSUP Client (strongswan)

IEI	IE	Type	Presence	Format	Length
	Message Type	Section 23.7.1	M	V	1
01	IMSI	Section 23.7.19	M	TLV	2-10
0a	Message Class	Section 23.7.30	M	TLV	3
04	PDP info complete	Section 23.7.18	M	TLV	2
05	PDP info	Section 23.7.3	M	TLV	2-N
15	PCO	Section 23.7.24	O	TLV	1-...
28	CN Domain	Section 23.7.15	O	TLV	3
60	Source Name	Section 23.7.31	O	TLV	2-...

23.7 Information Elements

23.7.1 Message Type

Type	Name
Description	0x04
Update Location Request	Section 23.6.6
0x05	Update Location Error
Section 23.6.7	0x06

Type	Name
Update Location Result	Section 23.6.8
0x08	Send Authentication Info Request
Section 23.6.2	0x09
Send Authentication Info Error	Section 23.6.3
0x0a	Send Authentication Info Result
Section 23.6.4	0x0b
Authentication Failure Report	Section 23.6.5
0x0c	Purge MS Request
[?]	0x0d
Purge MS Error	Section 23.6.13
0x0e	Purge MS Result
Section 23.6.14	0x10
Insert Subscriber Data Request	Section 23.6.15
0x11	Insert Subscriber Data Error
Section 23.6.16	0x12
Insert Subscriber Data Result	Section 23.6.17
0x14	Delete Subscriber Data Request
Section 23.6.18	0x15
Delete Subscriber Data Error	Section 23.6.19
0x16	Delete Subscriber Data Result
Section 23.6.20	0x1c
Location Cancellation Request	Section 23.6.9
0x1d	Location Cancellation Error
Section 23.6.10	0x1e
Location Cancellation Result	Section 23.6.11
0x20	Supplementary Service Request
Section 23.6.21	0x21
Supplementary Service Error	Section 23.6.22
0x22	Supplementary Service Result
Section 23.6.23	0x24
MO-forwardSM Request	Section 23.6.24
0x25	MO-forwardSM Error
Section 23.6.25	0x26

Type	Name
MO-forwardSM Result	Section 23.6.26
0x28	MT-forwardSM Request
Section 23.6.27	0x29
MT-forwardSM Error	Section 23.6.28
0x2a	MT-forwardSM Result
Section 23.6.29	0x2c
READY-FOR-SM Request	Section 23.6.30
0x2d	READY-FOR-SM Error
Section 23.6.31	0x2e
READY-FOR-SM Result	Section 23.6.32
0x30	CHECK-IMEI Request
Section 23.6.33	0x31
CHECK-IMEI Error	Section 23.6.34
0x32	CHECK-IMEI Result
Section 23.6.35	0x34
E Prepare Handover Request	Section 23.6.36
0x35	E Prepare Handover Error
Section 23.6.37	0x36
E Prepare Handover Result	Section 23.6.38
0x38	E Prepare Subsequent Handover Request
Section 23.6.39	0x39
E Prepare Subsequent Handover Error	Section 23.6.40
0x3a	E Prepare Subsequent Handover Result
Section 23.6.41	0x3c
E Send End Signal Request	Section 23.6.42
0x3d	E Send End Signal Error
Section 23.6.43	0x3e
E Send End Signal Result	Section 23.6.44
0x40	E Process Access Signalling Request
Section 23.6.45	0x44
E Forward Access Signalling Request	Section 23.6.46
0x47	E Close
Section 23.6.47	0x4B

Type	Name
E Abort	Section 23.6.48
0x4E	E Routing Error
Section 23.6.49	0x50
ePDG Tunnel Request	Section 23.6.50
0x51	ePDG Tunnel Error
Section 23.6.51	0x52
ePDG Tunnel Result	Section 23.6.52

The category of the message is indicated by the last two bits of the type. Request, Error and Result messages only differ in these last two bits, so it is trivial to transform them.

Ending Bits	Message Category
00	Request
01	Error
10	Result
11	Other

23.7.2 IP Address

The value part is encoded like in the Packet data protocol address IE defined in 3GPP TS 24.008, Chapter 10.5.6.4. PDP type organization must be set to *IETF allocated address*.

23.7.3 PDP Info

This is a container for information elements describing a single PDP.

IEI	IE	Type	Presence	Format	Length
	PDP Info IEI	Section 23.7.17	M	V	1
	Length of PDP Info IE		M	V	1
10	PDP Context ID	Section 23.7.5	C	TLV	3
11	PDP Address	Section 23.7.4	C	TLV	4-24
12	Access Point Name	Section 23.7.21	C	TLV	3-102
13	Quality of Service	Section 23.7.22	O	TLV	1-20
14	PDP-Charging Characteristics	Section 23.7.23	O	TLV	4

The conditional IE are mandantory unless mentioned otherwise.

23.7.4 PDP Address

The value part is encoded like in the Packet data protocol address IE defined in 3GPP TS 24.008, Chapter 10.5.6.4. Hence this value also relates to End User Address (EUA) IE defined in 3GPP TS 29.060, 7.7.27. The PDP type organization value must be set to *IETF allocated address*.

0
4
8
PDP type IEI
Res
Length
PDP type org
Spare
PDP type number
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)
Address (v4|v6|v4+v6)

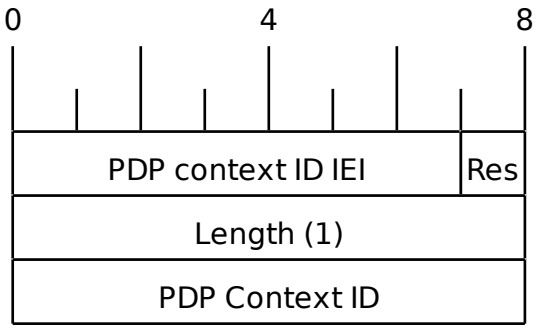
The spare bits are left undefined. While 3GPP TS 29.060 7.7.27 defines them as *1 1 1 1*, there are MAP traces where these bits are set to *0 0 0 0*. So the receiver shall ignore these bits.

Examples:

- IPv4: PDP type org: 1 (IETF), PDP type number: 0x21, 0 bytes address (dynamic addressing)
- IPv4: PDP type org: 1 (IETF), PDP type number: 0x21, 4 bytes address
- IPv6: PDP type org: 1 (IETF), PDP type number: 0x57, 16 bytes address
- IPv6: PDP type org: 1 (IETF), PDP type number: 0x8D, 20 bytes address (v4+v6)

23.7.5 PDP Context ID

The PDP type context ID IE consists of a single integer byte wrapped in a TLV. Values for PDP Context ID start at 1 (0 is reserved).



23.7.6 Auth tuple

This is a container for information elements describing a single authentication tuple.

IEI	IE	Type	Presence	Format	Length
	Auth Tuple IEI	Section 23.7.17	M	V	1
	Length of Auth Tuple IE		M	V	1
20	RAND	Section 23.7.7	M	TLV	18
21	SRES	Section 23.7.8	M	TLV	6
22	Kc	Section 23.7.9	M	TLV	10
23	IK	Section 23.7.10	C	TLV	18
24	CK	Section 23.7.11	C	TLV	18
25	AUTN	Section 23.7.12	C	TLV	18
27	RES	Section 23.7.14	C	TLV	2-18

The conditional IEs *IK*, *CK*, *AUTN* and *RES* are only present in case the subscriber supports UMTS AKA.

23.7.7 RAND

The 16-byte Random Challenge of the GSM Authentication Algorithm.

23.7.8 SRES

The 4-byte Authentication Result of the GSM Authentication Algorithm.

23.7.9 Kc

The 8-byte Encryption Key of the GSM Authentication and Key Agreement Algorithm.

23.7.10 IK

The 16-byte Integrity Protection Key generated by the UMTS Authentication and Key Agreement Algorithm.

23.7.11 CK

The 16-byte Ciphering Key generated by the UMTS Authentication and Key Agreement Algorithm.

23.7.12 AUTN

The 16-byte Authentication Nonce sent from network to USIM in the UMTS Authentication and Key Agreement Algorithm.

23.7.13 AUTS

The 14-byte Authentication Synchronization Nonce generated by the USIM in case the UMTS Authentication and Key Agreement Algorithm needs to re-synchronize the sequence counters between AUC and USIM.

23.7.14 RES

The (variable length, but typically 16 byte) Authentication Result generated by the USIM in the UMTS Authentication and Key Agreement Algorithm.

23.7.15 CN Domain

This single-byte information element indicates the Core Network Domain, i.e. if the message is related to Circuit Switched or Packet Switched services.

For backwards compatibility reasons, if no CN Domain IE is present within a request, the PS Domain is assumed.

Table 11: CN Domain Number

Type	Description
0x01	PS Domain
0x02	CS Domain

23.7.16 Cancellation Type

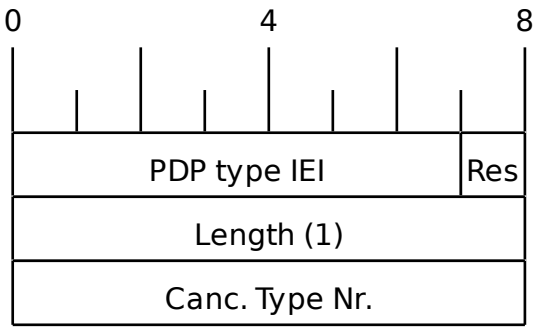


Table 12: Cancellation Type Number

Number	Description
0x00	Update Procedure
0x01	Subscription Withdrawn

23.7.17 IE Identifier (informational)

These are the standard values for the IEI. See the message definitions for the IEI that shall be used for the encoding.

Table 13: GSUP IE Identifiers

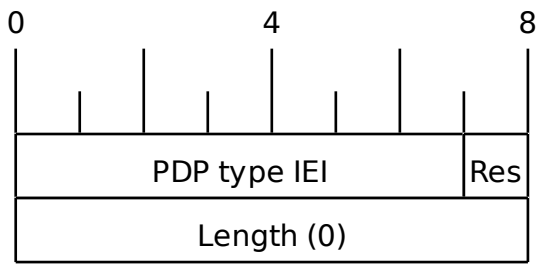
IEI	Info Element	Type / Encoding
0x01	IMSI	Mobile Identity, 3GPP TS 24.008 Ch. 10.5.1.4
0x02	Cause	Section 23.7.26
0x03	Auth Tuple	Section 23.7.6
0x04	PDP Info Compl	Section 23.7.18
0x05	PDP Info	Section 23.7.3
0x06	Cancel Type	Section 23.7.16
0x07	Freeze P-TMSI	Section 23.7.18
0x08	MSISDN	ISDN-AddressString/octet, Section 23.7.20
0x09	HLR Number	Section 23.7.25
0x0a	Message Class	Section 23.7.30
0x10	PDP Context ID	Section 23.7.5
0x11	PDP Address	[?]
0x12	Access Point Name	Section 23.7.21
0x13	QoS	Section 23.7.22
0x14	PDP-Charging Characteristics	Section 23.7.23
0x15	PCO	Section 23.7.24
0x20	RAND	Section 23.7.7
0x21	SRES	Section 23.7.8
0x22	Kc	Section 23.7.9
0x23	IK	Section 23.7.10

Table 13: (continued)

IEI	Info Element	Type / Encoding
0x24	CK	Section 23.7.11
0x25	AUTN	Section 23.7.12
0x26	AUTS	Section 23.7.13
0x27	RES	Section 23.7.14
0x28	CN Domain	Section 23.7.15
0x30	Session ID	Section 23.8.1
0x31	Session State	Section 23.8.2
0x35	Supplementary Service Info	Section 23.7.27
0x40	SM-RP-MR (Message Reference)	Section 23.8.3
0x41	SM-RP-DA (Destination Address)	Section 23.8.4
0x42	SM-RP-OA (Originating Address)	Section 23.8.5
0x43	SM-RP-UI (SM TPDU)	Section 23.8.7
0x44	SM-RP-Cause (RP Cause value)	Section 23.8.8
0x45	SM-RP-MMS (More Messages to Send)	Section 23.8.9
0x46	SM Alert Reason	Section 23.8.10
0x50	IMEI	Section 23.7.28
0x51	IMEI Check Result	Section 23.7.29
0x60	Source Name	Section 23.7.31
0x61	Destination Name	Section 23.7.32
0x62	AN-APDU	Section 23.7.33
0x63	RR Cause	Section 23.7.34
0x64	BSSAP Cause	Section 23.7.35
0x65	Session Management Cause	Section 23.7.37

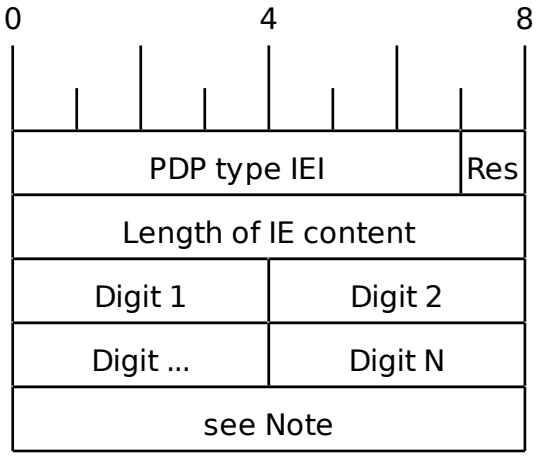
23.7.18 Empty field

This is used for flags, if and only if this IE is present, the flag is set. The semantics depend on the IEI and the context.



23.7.19 IMSI

The IMSI is encoded like in octet 4-N of the Called Party BCD Number defined in 3GPP TS 24.008, 10.5.4.7.

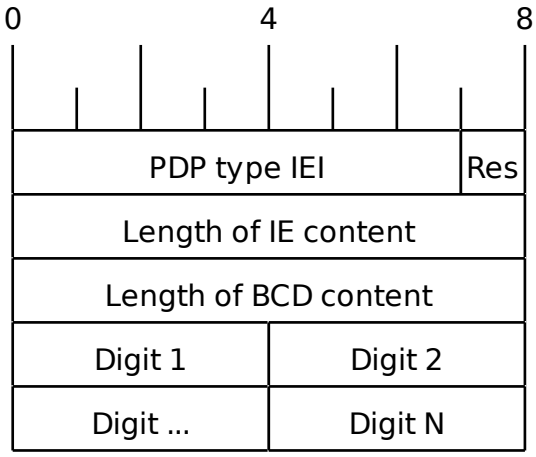


Note
Either *1 1 1 1 | Number digit N* (N odd) or *Number digit N | Number digit N-1* (N even), where N is the number of digits.

23.7.20 ISDN-AddressString / MSISDN / Called Party BCD Number

The MSISDN is encoded as an ISDN-AddressString in 3GPP TS 09.02 and Called Party BCD Number in 3GPP TS 24.008. It will be stored by the SGSN or VLR and then passed as is to the GGSN during the activation of the primary PDP Context.

The length of the BCD content specifies how many subsequent octets contain BCD data (it does **not** indicate the number of digits). This length must be less than the total length of the IE content. Any octets following the BCD content shall be ignored.

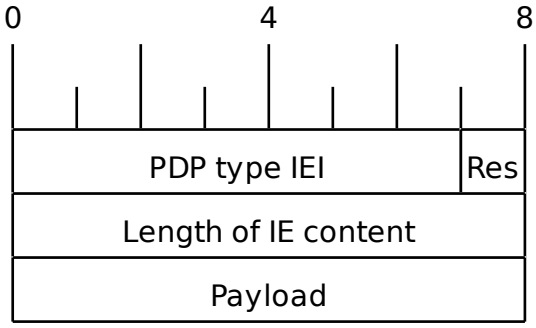


23.7.21 Access Point Name

This encodes the Access Point Name of a PDP Context. The encoding is defined in 3GPP TS 23.003.

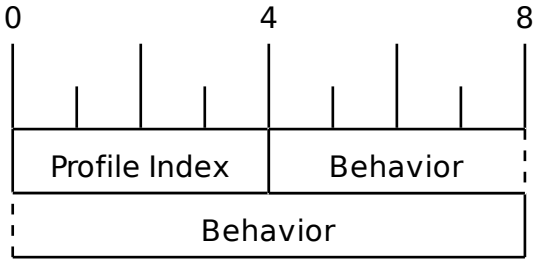
23.7.22 Quality of Service Subscribed Service

This encodes the subscribed QoS of a subscriber. It will be used by the SGSN during the PDP Context activation. If the length of the QoS data is 3 (three) octets it is assumed that these are octets 3-5 of the TS 3GPP TS 24.008 Quality of Service Octets. If it is more than three then then it is assumed that the first octet is the Allocation/Retention Priority and the rest are encoded as octets 3-N of 24.008.



23.7.23 PDP-Charging Characteristics

This encodes the ChargingCharacteristics of 3GPP TS 32.215. A HLR may send this as part of the InsertSubscriberData or within a single PDP context definition. If the HLR supplies this information it must be used by the SGSN or VLR when activating a PDP context.

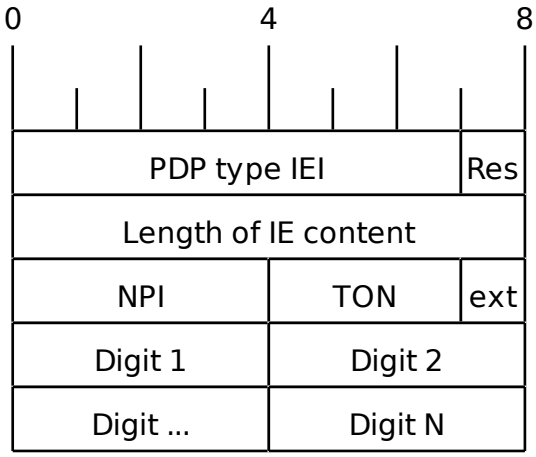


23.7.24 Protocol Configuration Options (PCO)

This encodes the Protocol Configuration Options (PCO) of 3GPP TS 29.060 clause 7.7.31, which are the same as those specified in 3GPP TS 24.008 10.5.6.3. It will be used by the ePDG during the PDP Context activation.

23.7.25 HLR Number encoded as 3GPP TS 09.02 ISDN-AddressString

The HLR Number is encoded as an ISDN-AddressString in 3GPP TS 09.02. It will be stored by the SGSN or VLR can be used by the CDR module to keep a record.



23.7.26 Cause

This IE shall be encoded according to the *GMM Cause* as described in Chapter 10.5.5.14 of 3GPP TS 24.008.

23.7.27 Supplementary Service Info

This IE shall be used together with both Section 23.8.2 and Section 23.8.1 IEs. It is used to carry the payload of Supplementary Services encoded according to GSM TS 04.80.

23.7.28 IMEI

The IMEI encoded as Called Party BCD Number in 3GPP TS 24.008. This IE has the same format as described in Section 23.7.20.

23.7.29 IMEI Check Result

Result of the Check IMEI request. A NACK could be sent in theory, if the ME is not permitted on the network (e.g. because it is on a blacklist).

Table 14: IMEI Check Result

Type	Description
0x01	ACK
0x02	NACK

23.7.30 Message Class

Indicate, which kind of message is being sent. This allows to trivially dispatch incoming GSUP messages to the right code paths, and should make writing a GSUP to MAP converter easier.

This IE was introduced together with inter-MSC handover code. Inter-MSC messages must include this IE and set it to the appropriate type. The intention of creating this IE was to use it with all GSUP messages eventually.

Type	Always present	Description
1	no	Subscriber Management
2	no	SMS
3	no	USSD
4	yes	Inter-MSC

23.7.31 Source Name

When the GSUP server is asked to forward a message between two GSUP clients, the source name is the IPA name of the client where the message is coming from. The source name IE is present, when the GSUP server forwards the message to the destination. Although redundant, the source name IE is also sent from the source to the GSUP server (so it is easier to follow the network traces).

Source and destination names are sent as nul-terminated strings.

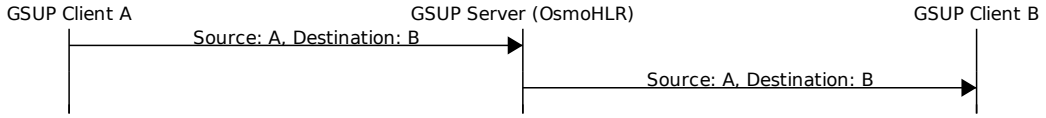


Figure 34: Message forwarding example

23.7.32 Destination Name

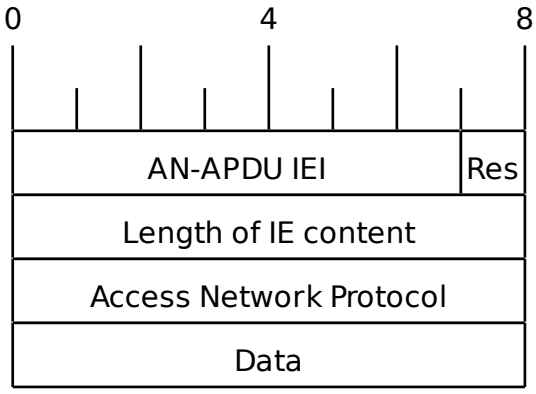
The receiving counterpart to source name (Section 23.7.31).

23.7.33 AN-APDU

This IE encodes the AN-APDU parameter described in 3GPP TS 29.002 7.6.9.1.

Table 15: Access Network Protocol

Type	Description
0x01	BSSAP
0x02	RANAP



23.7.34 RR Cause

This IE contains the reason for release or completion of an assignment or handover. See 3GPP TS 44.018 10.5.2.31 for reference.

23.7.35 BSSAP Cause

This IE indicates why an event is happening on the BSSAP interface. See 3GPP TS 48.008 3.2.2.5 for reference.

23.7.36 Current RAT Type

This IE contains the current RAT Type.

Table 16: Current RAT Type values

Type	Description
0x01	GERAN
0x02	UTRAN
0x03	EUTRAN

23.7.37 Session Management Cause

This IE contains the reason for rejecting a session management request. See 3GPP TS 24.008 10.5.6.6 / Table 10.5.157 for reference.

23.8 Session (transaction) management

Unlike TCAP/MAP, GSUP is just a transport layer without the dialogue/context. All communication is usually happening over a single connection. In order to fill this gap, there is a few optional IEs, which allow both communication sides to establish and terminate TCAP-like transactions over GSUP.

23.8.1 Session ID

This auxiliary IE shall be used together with Section 23.8.2. The purpose of this IE is to identify a particular transaction using the 4-byte unique identifier.

23.8.2 Session State

This auxiliary IE shall be used together with Section 23.8.1. The purpose of this IE is to indicate a state of a particular transacrion, i.e. initiate, continue or terminate it.

Table 17: Session state

State	TCAP alternative	Description
0x00	Undefined	Used when session management is not required
0x01	BEGIN	Used to initiate a new session
0x02	CONTINUE	Used to continue an existing session
0x03	END	Used to terminate an existing session

23.8.3 SM-RP-MR (Message Reference)

According to TS GSM 04.11, section 8.2.3, every single message on the SM-RL (SM Relay Layer) has a unique *message reference*, that is used to link an *RP-ACK* or *RP-ERROR* message to the associated (preceding) *RP-DATA* or *RP-SMMA* message transfer attempt.

In case of TCAP/MAP, this message reference is being mapped to the *Invoke ID*. But since GSUP has no *Invoke ID IE*, and it is not required for other applications (other than SMS), a special Section 23.8.3 is used to carry the message reference value 'as-is' (i.e. in range 0 through 255).

23.8.4 SM-RP-DA (Destination Address)

This IE represents the destination address used by the short message service relay sub-layer protocol. It can be one of the following:

- IMSI (see 3GPP TS 29.002, clause 7.6.2.1);
- MSISDN (see 3GPP TS 29.002, clause 7.6.2.17);
- service centre address (see 3GPP TS 29.002, clause 7.6.2.27).

Coding of this IE is described in Section 23.8.6. See 3GPP TS 29.002, section 7.6.8.1 for details.

23.8.5 SM-RP-OA (Originating Address)

This IE represents the originating address used by the short message service relay sub-layer protocol. It can be either of the following:

- MSISDN (see 3GPP TS 29.002, clause 7.6.2.17);
- service centre address (see 3GPP TS 29.002, clause 7.6.2.27).

Coding of this IE is described in Section 23.8.6. See 3GPP TS 29.002, section 7.6.8.2 for details.

23.8.6 Coding of SM-RP-DA / SM-RP-OA IEs

Basically, both Section 23.8.4 / Section 23.8.5 IEs contain a single TV of the following format:

Table 18: Coding of SM-RP-DA / SM-RP-OA IEs

Field	Presence	Length	Description
T	M	1	Identity type
V	O	1	ToN/NPI header
V	O	...	BCD encoded (or alphanumeric) identity

where the identity type can be one of the following:

Table 19: Identity types of SM-RP-DA / SM-RP-OA IEs

Type	ToN/NPI Header	Description
0x01	No	IMSI (see 3GPP TS 29.002, clause 7.6.2.1)
0x02	Yes	MSISDN (see 3GPP TS 29.002, clause 7.6.2.17)
0x03	Yes	Service centre address (see 3GPP TS 29.002, clause 7.6.2.27)
0xff	No	Omit value for noSM-RP-DA and noSM-RP-OA

Coding of the optional ToN/NPI header, as well as all possible ToN/NPI values, is described in 3GPP TS 129.002, section 17.7.8 "Common data types", and can be summarized as follows:

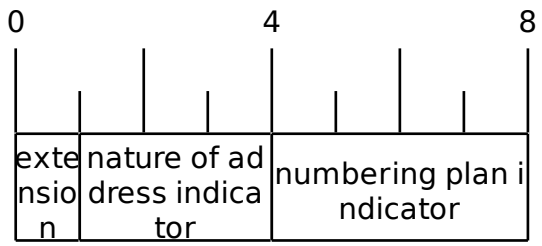


Figure 35: ToN/NPI header coding (as per 3GPP TS 129.002, MSB first)

Please note that unlike both Section 23.7.19 and Section 23.7.20, where the value part is encoded as LV (i.e. contains an additional length), an identity in both Section 23.8.4 / Section 23.8.5 IEs shall not contain the redundant length octet.

23.8.7 SM-RP-UI (SM TPDU)

This IE represents the user data field carried by the short message service relay sub-layer (i.e. SM-TL (Transfer Layer)) protocol. In case of errors (i.e. MO-/MT-forwardSM Error messages), this IE may contain optional diagnostic field payload from *RP-ERROR* message.

See 3GPP TS 29.002, section 7.6.8.4 for details.

23.8.8 SM-RP-Cause (RP Cause value)

According to TS GSM 04.11, *RP-Cause* is a variable length element always included in the *RP-ERROR* message, conveying a negative result of an *RP-DATA* message transfer attempt or *RP-SMMA* notification attempt.

The mapping between error causes in TS GSM 04.11 and TS GSM 09.02 (MAP) is specified in TS GSM 03.40. But since GSUP has no generic *User Error IE*, and it is not required for other applications (other than SMS), a special Section 23.8.8 is used to carry the cause value 'as-is'.

23.8.9 SM-RP-MMS (More Messages to Send)

This is an optional IE of MT-ForwardSM-Req message, that is used by SMSC to indicate that there are more MT SMS messages to be sent, so the network should keep the RAN connection open. See 3GPP TS 29.002, section 7.6.8.7.

23.8.10 SM Alert Reason

According to 3GPP TS 29.002, section 7.6.8.8, Alert Reason is used to indicate the reason why the service centre is alerted, e.g. the MS has got some memory to store previously rejected incoming SMS.

It can take one of the following values:

Table 20: SM Alert Reason values

Type	Description
0x01	MS present
0x02	Memory Available

24 VTY Process and Thread management

Most Osmocom programs provide, some support to tune some system settings related to the running process, its threads, its scheduling policies, etc.

All of these settings can be configured through the VTY, either during startup by means of usual config files or through direct human interaction at the telnet VTY interface while the process is running.

24.1 Scheduling Policy

The scheduler to use as well as some of its properties (such as realtime priority) can be configured at any time for the entire process. This sort of functionality is useful in order to increase priority for processes running time-constrained procedures, such as those acting on the Um interface, like *osmo-trx* or *osmo-bts*, where use of this feature is highly recommended.

Example: Set process to use RR scheduler

```
cpu-sched
 policy rr 1 ❶
```

❶ Configure process to use *SCHED_RR* policy with real time priority 1

24.2 CPU-Affinity Mask

Most operating systems allow for some sort of configuration on restricting the amount of CPUs a given process or thread can run on. The procedure is sometimes called as *cpu-pinning* since it allows to keep different processes pinned on a subset of CPUs to make sure the scheduler won't run two CPU-hungry processes on the same CPU.

The set of CPUs where each thread is allowed to run on is expressed by means of a bitmask in hexadecimal representation, where the right most bit relates to CPU 0, and the Nth most significant bit relates to CPU *N-1*. Setting the bit means the process is allowed to run on that CPU, while clearing it means the process is forbidden to run on that CPU.

Hence, for instance a cpu-affinity mask of *0x00* means the thread is not allowed on any CPU, which will cause the thread to stall until a new value is applied. A mask of *0x01* means the thread is only allowed to run on the 1st CPU (CPU 0). A mask of *0xff00* means CPUs 8-15 are allowed, while 0-7 are not.

For single-threaded processes (most of Osmocom are), it is usually enough to set this line in VTY config file as follows:

```
cpu-sched
 cpu-affinity self 0x01 ❶
```

❶ Allow main thread (the one managing the VTY) only on CPU 0

Or otherwise:

```
cpu-sched
 cpu-affinity all 0x01 ❶
```

❶ Allow all threads only on CPU 0

For multi-threaded processes, it may be desired to run some threads on a subset of CPUs while another subset may run on another one. In order to identify threads, one can either use the TID of the thread (each thread has its own PID in Linux), or its specific Thread Name in case it has been set by the application.

The related information on all threads available in the process can be listed through VTY. This allows identifying quickly the different threads, its current cpu-affinity mask, etc.

Example: Get osmo-trx Thread list information from VTY

```
OsmoTRX> show cpu-sched threads
Thread list for PID 338609:
 TID: 338609, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338610, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338611, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338629, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338630, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338631, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338634, NAME: 'UHDAsyncEvent', cpu-affinity: 0x3
 TID: 338635, NAME: 'TxLower', cpu-affinity: 0x3
 TID: 338636, NAME: 'RxLower', cpu-affinity: 0x3
 TID: 338637, NAME: 'RxUpper0', cpu-affinity: 0x3
 TID: 338638, NAME: 'TxUpper0', cpu-affinity: 0x3
 TID: 338639, NAME: 'RxUpper1', cpu-affinity: 0x3
 TID: 338640, NAME: 'TxUpper1', cpu-affinity: 0x3
```

At runtime, one can change the cpu-affinity mask for a given thread identifying it by either TID or name:

Example: Set CPU-affinity from VTY telnet interface

```
OsmoTRX> cpu-affinity TxLower 0x02 ❶
OsmoTRX> cpu-affinity TxLower 0x03 ❷
```

❶ Allow thread named *TxLower* (*338635*) only on CPU 1

❷ Allow with TID *338636* (*RxLower*) only on CPU 0 and 1

Since thread names are set dynamically by the process during startup or at a later point after creating the thread itself, One may need to specify in the config file that the mask must be applied by the thread itself once being configured rather than trying to apply it immediately. To specify so, the *delay* keyword is using when configuring in the VTY. If the *delay* keyword is not used, the VTY will report and error and fail at startup when trying to apply a cpu-affinity mask for a yet-to-be-created thread.

Example: Set CPU-affinity from VTY config file

```
cpu-sched
 cpu-affinity TxLower 0x01 delay ❶
```

❶ Allow thread named *TxLower* (*338635*) only on CPU 1. It will be applied by the thread itself when created.

25 Glossary

2FF
2nd Generation Form Factor; the so-called plug-in SIM form factor

3FF
3rd Generation Form Factor; the so-called microSIM form factor

3GPP
3rd Generation Partnership Project

4FF
4th Generation Form Factor; the so-called nanoSIM form factor

A Interface
Interface between BTS and BSC, traditionally over E1 (*3GPP TS 48.008* [3gpp-ts-48-008])

A3/A8
Algorithm 3 and 8; Authentication and key generation algorithm in GSM and GPRS, typically COMP128v1/v2/v3 or MILENAGE are typically used

A5
Algorithm 5; Air-interface encryption of GSM; currently only A5/0 (no encryption), A5/1 and A5/3 are in use

Abis Interface
Interface between BTS and BSC, traditionally over E1 (*3GPP TS 48.058* [3gpp-ts-48-058] and *3GPP TS 52.021* [3gpp-ts-52-021])

ACC
Access Control Class; every BTS broadcasts a bit-mask of permitted ACC, and only subscribers with a SIM of matching ACC are permitted to use that BTS

AGCH
Access Grant Channel on Um interface; used to assign a dedicated channel in response to RACH request

AGPL
GNU Affero General Public License, a copyleft-style Free Software License

AQPSK
Adaptive QPSK, a modulation scheme used by VAMOS channels on Downlink

ARFCN
Absolute Radio Frequency Channel Number; specifies a tuple of uplink and downlink frequencies

AUC
Authentication Center; central database of authentication key material for each subscriber

BCCH
Broadcast Control Channel on Um interface; used to broadcast information about Cell and its neighbors

BCC
Base Station Color Code; short identifier of BTS, lower part of BSIC

BTS
Base Transceiver Station

BSC
Base Station Controller

BSIC
Base Station Identity Code; 16bit identifier of BTS within location area

BSSGP
Base Station Subsystem Gateway Protocol (*3GPP TS 48.018* [3gpp-ts-48-018])

BVCI
BSSGP Virtual Circuit Identifier

CBC
Cell Broadcast Centre; central entity of Cell Broadcast service

CBCH
Cell Broadcast Channel; used to transmit Cell Broadcast SMS (SMS-CB)

CBS
Cell Broadcast Service

CBSP
Cell Broadcast Service Protocol (*3GPP TS 48.049* [3gpp-ts-48-049])

CC
Call Control; Part of the GSM Layer 3 Protocol

CCCH
Common Control Channel on Um interface; consists of RACH (uplink), BCCH, PCH, AGCH (all downlink)

Cell
A cell in a cellular network, served by a BTS

CEPT
Conférence européenne des administrations des postes et des télécommunications; European Conference of Postal and Telecommunications Administrations.

CGI
Cell Global Identifier comprised of MCC, MNC, LAC and BSIC

CSFB
Circiut-Switched Fall Back; Mechanism for switching from LTE/EUTRAN to UTRAN/GERAN when circuit-switched services such as voice telephony are required.

dB
deci-Bel; relative logarithmic unit

dBm
deci-Bel (milliwatt); unit of measurement for signal strength of radio signals

DHCP
Dynamic Host Configuration Protocol (*IETF RFC 2131* [ietf-rfc2131])

downlink
Direction of messages / signals from the network core towards the mobile phone

DSCP
Differentiated Services Code Point (*IETF RFC 2474* [ietf-rfc2474])

DSP
Digital Signal Processor

dvnixload
Tool to program UBL and the Bootloader on a sysmoBTS

EDGE
Enhanced Data rates for GPRS Evolution; Higher-speed improvement of GPRS; introduces 8PSK

EGPRS
Enhanced GPRS; the part of EDGE relating to GPRS services

EIR
Equipment Identity Register; core network element that stores and manages IMEI numbers

ESME
External SMS Entity; an external application interfacing with a SMSC over SMPP

ETSI
European Telecommunications Standardization Institute

FPGA
Field Programmable Gate Array; programmable digital logic hardware

Gb
Interface between PCU and SGSN in GPRS/EDGE network; uses NS, BSSGP, LLC

GERAN
GPRS/EDGE Radio Access Network

GFDL
GNU Free Documentation License; a copyleft-style Documentation License

GGSN
GPRS Gateway Support Node; gateway between GPRS and external (IP) network

GMSK
Gaussian Minimum Shift Keying; modulation used for GSM and GPRS

GPL
GNU General Public License, a copyleft-style Free Software License

Gp
Gp interface between SGSN and GGSN; uses GTP protocol

GPRS
General Packet Radio Service; the packet switched 2G technology

GPS
Global Positioning System; provides a highly accurate clock reference besides the global position

GSM
Global System for Mobile Communications. ETSI/3GPP Standard of a 2G digital cellular network

GSMTAP
GSM tap; pseudo standard for encapsulating GSM protocol layers over UDP/IP for analysis

GSUP
Generic Subscriber Update Protocol. Osmocom-specific alternative to TCAP/MAP

GT
Global Title; an address in SCCP

GTP
GPRS Tunnel Protocol; used between SGSN and GGSN

HLR
Home Location Register; central subscriber database of a GSM network

HNB-GW
Home NodeB Gateway. Entity between femtocells (Home NodeB) and CN in 3G/UMTS.

HPLMN
Home PLMN; the network that has issued the subscriber SIM and has his record in HLR

IE
Information Element

IMEI
International Mobile Equipment Identity; unique 14-digit decimal number to globally identify a mobile device, optionally with a 15th checksum digit

IMEISV
IMEI software version; unique 14-digit decimal number to globally identify a mobile device (same as IMEI) plus two software version digits (total digits: 16)

IMSI
International Mobile Subscriber Identity; 15-digit unique identifier for the subscriber/SIM; starts with MCC/MNC of issuing operator

IP
Internet Protocol (*IETF RFC 791* [ietf-rfc791])

IPA
ip.access GSM over IP protocol; used to multiplex a single TCP connection

Iu
Interface in 3G/UMTS between RAN and CN

IuCS
Iu interface for circuit-switched domain. Used in 3G/UMTS between RAN and MSC

IuPS
Iu interface for packet-switched domain. Used in 3G/UMTS between RAN and SGSN

LAC
Location Area Code; 16bit identifier of Location Area within network

LAPD
Link Access Protocol, D-Channel (*ITU-T Q.921* [itu-t-q921])

LAPDm
Link Access Protocol Mobile (*3GPP TS 44.006* [3gpp-ts-44-006])

LLC
Logical Link Control; GPRS protocol between MS and SGSN (*3GPP TS 44.064* [3gpp-ts-44-064])

Location Area
Location Area; a geographic area containing multiple BTS

LU
Location Updating; can be of type IMSI-Attach or Periodic. Procedure that indicates a subscriber's physical presence in a given radio cell.

M2PA
MTP2 Peer-to-Peer Adaptation; a SIGTRAN Variant (*RFC 4165* [ietf-rfc4165])

M2UA
MTP2 User Adaptation; a SIGTRAN Variant (*RFC 3331* [ietf-rfc3331])

M3UA
MTP3 User Adaptation; a SIGTRAN Variant (*RFC 4666* [ietf-rfc4666])

MCC
Mobile Country Code; unique identifier of a country, e.g. 262 for Germany

MFF
Machine-to-Machine Form Factor; a SIM chip package that is soldered permanently onto M2M device circuit boards.

MGW
Media Gateway

MM
Mobility Management; part of the GSM Layer 3 Protocol

MNC
Mobile Network Code; identifies network within a country; assigned by national regulator

MNCC
Mobile Network Call Control; Unix domain socket based Interface between MSC and external call control entity like osmo-sip-connector

MNO
Mobile Network Operator; operator with physical radio network under his MCC/MNC

MO
Mobile Originated. Direction from Mobile (MS/UE) to Network

MS
Mobile Station; a mobile phone / GSM Modem

MSC
Mobile Switching Center; network element in the circuit-switched core network

MSC pool
A number of redundant MSCs serving the same core network, which a BSC / RNC distributes load across; see also the "MSC Pooling" chapter in OsmoBSC's user manual [userman-osmobsc] and *3GPP TS 23.236* [3gpp-ts-23-236]

MSISDN
Mobile Subscriber ISDN Number; telephone number of the subscriber

MT
Mobile Terminated. Direction from Network to Mobile (MS/UE)

MTP
Message Transfer Part; SS7 signaling protocol (*ITU-T Q.701* [itu-t-q701])

MVNO
Mobile Virtual Network Operator; Operator without physical radio network

NCC
Network Color Code; assigned by national regulator

NITB
Network In The Box; combines functionality traditionally provided by BSC, MSC, VLR, HLR, SMSC functions; see OsmoNITB

NRI
Network Resource Indicator, typically 10 bits of a TMSI indicating which MSC of an MSC pool attached the subscriber; see also the "MSC Pooling" chapter in OsmoBSC's user manual [userman-osmobsc] and *3GPP TS 23.236* [3gpp-ts-23-236]

NSEI
NS Entity Identifier

NVCI
NS Virtual Circuit Identifier

NWL
Network Listen; ability of some BTS to receive downlink from other BTSs

NS
Network Service; protocol on Gb interface (*3GPP TS 48.016* [3gpp-ts-48-016])

OCXO
Oven Controlled Crystal Oscillator; very high precision oscillator, superior to a VCTCXO

OML
Operation & Maintenance Link (ETSI/*3GPP TS 52.021* [3gpp-ts-52-021])

OpenBSC
Open Source implementation of GSM network elements, specifically OsmoBSC, OsmoNITB, OsmoSGSN

OpenGGSN
Open Source implementation of a GPRS Packet Control Unit

OpenVPN
Open-Source Virtual Private Network; software employed to establish encrypted private networks over untrusted public networks

Osmocom
Open Source MObile COMmunications; collaborative community for implementing communications protocols and systems, including GSM, GPRS, TETRA, DECT, GMR and others

OsmoBSC
Open Source implementation of a GSM Base Station Controller

OsmoNITB
Open Source implementation of a GSM Network In The Box, combines functionality traditionally provided by BSC, MSC, VLR, HLR, AUC, SMSC

OsmoSGSN
Open Source implementation of a Serving GPRS Support Node

OsmoPCU
Open Source implementation of a GPRS Packet Control Unit

OTA
Over-The-Air; Capability of operators to remotely reconfigure/reprogram ISM/USIM cards

PC
Point Code; an address in MTP

PCH
Paging Channel on downlink Um interface; used by network to page an MS

PCP
Priority Code Point (*IEEE 802.1Q* [?])

PCU
Packet Control Unit; used to manage Layer 2 of the GPRS radio interface

PDCH
Packet Data Channel on Um interface; used for GPRS/EDGE signalling + user data

PIN
Personal Identification Number; a number by which the user authenticates to a SIM/USIM or other smart card

PLMN
Public Land Mobile Network; specification language for a single GSM network

PUK
PIN Unblocking Code; used to unblock a blocked PIN (after too many wrong PIN attempts)

RAC
Routing Area Code; 16bit identifier for a Routing Area within a Location Area

RACH
Random Access Channel on uplink Um interface; used by MS to request establishment of a dedicated channel

RAM
Remote Application Management; Ability to remotely manage (install, remove) Java Applications on SIM/USIM Card

RF
Radio Frequency

RFM
Remote File Management; Ability to remotely manage (write, read) files on a SIM/USIM card

Roaming
Procedure in which a subscriber of one network is using the radio network of another network, often in different countries; in some countries national roaming exists

Routing Area
Routing Area; GPRS specific sub-division of Location Area

RR
Radio Resources; Part of the GSM Layer 3 Protocol

RSL
Radio Signalling Link (*3GPP TS 48.058* [3gpp-ts-48-058])

RTP
Real-Time Transport Protocol (*IETF RFC 3550* [ietf-rfc3550]); Used to transport audio/video streams over UDP/IP

SACCH
Slow Associate Control Channel on Um interface; bundled to a TCH or SDCCH, used for signalling in parallel to active dedicated channel

SCCP
Signaling Connection Control Part; SS7 signaling protocol (*ITU-T Q.711* [itu-t-q711])

SDCCH
Slow Dedicated Control Channel on Um interface; used for signalling and SMS transport in GSM

SDK
Software Development Kit

SGs
Interface between MSC (GSM/UMTS) and MME (LTE/EPC) to facilitate CSFB and SMS.

SGSN
Serving GPRS Support Node; Core network element for packet-switched services in GSM and UMTS.

SIGTRAN
Signaling Transport over IP (*IETF RFC 2719* [ietf-rfc2719])

SIM
Subscriber Identity Module; small chip card storing subscriber identity

Site
A site is a location where one or more BTSs are installed, typically three BTSs for three sectors

SMPP
Short Message Peer-to-Peer; TCP based protocol to interface external entities with an SMSC

SMSC
Short Message Service Center; store-and-forward relay for short messages

SS7
Signaling System No. 7; Classic digital telephony signaling system

SS
Supplementary Services; query and set various service parameters between subscriber and core network (e.g. USSD, 3rd-party calls, hold/retrieve, advice-of-charge, call deflection)

SSH
Secure Shell; *IETF RFC 4250* [ietf-rfc4251] to 4254

SSN
Sub-System Number; identifies a given SCCP Service such as MSC, HLR

STP
Signaling Transfer Point; A Router in SS7 Networks

SUA
SCCP User Adaptation; a SIGTRAN Variant (*RFC 3868* [ietf-rfc3868])

syslog
System logging service of UNIX-like operating systems

System Information
A set of downlink messages on the BCCH and SACCH of the Um interface describing properties of the cell and network

TCH
Traffic Channel; used for circuit-switched user traffic (mostly voice) in GSM

TCP
Transmission Control Protocol; (*IETF RFC 793* [ietf-rfc793])

TFTP
Trivial File Transfer Protocol; (*IETF RFC 1350* [ietf-rfc1350])

TOS
Type Of Service; bit-field in IPv4 header, now re-used as DSCP (*IETF RFC 791* [ietf-rfc791])

TRX
Transceiver; element of a BTS serving a single carrier

TS
Technical Specification

u-Boot
Boot loader used in various embedded systems

UBI
An MTD wear leveling system to deal with NAND flash in Linux

UBL
Initial bootloader loaded by the TI Davinci SoC

UDP
User Datagram Protocol (*IETF RFC 768* [ietf-rfc768])

UICC
Universal Integrated Chip Card; A smart card according to *ETSI TR 102 216* [etsi-tr102216]

Um interface
U mobile; Radio interface between MS and BTS

uplink
Direction of messages: Signals from the mobile phone towards the network

USIM
Universal Subscriber Identity Module; application running on a UICC to provide subscriber identity for UMTS and GSM networks

USSD
Unstructured Supplementary Service Data; textual dialog between subscriber and core network, e.g. **100 → Your extension is 1234*

VAMOS
Voice services over Adaptive Multi-user channels on One Slot; an optional extension for GSM specified in Release 9 of 3GPP GERAN specifications (*3GPP TS 48.018* [3gpp-ts-48-018]) allowing two independent UEs to transmit and receive simultaneously on traffic channels

VCTCXO
Voltage Controlled, Temperature Compensated Crystal Oscillator; a precision oscillator, superior to a classic crystal oscillator, but inferior to an OCXO

VLAN
Virtual LAN in the context of Ethernet (*IEEE 802.1Q* [ieee-802.1q])

VLR
Visitor Location Register; volatile storage of attached subscribers in the MSC

VPLMN
Visited PLMN; the network in which the subscriber is currently registered; may differ from HPLMN when on roaming

VTY
Virtual TeletYpe; a textual command-line interface for configuration and introspection, e.g. the OsmoBSC configuration file as well as its telnet link on port 4242

A Osmocom TCP/UDP Port Numbers

The Osmocom GSM system utilizes a variety of TCP/IP based protocols. The table below provides a reference as to which port numbers are used by which protocol / interface.

Table 21: TCP/UDP port numbers

L4 Protocol	Port Number	Purpose	Software
UDP	1984	Osmux	osmo-mgw, osmo-bts
UDP	2427	MGCP GW	osmo-bsc_mgcp, osmo-mgw
TCP	2775	SMPP (SMS interface for external programs)	osmo-nitb
TCP	3002	A-bis/IP OML	osmo-bts, osmo-bsc, osmo-nitb
TCP	3003	A-bis/IP RSL	osmo-bts, osmo-bsc, osmo-nitb
TCP	4227	telnet (VTY)	osmo-pcap-client
TCP	4228	telnet (VTY)	osmo-pcap-server
TCP	4236	Control Interface	osmo-trx

Table 21: (continued)

L4 Protocol	Port Number	Purpose	Software
TCP	4237	telnet (VTY)	osmo-trx
TCP	4238	Control Interface	osmo-bts
TCP	4239	telnet (VTY)	osmo-stp
TCP	4240	telnet (VTY)	osmo-pcu
TCP	4241	telnet (VTY)	osmo-bts
TCP	4242	telnet (VTY)	osmo-nitb, osmo-bsc, cellmgr-ng
TCP	4243	telnet (VTY)	osmo-bsc_mgcp, osmo-mgw
TCP	4244	telnet (VTY)	osmo-bsc_nat
TCP	4245	telnet (VTY)	osmo-sgsn
TCP	4246	telnet (VTY)	osmo-gbproxy
TCP	4247	telnet (VTY)	OsmocomBB
TCP	4249	Control Interface	osmo-nitb, osmo-bsc
TCP	4250	Control Interface	osmo-bsc_nat
TCP	4251	Control Interface	osmo-sgsn
TCP	4252	telnet (VTY)	sysmobts-mgr
TCP	4253	telnet (VTY)	osmo-gtphub
TCP	4254	telnet (VTY)	osmo-msc
TCP	4255	Control Interface	osmo-msc
TCP	4256	telnet (VTY)	osmo-sip-connector
TCP	4257	Control Interface	osmo-ggsn, ggsn (OpenGGSN)
TCP	4258	telnet (VTY)	osmo-hlr
TCP	4259	Control Interface	osmo-hlr
TCP	4260	telnet (VTY)	osmo-ggsn
TCP	4261	telnet (VTY)	osmo-hnbgw
TCP	4262	Control Interface	osmo-hnbgw
TCP	4263	Control Interface	osmo-gbproxy
TCP	4264	telnet (VTY)	osmo-cbc
TCP	4265	Control Interface	osmo-cbc
TCP	4266	D-GSM MS Lookup: mDNS serve	osmo-hlr
TCP	4267	Control Interface	osmo-mgw
TCP	4268	telnet (VTY)	osmo-uecups
SCTP	4268	UECUPS	osmo-uecups
TCP	4269	telnet (VTY)	osmo-e1d
TCP	4270	telnet (VTY)	osmo-isdntap
TCP	4271	telnet (VTY)	osmo-smlc
TCP	4272	Control Interface	osmo-smlc
TCP	4273	telnet (VTY)	osmo-hnodeb
TCP	4274	Control Interface	osmo-hnodeb
TCP	4275	telnet (VTY)	osmo-upf
TCP	4276	Control Interface	osmo-upf
TCP	4277	telnet (VTY)	osmo-pfcp-tool
TCP	4278	Control Interface	osmo-pfcp-tool
UDP	4729	GSMTAP	Almost every osmocom project
TCP	5000	A/IP	osmo-bsc, osmo-bsc_nat
UDP	23000	GPRS-NS over IP default port	osmo-pcu, osmo-sgsn, osmo-gbproxy
TCP	48049	BSC-CBC (CBSP) default port	osmo-bsc, osmo-cbc

B Bibliography / References

B.0.0.0.1 References

[1] [userman-ice1usb] Osmocom Project: icE1usb User Manual.

[2] [userman-ogt] Pau Espin: osmo-gsm-tester User Manual.

[3] [userman-remsim] Harald Welte: osmo-remsim User Manual.

[4] [osmobts-abis-spec] Neels Hofmeyr & Harald Welte. OsmoBTS Abis Protocol Specification. https://ftp.osmocom.org/docs/latest/osmobts-abis.pdf

[5] [userman-osmobsc] Osmocom Project: OsmoBSC User Manual. https://ftp.osmocom.org/docs/latest/osmobsc-usermanual.pdf

[6] [vty-ref-osmobsc] Osmocom Project: OsmoBSC VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmobsc-vty-reference.pdf

[7] [userman-osmobts] Osmocom Project: OsmoBTS User Manual. https://ftp.osmocom.org/docs/latest/osmobts-usermanual.pdf

[8] [vty-ref-osmobts] Osmocom Project: OsmoBTS VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmobts-trx-vty-reference.pdf https://ftp.osmocom.org/docs/latest/osmobts-sysmo-vty-reference.pdf https://ftp.osmocom.org/docs/latest/osmobts-lc15-vty-reference.pdf https://ftp.osmocom.org/docs/latest/-osmobts-oc2g-vty-reference.pdf https://ftp.osmocom.org/docs/latest/osmobts-octphy-vty-reference.pdf https://ftp.osmocom.org/docs/latest/osmobts-virtual-vty-reference.pdf

[9] [userman-osmocbc] Osmocom Project: OsmoCBC User Manual. https://ftp.osmocom.org/docs/latest/-osmocbc-usermanual.pdf

[10] [vty-ref-osmocbc] Osmocom Project: OsmoCBC VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmocbc-vty-reference.pdf

[11] [userman-osmogbproxy] Osmocom Project: OsmoGBProxy User Manual. https://ftp.osmocom.org/docs/-latest/osmogbproxy-usermanual.pdf

[12] [vty-ref-osmogbproxy] Osmocom Project: OsmoGBPRoxy VTY Reference Manual. https://ftp.osmocom.org/-docs/latest/osmogbproxy-vty-reference.pdf

[13] [userman-osmoggsn] Osmocom Project: OpenGGSN User Manual. https://ftp.osmocom.org/docs/latest/-osmoggsn-usermanual.pdf

[14] [vty-ref-osmoggsn] Osmocom Project: OsmoGGSN VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmoggsn-vty-reference.pdf

[15] [userman-osmohlr] Osmocom Project: OsmoHLR User Manual. https://ftp.osmocom.org/docs/latest/osmohlr-usermanual.pdf

[16] [vty-ref-osmohlr] Osmocom Project: OsmoHLR VTY Reference Manual. https://ftp.osmocom.org/docs/latest/-osmohlr-vty-reference.pdf

[17] [userman-osmohnbgw] Osmocom Project: OsmoHNBGW User Manual. https://ftp.osmocom.org/docs/latest/-osmohnbgw-usermanual.pdf

[18] [vty-ref-osmohnbgw] Osmocom Project: OsmoHNBGW VTY Reference Manual. https://ftp.osmocom.org/-docs/latest/osmohnbgw-vty-reference.pdf

[19] [userman-osmomgw] Osmocom Project: OsmoMGW User Manual. https://ftp.osmocom.org/docs/latest/-osmomgw-usermanual.pdf

[20] [vty-ref-osmomgw] Osmocom Project: OsmoMGW VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmomgw-vty-reference.pdf

[21] [userman-osmomsc] Osmocom Project: OsmoMSC User Manual. https://ftp.osmocom.org/docs/latest/-osmomsc-usermanual.pdf

[22] [vty-ref-osmomsc] Osmocom Project: OsmoMSC VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmomsc-vty-reference.pdf

[23] [userman-osmonitb] Osmocom Project: OsmoNITB User Manual. https://ftp.osmocom.org/docs/latest/-osmonitb-usermanual.pdf

[24] [vty-ref-osmonitb] Osmocom Project: OsmoNITB VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmonitb-vty-reference.pdf

[25] [userman-osmopcu] Osmocom Project: OsmoPCU User Manual. https://ftp.osmocom.org/docs/latest/-osmopcu-usermanual.pdf

[26] [vty-ref-osmopcu] Osmocom Project: OsmoPCU VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmopcu-vty-reference.pdf

[27] [userman-osmosgsn] Osmocom Project: OsmoSGSN User Manual. https://ftp.osmocom.org/docs/latest/-osmosgsn-usermanual.pdf

[28] [vty-ref-osmosgsn] Osmocom Project: OsmoSGSN VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmosgsn-vty-reference.pdf

[29] [userman-osmosipconnector] Osmocom Project: OsmoSIPconnector User Manual. https://ftp.osmocom.org/-docs/latest/osmosipconnector-usermanual.pdf

[30] [vty-ref-osmosipconnector] Osmocom Project: OsmoSIPconnector VTY Reference Manual. https://ftp.osmocom.org/docs/latest/osmosipconnector-vty-reference.pdf

[31] [userman-osmosmlc] Osmocom Project: OsmoSMLC User Manual. https://ftp.osmocom.org/docs/latest/-osmosmlc-usermanual.pdf

[32] [vty-ref-osmosmlc] Osmocom Project: OsmoSMLC VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmosmlc-vty-reference.pdf

[33] [userman-osmostp] Osmocom Project: OsmoSTP User Manual. https://ftp.osmocom.org/docs/latest/osmostp-usermanual.pdf

[34] [vty-ref-osmostp] Osmocom Project: OsmoSTP VTY Reference Manual. https://ftp.osmocom.org/docs/latest/-osmostp-vty-reference.pdf

[35] [userman-osmotrx] Osmocom Project: OsmoTRX User Manual. https://ftp.osmocom.org/docs/latest/osmotrx-usermanual.pdf

[36] [vty-ref-osmotrx] Osmocom Project: OsmoTRX VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmotrx-uhd-vty-reference.pdf https://ftp.osmocom.org/docs/latest/osmotrx-lms-vty-reference.pdf https://ftp.osmocom.org/docs/latest/osmotrx-ipc-vty-reference.pdf https://ftp.osmocom.org/docs/latest/-osmotrx-usrp1-vty-reference.pdf

[37] [3gpp-ts-23-041] 3GPP TS 23.041: Technical realization of Cell Broadcast Service (CBS)

[38] [3gpp-ts-23-048] 3GPP TS 23.048: Security mechanisms for the (U)SIM application toolkit; Stage 2 https://www.3gpp.org/DynaReport/23048.htm

[39] [3gpp-ts-23-236] 3GPP TS 23.236: Intra-domain connection of Radio Access Network (RAN) nodes to multiple Core Network (CN) nodes https://www.3gpp.org/DynaReport/23236.htm

[40] [3gpp-ts-24-007] 3GPP TS 24.007: Mobile radio interface signalling layer 3; General Aspects https://www.3gpp.org/DynaReport/24007.htm

[41] [3gpp-ts-24-008] 3GPP TS 24.008: Mobile radio interface Layer 3 specification; Core network protocols; Stage 3. https://www.3gpp.org/dynareport/24008.htm

[42] [3gpp-ts-31-101] 3GPP TS 31.101: UICC-terminal interface; Physical and logical characteristics https://www.3gpp.org/DynaReport/31101.htm

[43] [3gpp-ts-31-102] 3GPP TS 31.102: Characteristics of the Universal Subscriber Identity Module (USIM) application https://www.3gpp.org/DynaReport/31102.htm

[44] [3gpp-ts-31-103] 3GPP TS 31.103: Characteristics of the IMS Subscriber Identity Module (ISIM) application https://www.3gpp.org/DynaReport/31103.htm

[45] [3gpp-ts-31-111] 3GPP TS 31.111: Universal Subscriber Identity Module (USIM) Application Toolkit (USAT) https://www.3gpp.org/DynaReport/31111.htm

[46] [3gpp-ts-31-115] 3GPP TS 31.115: Secured packet structure for (Universal) Subscriber Identity Module (U)SIM Toolkit applications https://www.3gpp.org/DynaReport/31115.htm

[47] [3gpp-ts-31-116] 3GPP TS 31.116: Remote APDU Structure for (U)SIM Toolkit applications https://www.3gpp.org/DynaReport/31116.htm

[48] [3gpp-ts-35-205] 3GPP TS 35.205: 3G Security; Specification of the MILENAGE algorithm set: General

[49] [3gpp-ts-35-206] 3GPP TS 35.206: 3G Security; Specification of the MILENAGE algorithm set: Algorithm specification https://www.3gpp.org/DynaReport/35206.htm

[50] [3gpp-ts-44-006] 3GPP TS 44.006: Mobile Station - Base Station System (MS - BSS) interface; Data Link (DL) layer specification https://www.3gpp.org/DynaReport/44006.htm

[51] [3gpp-ts-44-018] 3GPP TS 44.018: Mobile radio interface layer 3 specification; Radio Resource Control (RRC) protocol https://www.3gpp.org/DynaReport/44018.htm

[52] [3gpp-ts-44-064] 3GPP TS 44.064: Mobile Station - Serving GPRS Support Node (MS-SGSN); Logical Link Control (LLC) Layer Specification https://www.3gpp.org/DynaReport/44064.htm

[53] [3gpp-ts-45-002] 3GPP TS 45.002: Digital cellular telecommunications system (Phase 2+) (GSM); GSM/EDGE Multiplexing and multiple access on the radio path https://www.3gpp.org/DynaReport/45002.htm

[54] [3gpp-ts-48-008] 3GPP TS 48.008: Mobile Switching Centre - Base Station system (MSC-BSS) interface; Layer 3 specification https://www.3gpp.org/DynaReport/48008.htm

[55] [3gpp-ts-48-016] 3GPP TS 48.016: General Packet Radio Service (GPRS); Base Station System (BSS) - Serving GPRS Support Node (SGSN) interface; Network service https://www.3gpp.org/DynaReport/48016.htm

[56] [3gpp-ts-48-018] 3GPP TS 48.018: General Packet Radio Service (GPRS); Base Station System (BSS) - Serving GPRS Support Node (SGSN); BSS GPRS protocol (BSSGP) https://www.3gpp.org/DynaReport/48018.htm

[57] [3gpp-ts-48-049] 3GPP TS 48.049: Digital cellular communications system; Base Station Controller - Cell Broadcast Centre (BSC-CBC) interface specification; Cell Broadcast Service Protocol (CBSP) https://www.3gpp.org/DynaReport/48049.htm

[58] [3gpp-ts-48-056] 3GPP TS 48.056: Base Station Controller - Base Transceiver Station (BSC - BTS) interface; Layer 2 specification https://www.3gpp.org/DynaReport/48056.htm

[59] [3gpp-ts-48-058] 3GPP TS 48.058: Base Station Controller - Base Transceiver Station (BSC - BTS) Interface; Layer 3 specification https://www.3gpp.org/DynaReport/48058.htm

[60] [3gpp-ts-51-011] 3GPP TS 51.011: Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface

[61] [3gpp-ts-51-014] 3GPP TS 51.014: Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface https://www.3gpp.org/DynaReport/51014.htm

[62] [3gpp-ts-52-021] 3GPP TS 52.021: Network Management (NM) procedures and messages on the A-bis interface https://www.3gpp.org/DynaReport/52021.htm

[63] [etsi-tr102216] ETSI TR 102 216: Smart cards https://www.etsi.org/deliver/etsi_tr/102200_102299/102216/-03.00.00_60/tr_102216v030000p.pdf

[64] [etsi-ts102221] ETSI TS 102 221: Smart Cards; UICC-Terminal interface; Physical and logical characteristics https://www.etsi.org/deliver/etsi_ts/102200_102299/102221/13.01.00_60/ts_102221v130100p.pdf

[65] [etsi-ts101220] ETSI TS 101 220: Smart Cards; ETSI numbering system for telecommunication application providers https://www.etsi.org/deliver/etsi_ts/101200_101299/101220/12.00.00_60/ts_101220v120000p.pdf

[66] [etsi-ts102671] ETSI TS 102 671: Smart Cards; Machine to Machine UICC; Physical and logical characteristics https://www.etsi.org/deliver/etsi_ts/102600_102699/102671/18.01.00_60/ts_102671v180100p.pdf

[67] [ieee-802.1q] IEEE 802.1Q: Bridges and Bridged Networks https://ieeexplore.ieee.org/document/6991462

[68] [ietf-rfc768] IETF RFC 768: User Datagram Protocol https://tools.ietf.org/html/rfc768

[69] [ietf-rfc791] IETF RFC 791: Internet Protocol https://tools.ietf.org/html/rfc791

[70] [ietf-rfc793] IETF RFC 793: Transmission Control Protocol https://tools.ietf.org/html/rfc793

[71] [ietf-rfc1035] IETF RFC 1035: Domain Names - Implementation and Specification https://tools.ietf.org/html/-rfc1035

[72] [ietf-rfc1350] IETF RFC 1350: Trivial File Transfer Protocol https://tools.ietf.org/html/rfc1350

[73] [ietf-rfc2131] IETF RFC 2131: Dynamic Host Configuration Protocol https://tools.ietf.org/html/rfc2131

[74] [ietf-rfc2474] IETF RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv44 and IPv6 Headers https://tools.ietf.org/html/rfc2474

[75] [ietf-rfc2719] IETF RFC 2719: Signal Transport over IP https://tools.ietf.org/html/rfc2719

[76] [ietf-rfc3331] IETF RFC 3331: Message Transfer Part 2 User Adaptation Layer https://tools.ietf.org/html/-rfc3331

[77] [ietf-rfc3550] IETF RFC 3550: RTP: A Transport protocol for Real-Time Applications https://tools.ietf.org/-html/rfc3550

[78] [ietf-rfc3596] IETF RFC 3596: DNS Extensions to Support IP Version 6 https://tools.ietf.org/html/rfc3596

[79] [ietf-rfc3868] IETF RFC 3868: SCCP User Adaptation Layer https://tools.ietf.org/html/rfc3868

[80] [ietf-rfc4165] IETF RFC 4165: Message Transfer Part 2 Peer-to-Peer Adaptation Layer https://tools.ietf.org/-html/rfc4165

[81] [ietf-rfc4251] IETF RFC 4251: The Secure Shell (SSH) Protocol Architecture https://tools.ietf.org/html/-rfc4251

[82] [ietf-rfc4666] IETF RFC 4666: Message Transfer Part 3 User Adaptation Layer https://tools.ietf.org/html/-rfc4666

[83] [ietf-rfc5771] IETF RFC 5771: IANA Guidelines for IPv4 Multicast Address Assignments https://tools.ietf.org/html/rfc5771

[84] [itu-t-q701] ITU-T Q.701: Functional Description of the Message Transfer Part (MTP) https://www.itu.int/rec/-T-REC-Q.701/en/

[85] [itu-t-q711] ITU-T Q.711: Functional Description of the Signalling Connection Control Part https://www.itu.int/rec/T-REC-Q.711/en/

[86] [itu-t-q713] ITU-T Q.713: Signalling connection control part formats and codes https://www.itu.int/rec/T-REC-Q.713/en/

[87] [itu-t-q714] ITU-T Q.714: Signalling connection control part procedures https://www.itu.int/rec/T-REC-Q.714/en/

[88] [itu-t-q921] ITU-T Q.921: ISDN user-network interface - Data link layer specification https://www.itu.int/rec/-T-REC-Q.921/en

[89] [smpp-34] SMPP Develoepers Forum. Short Message Peer-to-Peer Protocol Specification v3.4 https://docs.nimta.com/SMPP_v3_4_Issue1_2.pdf

[90] [gnu-agplv3] Free Software Foundation. GNU Affero General Public License. https://www.gnu.org/licenses/-agpl-3.0.en.html

[91] [freeswitch_pbx] FreeSWITCH SIP PBX https://freeswitch.org

[92] [tw-ts-001] TW-TS-001: Enhanced RTP transport of FR and EFR codec frames in an IP-based GSM RAN https://www.freecalypso.org/specs/tw-ts-001-v010100.txt

[93] [tw-ts-002] TW-TS-002: Enhanced RTP transport of HRv1 codec frames in an IP-based GSM RAN https://www.freecalypso.org/specs/tw-ts-002-v010100.txt

[94] [tw-ts-003] TW-TS-003: BSSMAP extension for selection of enhanced RTP transport formats https://www.freecalypso.org/specs/tw-ts-003-v010002.txt

C GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. http://fsf.org/

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

C.1 PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

C.2 APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Section whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then

it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not Transparent is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, Title Page means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

C.3 VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section Section C.4.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

C.4 COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

C.5 MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

a. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

b. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

c. State on the Title Page the name of the publisher of the Modified Version, as the publisher.

d. Preserve all the copyright notices of the Document.

e. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

f. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

g. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

h. Include an unaltered copy of this License.

i. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

j. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

k. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

l. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

m. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

n. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Sections.

o. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Section and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

C.6 COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

C.7 COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

C.8 AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

C.9 TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

C.10 TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

C.11 FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

C.12 RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

C.13 ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with... Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.