<table>
<thead>
<tr>
<th>NUMBER</th>
<th>DATE</th>
<th>DESCRIPTION</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>July 24th, 2018</td>
<td>Initial version</td>
<td>DW</td>
</tr>
</tbody>
</table>
# 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 ....................... 2
   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 .................................. 3
      2.4.5 Documentation License .................... 4

3 Introduction ....................... 4
   3.1 Required Skills ......................... 4
   3.2 Getting assistance ....................... 4

4 Overview ........................... 4
   4.1 About OsmoMGW ......................... 5
   4.2 Software Components ...................... 5
      4.2.1 MGCP Implementation .................. 5
      4.2.2 RTP implementation .................... 6
      4.2.3 Audio transcoder ...................... 6
   4.3 Limitations ............................. 6
   4.4 Additional resources .................... 6

5 Running OsmoMGW .................. 6
   5.1 SYNOPSIS .................................. 6
   5.2 OPTIONS .................................... 6

6 The Osmocom VTY Interface ........ 7
   6.1 Accessing the telnet VTY ................. 8
   6.2 VTY Nodes .................................. 8
   6.3 Interactive help .......................... 9
      6.3.1 The question-mark (?) command ........ 9
      6.3.2 TAB completion ......................... 10
      6.3.3 The list command ....................... 10
      6.3.4 The attribute system .................... 12
      6.3.5 The expert mode ....................... 13
7 libosmocore Logging System
  7.1 Log categories .................................................. 14
  7.2 Log levels .......................................................... 15
  7.3 Log printing options .............................................. 15
  7.4 Log filters .......................................................... 16
  7.5 Log targets .......................................................... 16
    7.5.1 Logging to the VTY ............................................. 16
    7.5.2 Logging to the ring buffer ................................... 16
    7.5.3 Logging via gsmtap ............................................. 17
    7.5.4 Logging to a file .............................................. 17
    7.5.5 Logging to syslog ............................................. 18
    7.5.6 Logging to systemd-journal ................................... 18
    7.5.7 Logging to stderr ............................................. 20

8 Configuring OsmoMGW
  8.1 Configuring MGCP .................................................. 20
  8.2 Configuring the trunk ............................................ 20
  8.3 E1 trunk considerations .......................................... 21

9 MGCP Endpoints
  9.1 RTP proxy / RTP bridge endpoints ................................ 22
  9.2 E1/T1 endpoints .................................................... 23

10 MGCP Extensions
  10.1 X-Osmo-IGN ....................................................... 24
    10.1.1 X-Osmo-IGN Format .......................................... 24
    10.1.2 Supported X-Osmo-IGN Items ................................. 24
      10.1.2.1 X-Osmo-IGN: C ............................................ 24
  10.2 X-Osmux ............................................................ 24

11 Osmux
  11.1 Osmux and NAT .................................................... 25
  11.2 CID allocation .................................................... 25
  11.3 3GPP AoIP network setup with Osmux ............................ 26
  11.4 SCCPLite network setup with Osmux ............................. 28
  11.5 SCCPLite network setup with Osmux + BSC-NAT .................. 30
  11.6 Osmux and MGCP ................................................... 32
    11.6.1 X-Osmux Format .............................................. 32
    11.6.2 X-Osmux Considerations ....................................... 32
    11.6.3 X-Osmux Support ............................................. 33

Copyright © 2018 sysmocom - s.f.m.c. GmbH DRAFT 1.9.0-18-ge0058, 2021-Dec-25
12 QoS, DSCP/TOS, Priority and IEEE 802.1q PCP

12.1 IP Level (DSCP) .......................................................... 33
12.2 Packet Priority .............................................................. 34
12.3 Ethernet Level (PCP) ...................................................... 34
12.4 Putting things together .................................................... 36
  12.4.1 Full example of QoS for osmo-mgw ............................... 36

13 VTY Process and Thread management ................................... 37

13.1 Scheduling Policy .......................................................... 37
13.2 CPU-Affinity Mask ......................................................... 37

14 Glossary ........................................................................... 38

A Osmocom TCP/UDP Port Numbers ......................................... 47

B Bibliography / References .................................................... 48
  B.0.0.1 References .............................................................. 48

C GNU Free Documentation License ........................................ 52
  C.1 PREAMBLE ...................................................................... 52
  C.2 APPLICABILITY AND DEFINITIONS ................................. 52
  C.3 VERBATIM COPYING ...................................................... 53
  C.4 COPYING IN QUANTITY .................................................. 53
  C.5 MODIFICATIONS ............................................................. 54
  C.6 COMBINING DOCUMENTS ............................................... 55
  C.7 COLLECTIONS OF DOCUMENTS ..................................... 55
  C.8 AGGREGATION WITH INDEPENDENT WORKS .................. 55
  C.9 TRANSLATION ............................................................... 56
  C.10 TERMINATION .............................................................. 56
  C.11 FUTURE REVISIONS OF THIS LICENSE ......................... 56
  C.12 RELICENSING ............................................................... 56
  C.13 ADDENDUM: How to use this License for your documents ... 57
1 Foreword

Digital cellular networks based on the GSM specification were designed in the late 1980ies and first deployed in the early 1990ies 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 shoudering 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

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 benefitting 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, work-arounds
• 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 http://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.

4 Overview

This manual should help you getting started with OsmoMGW. It will cover aspects of configuring and running the media gateway.
4.1 About OsmoMGW

OsmoMGW is the Osmocom implementation of a media gateway to handle user plane (voice) traffic in cellular networks. It can connect and optionally transcode RTP voice streams between different network elements such as BTSs and external entities like SIP. It is typically co-located with both OsmoBSC and OsmoMSC and controlled by them via MGCP, the Media Gateway Control Protocol.

4.2 Software Components

OsmoMGW contains a variety of different software components, which we’ll quickly describe in this section.

4.2.1 MGCP Implementation

OsmoMGW brings its own MGCP implementation through which OsmoMGW is controlled.

The commands implemented are CRCX, MDCX, DLCX and RSIP. The command AUEP is implemented as a stub and will simply respond with OK.
4.2.2 RTP implementation

Support for IuUP which is used in 3G cells is quite lacking at the moment. 3G→3G and 2G→2G calls should work, but 3G→2G does not.

4.2.3 Audio transcoder

Transcoding is currently not supported in OsmoMGW.

4.3 Limitations

At the moment (November 2020), OsmoMGW implements RTP proxy / RTP bridge type endpoints and E1/T1 16k/8k sub-slots with TRAU frames for classic BTS support. To the RTP proxy / RTP bridge endpoints two RTP connections can be established, which then work as a tandem. E1/T1 endpoints support one RTP connection at a time that is associated with a sub-slot on an E1 line. We are planning to add further endpoint types for:

- classic E1/T1 timeslots (64kBps alaw/ulaw)
- announcement/playout end-points
- conference endpoints

4.4 Additional resources

You can find the OsmoMGW issue tracker and wiki online at

- https://osmocom.org/projects/osmo-mgw
- https://osmocom.org/projects/osmo-mgw/wiki

RFC 3435 for MGCP is located at


5 Running OsmoMGW

The OsmoMGW executable (osmo-mgw) offers the following command-line arguments:

5.1 SYNOPSIS

osmo-mgw [-hV] [-D] [-c CONFIGFILE] [-s]

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**, **--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-mgw.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 7
   for more information.

6 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 1: VTY Parameter Patterns

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Example</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.B.C.D</td>
<td>127.0.0.1</td>
<td>An IPv4 address</td>
</tr>
<tr>
<td>A.B.C.D/M</td>
<td>192.168.1.0/24</td>
<td>An IPv4 address and mask</td>
</tr>
<tr>
<td>X::X:0:0:0:0:0:0</td>
<td>::1</td>
<td>An IPv6 address</td>
</tr>
<tr>
<td>X::X:0:0:0:0:0:0</td>
<td>::1/128</td>
<td>An IPv6 address and mask</td>
</tr>
<tr>
<td>TEXT</td>
<td>example01</td>
<td>A single string without any spaces, tabs</td>
</tr>
<tr>
<td>.TEXT</td>
<td>Some information</td>
<td>A line of text</td>
</tr>
<tr>
<td>(OptionA</td>
<td>OptionB</td>
<td>OptionC)</td>
</tr>
<tr>
<td>&lt;0-10&gt;</td>
<td>5</td>
<td>A number from a range</td>
</tr>
</tbody>
</table>

6.1 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.

---

6.2 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.
6.3 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.

6.3.1 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

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

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>Displays program version</td>
</tr>
<tr>
<td>online-help</td>
<td>Online help</td>
</tr>
<tr>
<td>history</td>
<td>Display the session command history</td>
</tr>
<tr>
<td>cs7</td>
<td>ITU-T Signaling System 7</td>
</tr>
<tr>
<td>logging</td>
<td>Show current logging configuration</td>
</tr>
<tr>
<td>alarms</td>
<td>Show current logging configuration</td>
</tr>
<tr>
<td>talloc-context</td>
<td>Show talloc memory hierarchy</td>
</tr>
<tr>
<td>stats</td>
<td>Show statistical values</td>
</tr>
<tr>
<td>asciidoc</td>
<td>Asciidoc generation</td>
</tr>
<tr>
<td>rate-counters</td>
<td>Show all rate counters</td>
</tr>
<tr>
<td>fsm</td>
<td>Show information about finite state machines</td>
</tr>
<tr>
<td>fsm-instances</td>
<td>Show information about finite state machine instances</td>
</tr>
<tr>
<td>sgs-connections</td>
<td>Show SGs interface connections / MMEs</td>
</tr>
<tr>
<td>subscriber</td>
<td>Operations on a Subscriber</td>
</tr>
<tr>
<td>bsc</td>
<td>BSC</td>
</tr>
<tr>
<td>connection</td>
<td>Subscriber Connections</td>
</tr>
<tr>
<td>transaction</td>
<td>Transactions</td>
</tr>
</tbody>
</table>

Copyright © 2018 sysmocom - s.f.m.c. GmbH
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".

6.3.2 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> s

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

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

OsmoMSC> show

Type <tab> here.

6.3.3 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

<table>
<thead>
<tr>
<th>statistics</th>
<th>Display network statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>sms-queue</td>
<td>Display SMSqueue statistics</td>
</tr>
<tr>
<td>smpp</td>
<td>SMPP Interface</td>
</tr>
</tbody>
</table>

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

| Type <tab> here. |

| Type <tab> here. |

| Type <tab> here. |

| Type <tab> here. |

| Type <tab> here. |
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|mac|mgcp|ho|db|ref|ctrl|smpp|ranap|vlr|iusc|bssap|sgs|lglobal|llapd|llnp|lmux|lm|lms|lctrl|lgtp|lstats|lgaup|loap|lss7|laccp|lsua|lsua|m3ua|mgcp|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) ← tmsi|id) SENDER_ID send .LINE
subscriber (msisdn|extension|imsi|tmsi|id) ID silent-call start (any|tch/f|tch/any|sdccch)
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
```

6.3.4 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 implements 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
.. ... exit
.. ... end
. o.. type (unknown|bs11|nanobts|rbs2000|nokia_site|sysmobts)
. ... description .TEXT
. ... description
. o.. band BAND
. .r. cell_identity <0-65535>
. .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)
! ... nokia_site bts-reset-timer <15-100>
```

- This command has no attributes assigned.
- This command applies on A-bis OML link (re)establishment.
- This command applies on A-bis RSL link (re)establishment.
- 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.

### 6.3.5 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
OsmoBSC# list with-flags
...       
^ bts <0-255> (activate-all-lchan|deactivate-all-lchan)
  
^ bts <0-255> trx <0-255> (activate-all-lchan|deactivate-all-lchan)
.
^ bts <0-255> trx <0-255> timeslot <0-7> sub-slot <0-7> mdcx A.B.C.D 0-65535
.
^ bts <0-255> trx <0-255> timeslot <0-7> sub-slot <0-7> (borken|unused)
.
^ bts <0-255> trx <0-255> timeslot <0-7> sub-slot <0-7> handover <0-255>
.
^ bts <0-255> trx <0-255> timeslot <0-7> sub-slot <0-7> assignment
.
    bts <0-255> smscb-command (normal|schedule|default) <1-4> HEXSTRING
...  
```

1 This command enables the *expert* mode.

2, 3 This is a hidden command (only shown in the *expert* mode).

4, 5, 6, 7 This is a regular command that is always shown regardless of the mode.

### 7 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.

#### 7.1 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 a list of categories interactively on the vty, type: `logging level ?`
7.2 Log levels

For each of the log categories (see Section 7.1), 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.

7.3 Log printing options

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

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

- **log print category**
  Prefix each log message with the category name.

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

- **log print level**
  Prefix each log message with the name of the log level.

- **log print file**
  Prefix each log message with the source file and line number. Append the keyword last to append the file information instead of prefixing it.
7.4 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

7.5 Log targets

Each of the log targets represent certain destination for log messages. It can be configured independently by selecting levels (see Section 7.2) for categories (see Section 7.1) as well as filtering (see Section 7.4) and other options like logging timestamp for example.

7.5.1 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 7.1 for more details on categories and Section 7.2 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 7.4.

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

7.5.2 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.
7.5.3 Logging via gsmtap

When debugging complex issues it’s handy to be able to reconstruct exact chain of events. This is enabled by using GSMTAP log output where frames sent/received over the air are interspersed with the log lines. It also simplifies the bug handling as users don’t have to provide separate .pcap and .log files anymore - everything will be inside self-contained packet dump.

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.

```
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

No. Time Source Src Port Destination Dist Port Info
1 16.7714909505 127.0.0.1 4729 192.168.2.3 shut down omf for BTS 0
```

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 decrease 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
```

7.5.4 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.

### 7.5.5 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.

### 7.5.6 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.


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](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 ❷
logging filter all 1
logging print file basename
logging print category-hex 0
logging print category 1
logging print level 1
logging timestamp 0 ❸
logging color 1 ❹
logging level set-all notice
```

- ❷ Generic logging handler, rendering is done by libosmocore.
- ❸ Disable timestamping, systemd will timestamp every message anyway.
- ❹ 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.
7.5.7 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)#
```

8 Configuring OsmoMGW

A basic configuration of OsmoMGW mainly consists of specifying the IP address and port on which to listen to MGCP commands, but changing the port range at which the RTP streams terminate as well as limiting operation to a single call agent can be done as well as changing the number of endpoints.

8.1 Configuring MGCP

By default OsmoMGW listens for MGCP on port 2427 on any IP address. If a call agent address is configured then OsmoMGW will only answer MGCP commands from that IP port 2727, otherwise all sources are handled. A domain can be specified

Example: MGCP configuration

```
OsmoMGW(config-mgcp)# bind ip 127.0.0.1
OsmoMGW(config-mgcp)# bind port 2427
OsmoMGW(config-mgcp)# call-agent ip 127.0.0.1
OsmoMGW(config-mgcp)# domain mgw-bsc
OsmoMGW(config-mgcp)# local ip 127.0.0.1
```

8.2 Configuring the trunk

The first trunk is considered a virtual trunk in OsmoMGW. All endpoints of type "rtpbridge" are routed here. The virtual trunk is configured in the config-mgcp context.

All other trunks are configured in the config-mgcp-trunk context, but the commands used are identical. Right now trunks are considered only for "ds/e1" type endpoints.

Example: MGCP trunk configuration

```
OsmoMGW(config-mgcp)# number endpoints 63
OsmoMGW(config-mgcp)# rtp bind-ip 10.0.0.1
OsmoMGW(config-mgcp)# rtp port-range 12000-14000
```

- Maximum number of endpoints that can be allocated at once
- Use this IP when binding RTP endpoints
- Use ports in this range when binding RTP endpoints

There are some options to tweak how RTP forwarding behaves in OsmoMGW:

Example: MGCP trunk rtp options

```
OsmoMGW(config-mgcp)# rtp keep-alive 30
OsmoMGW(config-mgcp)# rtp-patch ssrc
OsmoMGW(config-mgcp)# rtp-patch timestamp
```
8.3 E1 trunk considerations

While the RTP bridge trunks are natively based on IP no special considerations are required during setup. E1 trunks are mapped on a physical E1 line, which has to be configured as shown below.

Example: E1 line setup

```
OsmoMGW(config-e1_input)# el_line 0 driver dahdi
OsmoMGW(config-e1_input)# el_line 0 port 2
```

1. Name of the libosmo-abis driver implementation ("dahdi")
2. Port number of the physical E1 port to use (2)

In osmo-mgw the e1_input node is used to configure the physical E1 line. The line number will be used internally to identify the configured E1 line. The port number is the physical E1 connector (sometimes called span) at the E1 hardware. Per trunk an individual E1 line will be needed. Beware that the E1 driver may also need configuration settings that are not discussed here.

Example: E1 trunk setup

```
OsmoMGW(config-mgcp)# trunk 0
OsmoMGW(config-mgcp-trunk)# line 0
```

1. Creation of a trunk (0)
2. Reference to the E1 line to use (0)

The E1 trunk is created along with a number, typically starting at 0, but if required any number from 0-64 is allowed. The E1 trunk configuration concerning the IP related aspects is nearly identical to the configuration of the virtual trunk. However, it is important that the user assigns one of the E1 line numbers that were configured under the e1_input node.

Note

The endpoint name that is used on MGCP level will include the trunk number, not the E1 line number. For simplicity (and compatibility with OsmoBSC) it is recommended to use equal numbers for trunk and E1 line. However, if required any E1 line can be mapped flexible on any trunk as long as the mapping is bijective.

Example: A typical configuration with one E1 trunk

```
e1_input
  el_line 0 driver dahdi
  el_line 0 port 2
mgcp
  bind ip 127.0.0.1
  rtp net-range 6000 6011
  rtp net-bind-ip 192.168.100.130
```
rtp ip-probing
rtp ip-dscp 46
no rtp keep-alive
bind port 2428
number endpoints 30
loop 0
force-realloc 1
osmux off
rtp-patch rfc5993hr
trunk 0
rtp keep-alive once
no rtp keep-alive
line 0

Note
One E1 trunk always covers a whole E1 line. All subslots (I.640) will be mapped to individual MGCP endpoints. As long as the endpoints remain unused the underlying E1 timeslot is not used.

Note
The E1 trunk implementation also works with T1 lines, however since T1 has 24 instead of 31 usable timeslots only the endpoints that fall into that 1-24 timeslot range will be useable.

9 MGCP Endpoints

MGCP organizes the switching resources in so called endpoints. Each endpoint is referenced by its unique identifier. While RFC 3435 specifies a naming scheme, the actual identifier naming is subject to the implementation and configuration.

9.1 RTP proxy / RTP bridge endpoints

OsmoMGW implements a freely configurable number of rtpbridge endpoints. Those endpoints are able to host two connections at a time to model the functionality of a tandem switch.

RTP bridge endpoint identifiers are referenced by the string rtpbridge/, a hexadecimal number without leading zeros and a domain name (configurable).

Example: List of virtual endpoints

rtpbridge/1@mgw
rtpbridge/2@mgw
rtpbridge/3@mgw
rtpbridge/4@mgw
rtpbridge/5@mgw
rtpbridge/6@mgw
rtpbridge/7@mgw
rtpbridge/8@mgw
rtpbridge/9@mgw
rtpbridge/a@mgw
rtpbridge/b@mgw
rtpbridge/c@mgw
rtpbridge/d@mgw
rtpbridge/e@mgw
rtpbridge/f@mgw
rtpbridge/10@mgw


9.2 E1/T1 endpoints

OsmoMGW supports E1 subslot multiplexing as specified by I.460. All possible subslot combinations are mapped on individual endpoints. The endpoint names are prefixed with `ds/e1-` followed by the trunk number and the E1 timeslot. The subslot is defined by a bit rate and a bit offset.

```
ds/e1-<trunk>/s-<timeslot>/su<bitrate>-<bitoffset>@<domain>
```

Example: List of endpoints on E1 trunk 0 at E1 timeslot 2

```
ds/e1-0/s-2/su64-0@mgw
ds/e1-0/s-2/su32-0@mgw
ds/e1-0/s-2/su32-4@mgw
ds/e1-0/s-2/su16-0@mgw
ds/e1-0/s-2/su16-2@mgw
ds/e1-0/s-2/su16-4@mgw
ds/e1-0/s-2/su16-6@mgw
ds/e1-0/s-2/su8-0@mgw
ds/e1-0/s-2/su8-1@mgw
ds/e1-0/s-2/su8-2@mgw
ds/e1-0/s-2/su8-3@mgw
ds/e1-0/s-2/su8-4@mgw
ds/e1-0/s-2/su8-5@mgw
ds/e1-0/s-2/su8-6@mgw
ds/e1-0/s-2/su8-7@mgw
```

When creating connections on endpoints that reside in one E1 timeslot the call agent must make sure that no overlapping endpoints are used. It is for example not possible to use `ds/e1-0/s-2/su16-2@mgw` and `ds/e1-0/s-2/su8-3@mgw` at the same time because they overlap.

Table 2: Subslot overlapping

<table>
<thead>
<tr>
<th>Bit offset</th>
<th>Subslots</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8k</td>
</tr>
<tr>
<td>1</td>
<td>8k</td>
</tr>
<tr>
<td>2</td>
<td>8k</td>
</tr>
<tr>
<td>3</td>
<td>8k</td>
</tr>
<tr>
<td>4</td>
<td>8k</td>
</tr>
<tr>
<td>5</td>
<td>8k</td>
</tr>
<tr>
<td>6</td>
<td>8k</td>
</tr>
<tr>
<td>7</td>
<td>8k</td>
</tr>
</tbody>
</table>

Note
The current implementation (December 2020) only implements TRAU frame encoding/decoding for 16K and 8K subslots. Endpoints with other bitrates are not yet usable.

Note
the VTY command "show mgcp" can be used to get a list of all available endpoints (including identifiers)

10 MGCP Extensions

The MGCP protocol is extendable. The following non-standard extensions are understood by OsmoMGW.
10.1 X-Osmo-IGN

X-Osmo-IGN indicates to OsmoMGW that specific items of an endpoint should be ignored, so that it is lenient on mismatching values that would normally indicate collisions or configuration errors.

10.1.1 X-Osmo-IGN Format

The value part of X-Osmo-IGN must be one or more items separated by one or more spaces. Each item consists of one or more non-whitespace characters.

Example: X-Osmo-IGN format with three ficticious items "X", "abc" and "123".

```
X-Osmo-IGN: X abc 123
```

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

10.1.2 Supported X-Osmo-IGN Items

Currently, the following X-Osmo-IGN items are supported:

- **C**: ignore CallID mismatches, i.e. differing "C" values between connections on the same endpoint.

Note: X-Osmo-IGN does not support ignoring mismatches on the domain part of an endpoint name, e.g. ignoring a mismatch on "example.com" in rtpbridge/123abc@example.com. Instead, you may globally configure OsmoMGW with mgcp/domain * to permit all domain parts.

10.1.2.1 X-Osmo-IGN: C

By default, OsmoMGW verifies that all CallIDs ("C" values) match for all connections on any one given endpoint. To ignore CallID mismatches, pass a C in the X-Osmo-IGN header, for the first or the second CRCX on an endpoint. When the X-Osmo-IGN: C is sent for any one CRCX on an endpoint, CallID mismatches will be ignored for that and all subsequent messages for that endpoint. Hence this header only needs to be included once per endpoint, in any CRCX message that precedes or coincides with a CallID mismatch.

This is particularly useful for a BSC that is connected to an A/SCCP-lite MSC, where the BSC and MSC each are expected to configure their respective own connection on a shared endpoint. For A/SCCP-lite, it is impossible for the BSC to know the CallID that the MSC will use, so CallID mismatches are inevitable. See also OsmoBSC, which will by default pass the X-Osmo-IGN: C header for endpoints associated with an A/SCCP-lite MSC.

Example: CRCX message that instructs OsmoMGW to ignore CallID mismatches

```
CRCX 2 rtpbridge/123abc@mgw MGCP 1.0
M: recvonly
C: 2
L: p:20
X-Osmo-IGN: C

v=0
c=IN IP4 123.12.12.123
m=audio 5904 RTP/AVP 97
a=rtpmap:97 GSM-EFR/8000
a=ptime:40
```

10.2 X-Osmux

See Section 11.6
11 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.


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.

11.1 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.

Moreover, NATs 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. That’s the reason why OsmoMGW provides the osmux dummy VTY command to enable sending dummy packets from time to time to keep the connections alive.

11.2 CID allocation

Each peer (BSC/MGW and MSC/MGW) allocates its own recvCID, and receives from the peer through the used GSM protocol the peer’s recvCID, which becomes the local sendCID for that connection.

| BSC/MGW(recvCID=Y,sendCID=?)|--X-->MSC/MGW(recvCID=X,sendCID=?) |
| BSC/MGW(recvCID=Y,sendCID=X)--Y-->MSC/MGW(recvCID=X,sendCID=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, sendCID> 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 `recvCID` 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).

### 11.3 3GPP AoIP network setup with Osmux

![Sample node diagram of a 3GPP AoIP network with Osmux enabled](image)

Figure 5: Sample node diagram of a 3GPP AoIP network with Osmux enabled
Figure 6: MO-call with Osmux enable on 3GPP AoIP
11.4 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
11.5 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/SCCP-lite with a BSC-NAT between BSC and MSC
11.6 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.

11.6.1 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 sendCID.

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 recvCID.

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: *
```

11.6.2 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 sendCID, it can use an astersik (*) and provide sendCID later within MDCX messages.

All subsequent MDCX messages sent towards an Osmux connection must contain the original sendCID sent during CRCX. The same way, all MDCX response shall contain the recvCID 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
200 189 OK
I: 07E41584
X-Osmux: 2
Z: rtpbridge/1@mgw

v=0
c=IN IP4 172.18.1.20
m=audio 11002 RTP/AVP 112
a=rtpmap:112 AMR/8000
a=ptime:20

11.6.3 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

```
mgrpc
...
osmux on
osmux bind-ip 172.18.1.20
osmux port 1984
osmux batch-factor 4
osmux dummy on
```

1. Allow clients to set allocate Osmux connections in rtpbridge endpoints, while still allowing RTP connections
2. Bind the Osmux socket to the provided IP address
3. Bind the Osmux socket to the provided UDP port
4. Batch up to 4 RTP payloads of the same stream on each Osmux frame
5. Periodically send Osmux dummy frames, useful to punch a hole in NATs and maintain connections opened.

12 QoS, DSCP/TOS, Priority and IEEE 802.1q PCP

In many use cases operators want to apply different QoS classes for user plane vs. control plane traffic. IP Routers, Ethernet switches and other network gear can then perform intelligent queue management as required for the respective service.

For example, voice user plane frames need a rather stable and short latency, while IP user plane and control plane traffic has less critical latency requirements.

12.1 IP Level (DSCP)

At IP level, different priorities / classes of traffic are expressed in accordance to [ietf-rfc2474] by the DSCP (Differentiated Services Code Point) field of the IP header. DSCP resembles the upper 6 bits of the field formerly known as the TOS bits as per [ietf-rfc791].
On Linux and other operating systems with BSD-style sockets API, the applications can request a specific DSCP value to be used for packets generated by those sockets.

Osmocom CNI software such as osmo-bts and osmo-mgw support setting the DSCP value via VTY commands, see e.g. the `rtp ip-dscp` setting of the `bts` node in osmo-bts.

### 12.2 Packet Priority

In the Linux network stack, every packet is represented by `struct sk_buff`, which has an associated *priority*. Furthermore, every socket through which applications send data have an associated *socket priority*. Each time a packet is transmitted through a given socket, the packet inherits the packet priority from the socket priority.

Furthermore, there is a mapping table that maps DSCP/TOS bits to priority. The sixteen different TOS bit values are mapped to priority values as follows:

<table>
<thead>
<tr>
<th>TOS (binary)</th>
<th>DSCP (binary)</th>
<th>Priority (decimal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>xxx0000x</td>
<td>xxx000</td>
<td>0</td>
</tr>
<tr>
<td>xxx0001x</td>
<td>xxx000</td>
<td>0</td>
</tr>
<tr>
<td>xxx0010x</td>
<td>xxx001</td>
<td>0</td>
</tr>
<tr>
<td>xxx0011x</td>
<td>xxx001</td>
<td>0</td>
</tr>
<tr>
<td>xxx0100x</td>
<td>xxx010</td>
<td>2</td>
</tr>
<tr>
<td>xxx0101x</td>
<td>xxx010</td>
<td>2</td>
</tr>
<tr>
<td>xxx0110x</td>
<td>xxx011</td>
<td>2</td>
</tr>
<tr>
<td>xxx0111x</td>
<td>xxx011</td>
<td>2</td>
</tr>
<tr>
<td>xxx1000x</td>
<td>xxx100</td>
<td>6</td>
</tr>
<tr>
<td>xxx1001x</td>
<td>xxx100</td>
<td>6</td>
</tr>
<tr>
<td>xxx1010x</td>
<td>xxx101</td>
<td>6</td>
</tr>
<tr>
<td>xxx1011x</td>
<td>xxx101</td>
<td>6</td>
</tr>
<tr>
<td>xxx1100x</td>
<td>xxx110</td>
<td>4</td>
</tr>
<tr>
<td>xxx1101x</td>
<td>xxx110</td>
<td>4</td>
</tr>
<tr>
<td>xxx1110x</td>
<td>xxx111</td>
<td>4</td>
</tr>
<tr>
<td>xxx1111x</td>
<td>xxx111</td>
<td>4</td>
</tr>
</tbody>
</table>

This table of default DSCP/TOS → priority bit mappings cannot be modified.

However, the per-packet *priority* values can be set by various means of network policy, including

- by packet filter rules (iptables, ip6tables, nftables)
  - if you use *iptables*, using `CLASSIFY --set-class` in the mangle table
  - if you use *nftables*, using `meta priority set` in the mangle table

- by the application using the `SO_PRIORITY` socket option (currently not yet supported by Osmocom CNI)

### 12.3 Ethernet Level (PCP)

At Ethernet level, different priorities / QoS classes are expressed by the so-called PCP (Priority Code Point) field in the IEEE 802.1q (VLAN) header.

**NOTE**

This means that PCP functionality requires the use of IEEE 802.1q VLAN. You cannot use PCP without VLAN.
The Linux kernel assigns IEEE 802.1q PCP bits based on a mapping between the priority and the PCP value. Each VLAN network device maintains a separate map for both egress (transmit) and ingress (receive) path.

The current priority mappings can be inspected via the /proc filesystem. For example, if you have a VLAN device eth0.9 for VLAN ID 9 on the net-device eth0, you can use the following example:

Example: Inspecting the current egress QoS map

```
$ sudo cat /proc/net/vlan/eth0.9
eth0.9 VID: 9 REORDER_HDR: 1 dev->priv_flags: 1021
  total frames received 123340
  total bytes received 40668066
  Broadcast/Multicast Rcvd 1106

  total frames transmitted 10499
  total bytes transmitted 1570809
Device: eth0
INGRESS priority mappings: 0:0 1:0 2:0 3:0 4:0 5:0 6:0 7:0
EGRESS priority mappings: 1:1 2:2 3:3
```

1. make sure to specify your specific VLAN interface name here instead of eth0.9
2. ingress priority mappings (all PCP values mapped to priority 0)
3. egress priority mappings (empty)

As we can see in the above example, there are no egress priority mappings yet. Let’s create three new mappings, mapping priority value 1 to PCP 1, priority 2 to PCP 2, and priority 3 to PCP 3:

Example: Creating three new egress QoS mappings

```
$ sudo ip link set dev eth0.9 type vlan egress-qos-map 1:1 2:2 3:3
$ sudo cat /proc/net/vlan/eth0.9
eth0.9 VID: 9 REORDER_HDR: 1 dev->priv_flags: 1021
  total frames received 123898
  total bytes received 40843611
  Broadcast/Multicast Rcvd 1106

  total frames transmitted 10517
  total bytes transmitted 1574357
Device: eth0
INGRESS priority mappings: 0:0 1:0 2:0 3:0 4:0 5:0 6:0 7:0
EGRESS priority mappings: 1:1 2:2 3:3
```

1. make sure to specify your specific VLAN interface name here instead of eth0.9
2. command to define three new egress QoS maps
3. command to re-display the current status
4. three new egress mappings are shown as given in ip command

**NOTE**
The settings of the ip command are volatile and only active until the next reboot (or the network device or VLAN is removed). Please refer to the documentation of your specific Linux distribution in order to find out how to make such settings persistent by means of an ifup hook whenever the interface comes up. For CentOS/RHEL 8 this can e.g. be achieved by means of an /sbin/ifup-local script (when using network-scripts and not NetworkManager). For Debian or Ubuntu, this typically involves adding up lines to /etc/network/interfaces or a /etc/network/if-up.d script.
12.4 Putting things together

Assuming one needs to set both the DSCP bits as well as the PCP for certain traffic, the above-mentioned mechanisms need to be combined as follows:

1. configure the osmocom program to set the DSCP value
2. use the default DSCP → priority mapping, if possible
3. configure an egrees QoS map to map from priority to PCP

If the desired combination of DSCP + PCP cannot be achieved that way, due to the rather static default kernel mapping table, one needs to go one step further:

1. configure the osmocom program to set the DSCP value
2. use packet filter rules to set the priority based on DSCP
3. configure an egrees QoS map to map from priority to PCP

12.4.1 Full example of QoS for osmo-mgw

In the below example we will show the full set of configuration required for both DSCP and PCP differentiation of RTP traffic by osmo-mgw.

What we want to achieve in this example is the following configuration:

Table 4: DSCP and PCP assignments for osmo-mgw Abis downlink traffic in this example

<table>
<thead>
<tr>
<th>Traffic</th>
<th>DSCP</th>
<th>PCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTP</td>
<td>46</td>
<td>6</td>
</tr>
</tbody>
</table>

1. configure the osmo-mgw program to set the DSCP value
2. configure an egrees QoS map to map from priority to PCP

Example Step 1: add related VTY configuration to osmo-mgw.cfg

```plaintext
... 
mgcp 
  rtp ip-dscp 46 
  rtp socket-priority 6 
... 
```

Example Step 2: egress QoS map to map from socket priority to PCP values

```
$ sudo ip link set dev eth0.9 type vlan egress-qos-map 0:0 5:5 6:6 7:7
```

1. make sure to specify your specific VLAN interface name here instead of `eth0.9`.
2. create a egress QoS map that maps the priority value 1:1 to the PCP. We also include the mapping 5:5 and 7:7 from the osmo-bsc example (see [userman-osmobsc]) here.
NOTE
The settings of the `ip` command are volatile and only active until the next reboot (or the network device or VLAN is removed). Please refer to the documentation of your specific Linux distribution in order to find out how to make such settings persistent by means of an `ifup` hook whenever the interface comes up. For CentOS/RHEL 8 this can e.g. be achieved by means of an `/sbin/ifup-local` script (when using `network-scripts` and not NetworkManager). For Debian or Ubuntu, this typically involves adding `up` lines to `/etc/network/interfaces` or a `/etc/network/if-up.d` script.

13 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.

13.1 Scheduling Policy

The scheduler to use as well as some of its properties (such as real-time 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
```

Configure process to use `SCHED_RR` policy with real time priority 1

13.2 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
```

Allow main thread (the one managing the VTY) only on CPU 0

Or otherwise:

```
cpu-sched
   cpu-affinity all 0x01
```

Copyright © 2018 sysmocom - s.f.m.c. GmbH

DRAFT 1.9.0-18-ge0058, 2021-Dec-25
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**

```plaintext
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**

```plaintext
OsmoTRX> cpu-affinity TxLower 0x02
OsmoTRX> cpu-affinity TxLower 0x03
```

Allow thread named *TxLower* (338635) only on CPU 1

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**

```plaintext
cpu-sched
cpu-affinity TxLower 0x01 delay
```

Allow thread named *TxLower* (338635) only on CPU 1. It will be applied by the thread itself when created.

### 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 MILLENAGE 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

AQP SK
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])

BVC I
BSSGP Virtual Circuit Identifier

CBCH
Cell Broadcast Channel; used to transmit Cell Broadcast SMS (SMS-CB)
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
Circuit-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](https://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](https://www.ietf.org/rfc/rfc793))

TFTP
Trivial File Transfer Protocol; ([IETF RFC 1350](https://www.ietf.org/rfc/rfc1350))

TOS
Type Of Service; bit-field in IPv4 header, now re-used as DSCP ([IETF RFC 791](https://www.ietf.org/rfc/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](https://www.ietf.org/rfc/rfc768))

UICC
Universal Integrated Chip Card; A smart card according to [ETSI TR 102 216](https://www.etsi.org/deliver/etsi-tr/102200/102216/16.01.01_60/tr102216v160101p.pdf)

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](https://www.3gpp.org/SpecsSpecDetails.aspx?SpecId=TS4801818)) 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](https://standards.ieee.org/standard/802.1q-2014.html))
**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 5: TCP/UDP port numbers](image)

```
<table>
<thead>
<tr>
<th>L4 Protocol</th>
<th>Port Number</th>
<th>Purpose</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP</td>
<td>2427</td>
<td>MGCP GW</td>
<td>osmo-bsc_mgc, osmo-mgw</td>
</tr>
<tr>
<td>TCP</td>
<td>2775</td>
<td>SMPP (SMS interface for external programs)</td>
<td>osmo-nitb</td>
</tr>
<tr>
<td>TCP</td>
<td>3002</td>
<td>A-bis/IP OML</td>
<td>osmo-bts, osmo-bsc, osmo-nitb</td>
</tr>
<tr>
<td>TCP</td>
<td>3003</td>
<td>A-bis/IP RSL</td>
<td>osmo-bts, osmo-bsc, osmo-nitb</td>
</tr>
<tr>
<td>TCP</td>
<td>4227</td>
<td>telnet (VTY)</td>
<td>osmo-pcap-client</td>
</tr>
<tr>
<td>TCP</td>
<td>4228</td>
<td>telnet (VTY)</td>
<td>osmo-pcap-server</td>
</tr>
<tr>
<td>TCP</td>
<td>4236</td>
<td>Control Interface</td>
<td>osmo-trx</td>
</tr>
<tr>
<td>TCP</td>
<td>4237</td>
<td>telnet (VTY)</td>
<td>osmo-trx</td>
</tr>
<tr>
<td>TCP</td>
<td>4238</td>
<td>Control Interface</td>
<td>osmo-bts</td>
</tr>
<tr>
<td>TCP</td>
<td>4239</td>
<td>telnet (VTY)</td>
<td>osmo-stp</td>
</tr>
<tr>
<td>TCP</td>
<td>4240</td>
<td>telnet (VTY)</td>
<td>osmo-pcu</td>
</tr>
<tr>
<td>TCP</td>
<td>4241</td>
<td>telnet (VTY)</td>
<td>osmo-bts</td>
</tr>
<tr>
<td>TCP</td>
<td>4242</td>
<td>telnet (VTY)</td>
<td>osmo-nitb, osmo-bsc, cellmgr-ng</td>
</tr>
<tr>
<td>TCP</td>
<td>4243</td>
<td>telnet (VTY)</td>
<td>osmo-bsc_mgc, osmo-mgw</td>
</tr>
<tr>
<td>TCP</td>
<td>4244</td>
<td>telnet (VTY)</td>
<td>osmo-bsc_nat</td>
</tr>
<tr>
<td>TCP</td>
<td>4245</td>
<td>telnet (VTY)</td>
<td>osmo-sgsn</td>
</tr>
<tr>
<td>TCP</td>
<td>4246</td>
<td>telnet (VTY)</td>
<td>osmo-gbproxy</td>
</tr>
<tr>
<td>TCP</td>
<td>4247</td>
<td>telnet (VTY)</td>
<td>OsmocomBB</td>
</tr>
<tr>
<td>TCP</td>
<td>4249</td>
<td>Control Interface</td>
<td>osmo-nitb, osmo-bsc</td>
</tr>
<tr>
<td>TCP</td>
<td>4250</td>
<td>Control Interface</td>
<td>osmo-bsc_nat</td>
</tr>
<tr>
<td>TCP</td>
<td>4251</td>
<td>Control Interface</td>
<td>osmo-sgsn</td>
</tr>
<tr>
<td>TCP</td>
<td>4252</td>
<td>telnet (VTY)</td>
<td>sysmobts-mgr</td>
</tr>
<tr>
<td>TCP</td>
<td>4253</td>
<td>telnet (VTY)</td>
<td>osmo-gtphub</td>
</tr>
<tr>
<td>TCP</td>
<td>4254</td>
<td>telnet (VTY)</td>
<td>osmo-msc</td>
</tr>
<tr>
<td>TCP</td>
<td>4255</td>
<td>Control Interface</td>
<td>osmo-msc</td>
</tr>
<tr>
<td>TCP</td>
<td>4256</td>
<td>telnet (VTY)</td>
<td>osmo-sip-connector</td>
</tr>
<tr>
<td>TCP</td>
<td>4257</td>
<td>Control Interface</td>
<td>osmo-ggsn, ggsn (OpenGGSN)</td>
</tr>
<tr>
<td>TCP</td>
<td>4258</td>
<td>telnet (VTY)</td>
<td>osmo-hlr</td>
</tr>
<tr>
<td>TCP</td>
<td>4259</td>
<td>Control Interface</td>
<td>osmo-hlr</td>
</tr>
<tr>
<td>TCP</td>
<td>4260</td>
<td>telnet (VTY)</td>
<td>osmo-ggsn</td>
</tr>
<tr>
<td>TCP</td>
<td>4261</td>
<td>telnet (VTY)</td>
<td>osmo-hnbgw</td>
</tr>
<tr>
<td>TCP</td>
<td>4262</td>
<td>Control Interface</td>
<td>osmo-hnbgw</td>
</tr>
<tr>
<td>TCP</td>
<td>4263</td>
<td>Control Interface</td>
<td>osmo-gbproxy</td>
</tr>
<tr>
<td>TCP</td>
<td>4264</td>
<td>telnet (VTY)</td>
<td>osmo-cbc</td>
</tr>
<tr>
<td>TCP</td>
<td>4265</td>
<td>Control Interface</td>
<td>osmo-cbc</td>
</tr>
<tr>
<td>TCP</td>
<td>4266</td>
<td>D-GSM MS Lookup: mDNS serve</td>
<td>osmo-hlr</td>
</tr>
</tbody>
</table>
```
Table 5: (continued)

<table>
<thead>
<tr>
<th>L4 Protocol</th>
<th>Port Number</th>
<th>Purpose</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>4267</td>
<td>Control Interface</td>
<td>osmo-mgw</td>
</tr>
<tr>
<td>TCP</td>
<td>4268</td>
<td>telnet (VTY)</td>
<td>osmo-uecups</td>
</tr>
<tr>
<td>SCTP</td>
<td>4268</td>
<td>UECUPS</td>
<td>osmo-uecups</td>
</tr>
<tr>
<td>TCP</td>
<td>4269</td>
<td>telnet (VTY)</td>
<td>osmo-e1d</td>
</tr>
<tr>
<td>TCP</td>
<td>4271</td>
<td>telnet (VTY)</td>
<td>osmo-smc</td>
</tr>
<tr>
<td>TCP</td>
<td>4272</td>
<td>Control Interface</td>
<td>osmo-smc</td>
</tr>
<tr>
<td>UDP</td>
<td>4729</td>
<td>GSMTAP</td>
<td>Almost every osmocom project</td>
</tr>
<tr>
<td>TCP</td>
<td>5000</td>
<td>A/IP</td>
<td>osmo-bsc, osmo-bsc_nat</td>
</tr>
<tr>
<td>UDP</td>
<td>23000</td>
<td>GPRS-NS over IP default port</td>
<td>osmo-pcu, osmo-sgsn, osmo-gbproxy</td>
</tr>
</tbody>
</table>

B Bibliography / References

B.0.0.0.1 References

[58] [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


[61] [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

[62] [etsi-tr102216] ETSI TR 102 216: Smart cards https://www.etsi.org/deliver/etsi_tr/102200_102299/102216/03.00.00/03.00.00_60/tr_102216v030000p.pdf

[63] [etsi-ts102221] ETSI TS 102 221: Smart Cards; UICC-Terminal interface; Physical and logical characteristics https://www.etsi.org/deliver/etsi_ts/102200_102299/102221/13.01.00/13.01.00_60/ts_102221v130100p.pdf

[64] [etsi-ts101220] ETSI TS 101 220: Smart Cards; ETSI numbering system for telecommunication application providers https://www.etsi.org/deliver/etsi_ts/101200_101299/101220/12.00.00/12.00.00_60/ts_101220v120000p.pdf

[65] [ieee-802.1q] IEEE 802.1Q: Bridges and Bridged Networks https://ieeexplore.ieee.org/document/6991462


C GNU Free Documentation License

Version 1.3, 3 November 2008


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 [?].

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.