sysmocom

sysmocom - s.f.m.c. GmbH



OsmoPCU User Manual

by Harald Welte

DRAFT

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with 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".

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

HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
DRAFT	unknown		HW

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

Contents

1 Foreword **1**
1.1 Acknowledgements . . . 1
1.2 Endorsements . . . 2

2 Preface **2**
2.1 FOSS lives by contribution! . . . 2
2.2 Osmocom and sysmocom . . . 3
2.3 Corrections . . . 3
2.4 Legal disclaimers . . . 3
2.4.1 Spectrum License . . . 3
2.4.2 Software License . . . 3
2.4.3 Trademarks . . . 3
2.4.4 Liability . . . 4
2.4.5 Documentation License . . . 4

3 Introduction **4**
3.1 Required Skills . . . 4
3.2 Getting assistance . . . 5

4 Overview **5**
4.1 About OsmoPCU . . . 5
4.1.1 OsmoPCU in co-location with OsmoBTS . . . 5
4.1.2 OsmoPCU in co-location with OsmoBSC . . . 5
4.2 Software Components . . . 6
4.2.1 Gb Implementation . . . 6
4.2.2 `pcu_sock` Interface to OsmoBTS/OsmoBSC . . . 7

5 Running OsmoPCU **7**
5.1 SYNOPSIS . . . 7
5.2 OPTIONS . . . 7

6 The Osmocom VTY Interface **8**
6.1 Accessing the telnet VTY . . . 9
6.2 VTY Nodes . . . 9
6.3 Interactive help . . . 10
6.3.1 The question-mark (`?`) command . . . 10
6.3.2 TAB completion . . . 11
6.3.3 The `list` command . . . 11
6.3.4 The attribute system . . . 13
6.3.5 The expert mode . . . 14

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

7 libosmocore Logging System **15**
7.1 Log categories . . . 15
7.2 Log levels . . . 16
7.3 Log printing options . . . 16
7.4 Log filters . . . 17
7.5 Log targets . . . 17
7.5.1 Logging to the VTY . . . 17
7.5.2 Logging to the ring buffer . . . 17
7.5.3 Logging via gsmtap . . . 18
7.5.4 Logging to a file . . . 19
7.5.5 Logging to syslog . . . 20
7.5.6 Logging to systemd-journal . . . 20
7.5.7 Logging to stderr . . . 22

8 Configuring OsmoPCU **22**
8.1 Configuring the Coding Schemes and Rate Adaption . . . 23
8.1.1 Initial Coding Scheme . . . 23
8.1.2 Maximum Coding Scheme . . . 23
8.1.3 Rate Adaption Error Thresholds . . . 23
8.1.4 Rate Adation Link Quality Thresholds . . . 23
8.1.5 Data Size based CS downgrade Threshold . . . 23
8.2 Miscellaneous Configuration / Tuning Parameters . . . 24
8.2.1 Downlink TBF idle time . . . 24
8.2.2 MS idle time . . . 24
8.2.3 Forcing two-phase access . . . 24
8.3 Configuring BSSGP flow control . . . 24
8.3.1 Normal BSSGP Flow Control Tuning parameters . . . 24
8.3.2 Extended BSSGP Flow Control Tuning parameters . . . 24
8.4 Configuring LLC queue . . . 25
8.5 Configuring MS power control . . . 25
8.6 Configuring Network Assisted Cell Change (NACC) . . . 26
8.6.1 Neighbor Address Resolution . . . 26
8.6.1.1 OsmoBSC CTRL interface (deprecated) . . . 26
8.6.2 System Information Resolution . . . 27
8.7 Configuring E1 line for CCU access . . . 27
8.8 GPRS vs EGPRS considerations . . . 27
8.8.1 Configuration . . . 27
8.8.2 GPRS+EGPRS multiplexing . . . 28
8.9 Configuring GSMTAP tracing . . . 29

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

9 Counters **30**
9.1 Rate Counters . . . 30
9.2 Osmo Stat Items . . . 33

10 Code Architecture **35**

11 Gb interface using libosmogb **36**
11.1 Gb interface configuration . . . 36
11.1.1 NS-over-UDP configuration . . . 36
11.1.2 NS-over-FR-GRE configuration . . . 36
11.1.3 NS Timer configuration . . . 36
11.2 Examining Gb interface status . . . 37
11.3 FIXME . . . 37
11.3.1 Blocking / Unblocking / Resetting NS Virtual Connections . . . 37
11.4 Gb interface logging filters . . . 38

12 QoS, DSCP/TOS, Priority and IEEE 802.1q PCP **38**
12.1 IP Level (DSCP) . . . 38
12.2 Packet Priority . . . 38
12.3 Ethernet Level (PCP) . . . 39
12.4 Putting things together . . . 40
12.4.1 Full example of QoS for osmo-pcu uplink QoS . . . 41

13 VTY Process and Thread management **41**
13.1 Scheduling Policy . . . 42
13.2 CPU-Affinity Mask . . . 42

14 Glossary **43**

A Osmocom TCP/UDP Port Numbers **52**

B Bibliography / References **53**
B.0.0.0.1 References . . . 53

C GNU Free Documentation License **57**
C.1 PREAMBLE . . . 58
C.2 APPLICABILITY AND DEFINITIONS . . . 58
C.3 VERBATIM COPYING . . . 59
C.4 COPYING IN QUANTITY . . . 59
C.5 MODIFICATIONS . . . 59
C.6 COMBINING DOCUMENTS . . . 60
C.7 COLLECTIONS OF DOCUMENTS . . . 61

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

C.8 AGGREGATION WITH INDEPENDENT WORKS 61
C.9 TRANSLATION 61
C.10 TERMINATION 61
C.11 FUTURE REVISIONS OF THIS LICENSE 62
C.12 RELICENSING 62
C.13 ADDENDUM: How to use this License for your documents 62

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

1 Foreword

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

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

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

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

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

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

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

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

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

1.1 Acknowledgements

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

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

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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

May the source be with you!

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

1.2 Endorsements

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

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

2 Preface

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

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

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

2.1 FOSS lives by contribution!

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

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

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

We're looking forward to receiving your contributions.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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

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

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

2.4.4 Liability

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

2.4.5 Documentation License

Please see Appendix C for further information.

3 Introduction

3.1 Required Skills

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

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

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

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

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

Some of the useful skills we recommend are:

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

3.2 Getting assistance

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

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

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

4 Overview

4.1 About OsmoPCU

OsmoPCU is the Osmocom implementation of the GPRS PCU (Packet Control Unit) element inside the GPRS network. Depending on the BTS type the PCU will be co-located within the BTS or run in co-location with the BSC.

4.1.1 OsmoPCU in co-location with OsmoBTS

In most OsmoPCU-supported base stations, the PCU is co-located with the BTS. In this scenario OsmoPCU and OsmoBTS run on the same host system. Both are interconnected using a unix domain socket based interface. (see also: Section 4.2.2)

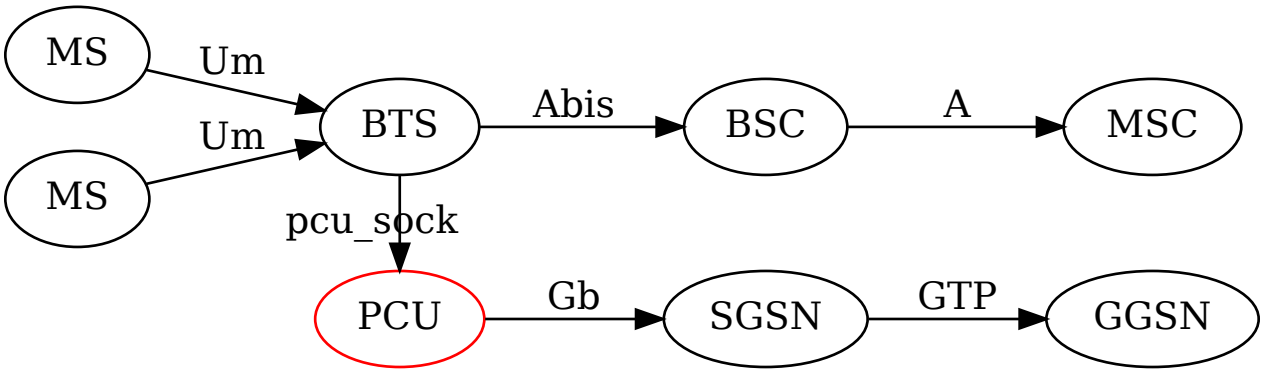


Figure 1: GPRS network architecture with PCU in BTS

Note
Depending on the hardware architecture, OsmoPCU may also have direct access on the PHY interface to exchange PDCH traffic efficiently. The socket interface is then only used for signalling.

4.1.2 OsmoPCU in co-location with OsmoBSC

Classic E1 based BTSs usually do not include a PCU. Instead those base stations typically rely on an external PCU that is co-located with the BSC. The signalling traffic (paging, channel assignments ect.) is then exchanged with the BTS via RSL, while the PDCH traffic is handled by the PCU through a dedicated TRAU frame based E1 connection.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

OsmoPCU supports this scenario as well. Due to the dedicated E1 connection, the implementation is complex and strongly hardware specific. As of now (March 2023) OsmoPCU supports Ericsson RBS2000/RBS6000 only. This implementation has been made possible through funding by the NLnet Foundation.

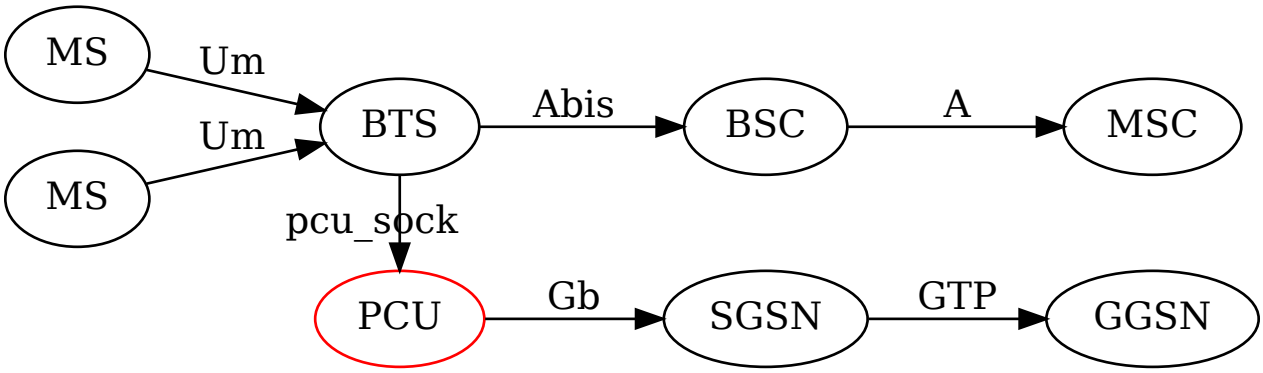


Figure 2: GPRS network architecture with PCU in BTS

When OsmoPCU runs in co-location to OsmoBSC, both are connected through the same unix domain socket interface as mentioned above. (see also: Section 4.1.1) The socket is used to pass signalling traffic between PCU and BSC while the PCU controls the PDCH by directly talking to the BTS CCU (channel coding unit) through a dedicated E1 connection. The E1 line interface uses TRAU frames and is vastly comparable to the interface that is used when speech is transferred.

Since the PCU is mainly set up by OsmoBSC (or OsmoBTS) via the PCU socket, the configuration in the BSC co-located scenario is no different from the BTS co-located scenario. However, since the PCU requires a direct E1 connection to the BTS an E1 line must be set up. (See also: Section 8.7)

4.2 Software Components

OsmoPCU consists of a variety of components, including

- Gb interface (NS/BSSGP protocol)
- `pcu_sock` interface towards OsmoBTS (or OsmoBSC)
- TBF management for uplink and downlink TBF
- RLC/MAC protocol implementation
- per-MS context for each MS currently served
- CSN.1 encoding/decoding routines

4.2.1 Gb Implementation

OsmoPCU implements the ETSI/3GPP specified Gb interface, including TS 08.16 (NS), TS 08.18 (BSSGP) protocols. As transport layer for NS, it supports NS/IP (NS encapsulated in UDP/IP).

The actual Gb Implementation is part of the libosmogb library, which is in turn part of the libosmocore software package. This allows the same Gb implementation to be used from OsmoPCU, OsmoGbProxy as well as OsmoSGSN.

The Gb Implementation supports two different dialects:

- `classic` dialect, which implements a non-standard dialect from the days where Frame Relay was the default and no IP transport had yet been specified by 3GPP.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

- `ip-sns` dialect, which implements the 3GPP standard for NS-over-IP, including the IP-SNS protocol.

In case of doubt, and assuming you are directly interfacing to a modern 3rd party SGSN, use the `ip-sns` dialect.

Selection of the dialect is done using the `gb-dialect` VTY option at the `pcu` node.

4.2.2 `pcu_sock` Interface to OsmoBTS/OsmoBSC

The interface towards OsmoBTS/OsmoBSC is called *pcu_sock* and implemented as a set of non-standardized primitives over a unix domain socket. The default file system path for this socket is `/tmp/pcu_bts`.

The PCU socket path can be freely configured to a different file/path name, primarily to permit running multiple independent BTS+PCU (or BSC+PCU) pairs on a single Linux machine without having to use filesystem namespaces or other complex configurations.

Note
If you change the PCU socket path on OsmoBTS/OsmoBSC by means of the `pcu-socket` VTY configuration command, you must ensure to make the identical change on the OsmoPCU side.

---

5 Running OsmoPCU

The OsmoPCU executable (`osmo-pcu`) offers the following command-line options:

5.1 SYNOPSIS

osmo-pcu [-hl-V] [-D] [-c *CONFIGFILE*] [-r *PRIO*] [-m *MCC*] [-n *MNC*] [-i A.B.C.D]

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-pcu.cfg` in the current working directory.

-m, --mcc *MCC*
Use the given MCC instead of that provided by BTS via PCU socket

-n, --mnc *MNC*
Use the given MNC instead of that provided by BTS via PCU socket

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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

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

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

Table 1: (continued)

Pattern	Example	Explanation
<0-10>	5	A number from a range

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.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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

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

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

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

Example: Typing ? after a partial command

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

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

```
  statistics       Display network statistics
  sms-queue        Display SMSqueue statistics
  smpp          SMPP Interface
```

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

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

Example: Typing ? after show bsc

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

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

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> s❶
show       sms        subscriber
```

❶ Type `<tab>` here.

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

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

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

❶ Type `<tab>` here.

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

```
OsmoMSC> list
  show version
  show online-help
  list
  exit
  help
  enable
  terminal length <0-512>
  terminal no length
  who
  show history
  show cs7 instance <0-15> users
```

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

```
show cs7 (sua|m3ua|ipa) [<0-65534>]
show cs7 instance <0-15> asp
show cs7 instance <0-15> as (active|all|m3ua|sua)
show cs7 instance <0-15> sccp addressbook
show cs7 instance <0-15> sccp users
show cs7 instance <0-15> sccp ssn <0-65535>
show cs7 instance <0-15> sccp connections
show cs7 instance <0-15> sccp timers
logging enable
logging disable
logging filter all (0|1)
logging color (0|1)
logging timestamp (0|1)
logging print extended-timestamp (0|1)
logging print category (0|1)
logging print category-hex (0|1)
logging print level (0|1)
logging print file (0|1|basename) [last]
logging set-log-mask MASK
logging level (rll|cc|mm|rr|mncc|pag|msc|mgcp|ho|db|ref|ctrl|smpp|ranap|vlr|iucs|bssap| ↩
    sgs|lglobal|llapd|linp|lmux|lmi|lmib|lsms|lctrl|lgtp|lstats|lgsup|loap|lss7|lsccp|lsua ↩
    |lm3ua|lmgcp|ljibuf|lrspro) (debug|info|notice|error|fatal)
logging level set-all (debug|info|notice|error|fatal)
logging level force-all (debug|info|notice|error|fatal)
no logging level force-all
show logging vty
show alarms
show talloc-context (application|all) (full|brief|DEPTH)
show talloc-context (application|all) (full|brief|DEPTH) tree ADDRESS
show talloc-context (application|all) (full|brief|DEPTH) filter REGEXP
show stats
show stats level (global|peer|subscriber)
show asciidoc counters
show rate-counters
show fsm NAME
show fsm all
show fsm-instances NAME
show fsm-instances all
show sgs-connections
show subscriber (msisdn|extension|imsi|tmsi|id) ID
show subscriber cache
show bsc
show connection
show transaction
sms send pending
sms delete expired
subscriber create imsi ID
subscriber (msisdn|extension|imsi|tmsi|id) ID sms sender (msisdn|extension|imsi|tmsi|id)  ↩
    SENDER_ID send .LINE
subscriber (msisdn|extension|imsi|tmsi|id) ID silent-sms sender (msisdn|extension|imsi| ↩
    tmsi|id) SENDER_ID send .LINE
subscriber (msisdn|extension|imsi|tmsi|id) ID silent-call start (any|tch/f|tch/any|sdcch)
subscriber (msisdn|extension|imsi|tmsi|id) ID silent-call stop
subscriber (msisdn|extension|imsi|tmsi|id) ID ussd-notify (0|1|2) .TEXT
subscriber (msisdn|extension|imsi|tmsi|id) ID ms-test close-loop (a|b|c|d|e|f|i)
subscriber (msisdn|extension|imsi|tmsi|id) ID ms-test open-loop
subscriber (msisdn|extension|imsi|tmsi|id) ID paging
show statistics
show sms-queue
logging filter imsi IMSI
show smpp esme
```

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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 implemets a system of attribute flags, which can be displayed using the `show` command with the parameter `vty-attributes`

Example: Typing show vty-attributes at the VTY prompt

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

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

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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

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

Example: Typing list with-flags at the VTY prompt

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

❶ This command has no attributes assigned.

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

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

❹, ❺ This command applies immediately.

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

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

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

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

reduces the risk of an accidental invocation and potential service degradation; on the other, it complicates intentional use of the hidden commands.

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

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

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

```
OsmoBSC> enable expert-mode ❶
OsmoBSC# list with-flags
  ...
  ^   bts <0-255> (activate-all-lchan|deactivate-all-lchan) ❷
  ^   bts <0-255> trx <0-255> (activate-all-lchan|deactivate-all-lchan) ❸
  .   bts <0-255> trx <0-255> timeslot <0-7> sub-slot <0-7> mdcx A.B.C.D <0-65535> ❹
  ^   bts <0-255> trx <0-255> timeslot <0-7> sub-slot <0-7> (borken|unused) ❺
  .   bts <0-255> trx <0-255> timeslot <0-7> sub-slot <0-7> handover <0-255> ❻
  .   bts <0-255> trx <0-255> timeslot <0-7> sub-slot <0-7> assignment ❼
  .   bts <0-255> smscb-command (normal|schedule|default) <1-4> HEXSTRING ❽
  ...
```

❶ This command enables the *expert* mode.

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

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

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 list of categories interactively on the vty, type: `logging level ?`

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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

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

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

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

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

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

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

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

7.5.3 Logging via gsmtap

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

GSMTAP logging is now a method how Osmocom software can also encapsulate its own log output in GSMTAP frames. We're not trying to re-invent rsyslog here, but this is very handy When debugging complex issues. It enables the reader of the pcap file containing GSMTAP logging together with other protocol traces to reconstruct exact chain of events. A single pcap file can then contain both the log output of any number of Osmocom programs in the same timeline of the messages on various interfaces in and out of said Osmocom programs.

It's configured as follows:

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

❶ Configure a GSMTAP logging target recording to 192.168.2.3

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.

The GSMTAP logging target also has different implementation I/O modes, which can be configured during the VTY "log gmstap" VTY command:

nonblocking-io
: Use non-blocking, synchronous I/O. This is the default mode. It will set the GSMTAP source socket as non-blocking, and attempt synchronous write to the UDP socket, discarding a message if the UDP socket's send buffer is full during write.

blocking-io
: Use blocking, synchronous I/O. It will use the GSMTAP source socket in blocking mode, and attempt synchronous write to the UDP socket, blocking the entire thread if the UDP socket's send buffer is full during write. This I/O mode should not be used in any osmocom program driven by an event loop (which includes any program using a VTY), since it may block the thread for an unspecified (potentially long or even infinite) amount of time and hence prevent the event loop to run and process events. This I/O mode should only be used under specific circumstances, like debugging a crash, in which case it's interesting to get the logging output to the UDP socket and hence recorded before the program crashes.

wq
: Workqueue mode: Use non-blocking, asynchronous I/O. It will set the GSMTAP source socket as non-blocking, and attempt asynchronous write to the UDP socket by means of a work queue driven by the process poll event loop, delaying writing to the UDP socket until there's enough free space in its send buffer. Since the workqueue has a maximum size, messages will be discarded similar to *nonblocking-io* once the UDP socket's send buffer **and** the GSMTAP log target workqueue are full. This I/O mode provides some extra buffering space in the userspace workqueue, at the expense of extra overhead and delay to transmit messages asynchronously waiting for POLLOUT event from the UDP socket.

In any of the I/O modes above, the UDP socket send buffer size of the GSMTAP log target influences where messages may be dropped (*nonblocking-io*|*wq*) or the process blocked (*blocking-io*) whenever a high number of messages or test are transmitted in a short time. For that reason, the GSMTAP log target will attempt to set the `SO_SNDBUF` sockopt of the UDP socket to 4MB. In Linux, the `SO_SNDBUF` value being set on the socket will be trimmed by the kernel to a maximum allowed by sysctl `net.core.wmem_max`. Hence, if you want to be on a safer side minimizing dropping messages or blocking the process (depending on the I/O mode set), make sure sysctl `net.core.wmem_max` on your system is set to `4194304` (4MB) or more.

Note
Linux Kernel used to have a `net.core.wmem_max` of `212992`(208KB) until recently (2025, a6d4f25888b83b8300aef28d9ee22765c1cc9b34), where it was increased to `4194304` (4MB) .

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

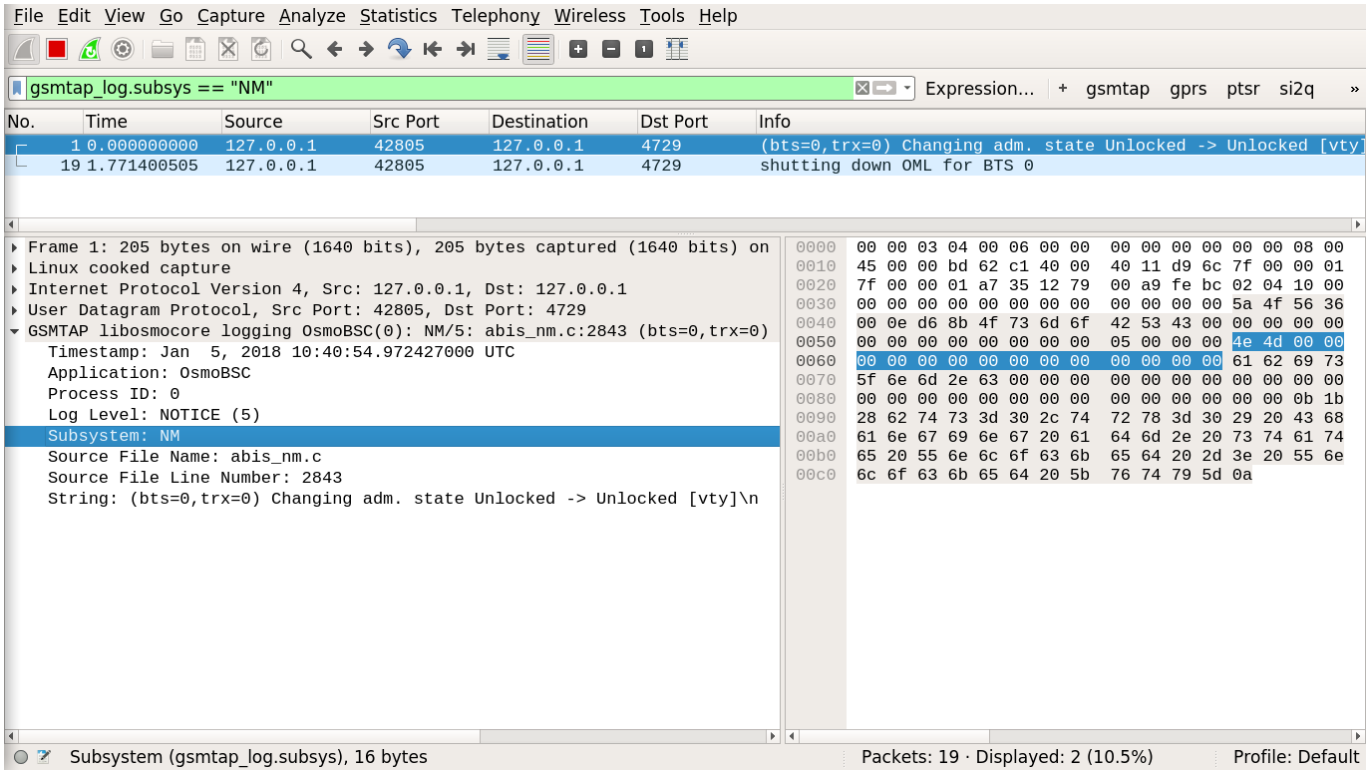


Figure 3: Wireshark with logs delivered over GSMTAP

Note

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

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 ❶
OsmoBSC(config-log)#
```

❶ Configure a file logging target writing to /path/to/my/file

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`.

The file logging target will by default use a synchronous non-blocking I/O write. If writing cannot be performed synchronously (would block), the message is stored in a workqueue where asynchronous write (driven by poll event loop) will be performed at a later stage. The file logging target can be configured during the VTY `log file` VTY command to perform blocking I/O with the `blocking-io` parameter. This I/O mode should not be used in any osmocom program driven by an event loop (which includes any program using a VTY), since it may block the thread for an unspecified (potentially long or even infinite) amount of time and hence prevent the event loop to run and process events. This I/O mode should only be used under specific circumstances, like debugging a crash, in which case it's interesting to get the logging output to the UDP socket and hence recorded before the program crashes.

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.

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

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

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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

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

You can configure systemd based logging in two ways:

Example: systemd-journal target with offloaded rendering

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

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

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

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

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

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

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

Example: systemd-journal target with libosmocore based rendering

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

❶ Generic logging handler, rendering is done by libosmocore.

❷ Disable timestamping, systemd will timestamp every message anyway.

❸ 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*.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

Note
systemd-journal logging target uses libsystemd, which implements *Native Journal Protocol* (*AF_UNIX _DATAGRAM* socket). That socket is internally configured in synchronous blocking I/O mode, which means transmitting a big amount of messages or text over it in a short amount of time (before systemd-journald has time to read it) could potentially block the osmocom program's thread issuing the logging. libsystemd currently sets the *AF_UNIX* socket `SO_SNDBUF` to 8MB. In Linux, the `SO_SNDBUF` value being set on the socket will be trimmed by the kernel to a maximum allowed by sysctl `net.core.wmem_max`. Hence, if you want to be on a safer side minimizing dropping messages or blocking the process (depending on the I/O mode set), make sure sysctl `net.core.wmem_max` on your system is set to `8388608` (8MB) or more. Besides, the kernel limits the amount of messages queued in *AF_UNIX* sockets to value set in sysctl `net.unix.max_dgram_qlen`, so you may also want to increase that one to be on the safe side if planning to log with a high frequency.

Note
Linux Kernel used to have a `net.core.wmem_max` of `212992`(208KB) until recently (2025, a6d4f25888b83b8300aef28d9ee22765c1cc9b34), where it was increased to `4194304` (4MB) .

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)#
```

The stderr logging target can be configured during the VTY `log stderr` VTY command to perform blocking I/O with the `blocking-io` parameter. See Section 7.5.4 for more information.

8 Configuring OsmoPCU

Contrary to other network elements (like OsmoBSC, OsmoNITB), the OsmoPCU has a relatively simple minimum configuration.

This is primarily because most of the PCU configuration happens indirectly from the BSC, who passes the configuation over A-bis OML via OsmoBTS and its PCU socket into OsmoPCU.

A minimal OsmoPCU configuration file is provided below for your reference:

Example: Minimal OsmoPCU configuration file (osmo-pcu.cfg)

```
pcu
 flow-control-interval 10 ❶
 cs 2 ❷
 alloc-algorithm dynamic ❸
 gamma 0
```

❶ send a BSSGP flow-control PDU every 10 seconds

❷ start a TBF with the initial coding scheme 2

❸ dynamically chose between single-slot or multi-slot TBF allocations depending on system load

However, there are plenty of tuning parameters for people interested to optimize PCU throughput or latency according to their requirements.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

8.1 Configuring the Coding Schemes and Rate Adaption

As a reminder:

- GPRS supports Coding Schemes 1-4 (CS1-4), all of them use GMSK modulation (same as GSM).
- EGPRS supports MCS1-9, where MCS1-4 is GMSK, and MCS5-9 use 8-PSK modulation.

The range of Coding Schemes above only apply to RLCMAC data blocks; RLCMAC control blocks are always transmitted with CS1 (GMSK). Hence, CS1 is always supported and must be always permitted.

The BSC includes a bit-mask of permitted [E]GPRS coding schemes as part of the A-bis OML configuration, controlled by VTY `gprs mode (none|gprs|egprs)`. This is passed from the BTS via the PCU socket into OsmoPCU, and the resulting set can be further constrained by OsmoPCU VTY configuration.

Some additional OsmoPCU parameters can be set as described below.

8.1.1 Initial Coding Scheme

You can use the `cs <1-4> [<1-4>]` command at the `pcu` VTY config node to set the initial GPRS coding scheme to be used. The optional second value allows to specify a different initial coding scheme for uplink.

Similarly, `mcs <1-9> [<1-9>]` can be used to set up the initial EGPRS coding scheme.

8.1.2 Maximum Coding Scheme

You can use the `cs max <1-4> [<1-4>]` command at the `pcu` VTY config node to set the maximum GPRS coding scheme that should be used as part of the rate adaption. The optional second value allows to specify a different maximum coding scheme for uplink.

Similarly, `mcs max <1-9> [<1-9>]` can be used to set up the maximum EGPRS coding scheme.

The actual Maximum Coding Scheme for each direction used at runtime is actually the result of taking the maximum value from the permitted GPRS coding schemes received by the BSC (or BTS) over PCUIF which is equal or lower tho the configured value.

Example: PCUIF announces permitted MCS bit-mask (`MCS1 MCS2 MCS3 MCS4`) and OsmoPCU is configured `mcs max 6`, then the actual maximum MCS used at runtime will be `MCS4`.

8.1.3 Rate Adaption Error Thresholds

You can use the `cs threshold <0-100> <0-100>` command at the `pcu` VTY config node to determine the upper and lower limit for the error rate percentage to use in the rate adaption. If the upper threshold is reached, a lower coding sheme is chosen, and if the lower threshold is reached, a higher coding scheme is chosen.

8.1.4 Rate Adation Link Quality Thresholds

You can use the `cs link-quality-ranges cs1 <0-35> cs2 <0-35> <0-35> cs3 <0-35> <0-35> cs4 <0-35>` command at the `pcu` VTY config node to tune the link quality ranges for the respective coding schemes.

8.1.5 Data Size based CS downgrade Threshold

You can use the `cs downgrade-threshold <1-10000>` command at the `pcu` VTY config node to ask the PCU to down-grade the coding scheme if less than the specified number of octets are left to be transmitted.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

8.2 Miscellaneous Configuration / Tuning Parameters

8.2.1 Downlink TBF idle time

After a down-link TBF is idle (all data in the current LLC downlink queue for the MS has been transmitted), we can keep the TBF established for a configurable time. This avoids having to go through a new one or two phase TBF establishment once the next data for downlink arrives.

You can use the `dl-tbf-idle-time <1-5000>` to specify that time in units of milli-seconds. The default is 2 seconds.

8.2.2 MS idle time

Using the `ms-idle-time <1-7200>` command at the `pcu` VTY config node you can configure the number of seconds for which the PCU should keep the MS data structure alive before releasing it if there are no active TBF for this MS.

The OsmoPCU default value is 60 seconds, which is slightly more than what 3GPP TS 24.008 recommends for T3314 (44s).

The MS data structure only consumes memory in the PCU and does not require any resources of the air interface.

8.2.3 Forcing two-phase access

If the MS is using a single-phase access, you can still force it to use a two-phase access using the `two-phase-access` VTY configuration command at the `pcu` VTY config node.

8.3 Configuring BSSGP flow control

BSSGP between SGSN and PCU contains a two-level nested flow control mechanism:

1. one global flow control instance for the overall (downlink) traffic from the SGSN to this PCU
2. a per-MS flow control instance for each individual MS served by this PCU

Each of the flow control instance is implemented as a TBF (token bucket filter).

8.3.1 Normal BSSGP Flow Control Tuning parameters

You can use the following commands at the `pcu` VTY config node to tune the BSSGP flow control parameters:

`flow-control-interval <1-10>`
configure the interval (in seconds) between subsequent flow control PDUs from PCU to SGSN

`flow-control bucket-time <1-65534>`
set the target downlink maximum queueing time in centi-seconds. The PCU will attempt to adjust the advertised bucket size to match this target.

8.3.2 Extended BSSGP Flow Control Tuning parameters

There are some extended flow control related parameters at the `pcu` VTY config node that override the automatic flow control as specified in the BSSGP specification. Use them with care!

`flow-control force-bvc-bucket-size <1-6553500>`
force the BVC (global) bucket size to the given number of octets

`flow-control force-bvc-leak-rate <1-6553500>`
force the BVC (global) bucket leak rate to the given number of bits/s

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

flow-control force-ms-bucket-size <1-6553500>
force the per-MS bucket size to the given number of octets

flow-control force-ms-leak-rate <1-6553500>
force the per-MS bucket leak rate to the given number of bits/s

8.4 Configuring LLC queue

The downlink LLC queue in the PCU towards the MS can be tuned with a variety of parameters at the `pcu` VTY config node, depending on your needs.

queue lifetime <1-65534>
Each downlink LLC PDU is assigned a lifetime by the SGSN, which is respected by the PDU **unless** you use this command to override the PDU lifetime with a larger value (in centi-seconds)

queue lifetime infinite
Never drop LLC PDUs, i.e. give them an unlimited lifetime.

queue hysteresis <1-65535>
When the downlink LLC queue is full, the PCU starts dropping packets. Using this parameter, we can set the lifetime hysteresis in centi-seconds, i.e. it will continue discarding until "lifetime - hysteresis" is reached.

queue codel
Use the *CoDel* (Controlled Delay) scheduling algorithm, which is designed to overcome buffer bloat. It will use a default interval of 4 seconds.

queue codel interval <1-1000>
Use the *CoDel* (Controlled Delay) scheduling algorithm, which is designed to overcome buffer bloat. Use the specified interval in centi-seconds.

queue idle-ack-delay <1-65535>
Delay the request for an ACK after the last downlink LLC frame by the specified amount of centi-seconds.

8.5 Configuring MS power control

GPRS MS power control works completely different than the close MS power control loop in circuit-switched GSM.

Rather than instructing the MS constantly about which transmit power to use, some parameters are provided to the MS by which the MS-based power control algorithm is tuned.

See 3GPP TS 05.08 for further information on the algorithm and the parameters.

You can set those parameters at the `pcu` VTY config node as follows:

gamma <0-62>
Set the gamma parameter for MS power control in units of dB.

Parameter `ALPHA` is set on the BSC VTY configuration file on a per-BTS basis, and forwarded by OsmoPCU to the MS through the SI13 received from the former over PCUIF. OsmoPCU VTY command `alpha <0-10>` overrides the value received over PCUIF to keep backward compatibility with existing config files, but it is currently deprecated and its use is discouraged; one should configure it per-BTS in OsmoBSC VTY instead.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

8.6 Configuring Network Assisted Cell Change (NACC)

Network Assisted Cell Change, defined in 3GPP TS 44.060 sub-clause 8.8, is a feature providing the MS aid when changing to a new cell due to autonomous reselection. In summary, the MS informs the current cell its intention to change to a new target cell, and the network decides whether to grant the intended cell change or order a change to another neighbor cell. It also provides several System Informations of the target cell to the MS to allow for quicker reselection towards it.

OsmoPCU will automatically provide the required neighbor System Information when the MS requests NACC towards a target cell also under the management of the same OsmoPCU instance, since it already has the System Information of all BTS under their control, obtained through PCUIF when the BTS registers against OsmoPCU, so no specific user configuration is required here.

In general, OsmoPCU requires to gather the information from somewhere else before being able to provide it to the MS requesting the NACC.

If OsmoPCU fails to gather the System Information, it will simply answer the MS allowing the proposed changed but without previously providing the System Information of the target cell.

8.6.1 Neighbor Address Resolution

First of all, it needs to translate the <ARFCN + BSIC> identity of the target cell to change to, provided by the MS, into an identity that the Core Network can use and understand to identify the target cell, which happens to be a key composed of <RAI + Cell Identity>. This key is also named conveniently as CGI-PS, since it actually equals to the Circuit Switch CGI + RAC.

In order to apply this target cell identity translation, OsmoPCU uses the OsmoBSC Neighbor Resolution Service. This service is nowadays provided by means of PCUIF container messages, which are transparently forwarded in both directions by the BTS using the IPA multiplex of the OML connection against the BSC. No specific configuration is required in any of the involved nodes, they should behave properly out of the box.

These neighbor address resolutions (<ARFCN + BSIC> ⇒ <RAI + CI>) are by default cached for a while, in order to avoid querying the BSC frequently. As a result, the resolution time is also optimized.

Example: Configure Neighbor Resolution cache and timeouts

```
pcu
 timer X1 500 ❶
 timer X0 60 ❷
```

❶ Time out if the BSC doesn't answer our resolution request after 500 ms

❷ Keep resolved neighbor addresses cached for 60 seconds

8.6.1.1 OsmoBSC CTRL interface (deprecated)



Caution
This interface is nowadays considered deprecated and should not be used anymore. Any related VTY options should be dropped from configuration files, to let OsmoPCU use the new interface instead. This section is kept here for a while as a reference for old deployments using old versions of the programs.

This Neighbor Address Resolution Service was initially implemented by means of a separate CTRL interface (see OsmoBSC User Manual), where OsmoPCU would create a CTRL connection to the BSC each time an address resolution was required.

Older versions of OsmoBSC may not support the current Neighbor Address Resolution Service over the IPA multiplex (see above). For those cases, OsmoPCU can be configured to use the old deprecated CTRL interface.

By default, the use of this interface is not configured and hence disabled in OsmoPCU. As a result, until configured, the network won't be able to provide the System Information to the MS prior to allowing the change during NACC against remote cells, which means the cell change will take longer to complete. In order to configure the interface, the OsmoBSC IP address and port to connect to must be configured in OsmoPCU VTY.

Example: Configure Neighbor Resolution CTRL interface against OsmoBSC

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

```
pcu
 neighbor resolution 172.18.13.10 4248 ❶
```

❶ Port 4248 is the default and hence could be omitted in this case

8.6.2 System Information Resolution

Once OsmoPCU gains knowledge of the target cell's address in the Core Network, it can query its System Information.

OsmoPCU will gather the requested System Information of target cells under its control without need for any external query, since the System Information of all BTSs it manages are received over PCUIF and stored internally in OsmoPCU.

For those targets cells not managed by the OsmoPCU instance, the query is accomplished by using RIM procedures (NACC RAN-INFO application) over the Gb interface against the SGSN that OsmoPCU is connected to. In its turn, the SGSN will potentially forward this query to the PCU serving the target cell, which will provide back the System Information of that cell.

The System Information received from external PCUs over RIM are by default cached for a while in order to avoid querying the SGSN frequently and, as a result, optimizing the resolution time too.

Example: Configure System Information resolution

```
pcu
 timer X2 500 ❶
 timer X11 60 ❷
```

❶ Time out if the SGSN doesn't answer our RIM RAN-INFO request request after 500 ms

❷ Keep resolved remote neighbor System Information cached for 60 seconds

8.7 Configuring E1 line for CCU access

Depending on the configuration the PCU may require direct access to a BTS CCU (channel coding unit) via an E1 line. This is in particular the case when OsmoPCU runs in co-location with OsmoBSC.

The exact timeslot configuration is passed to the PCU via the pcu_sock interface. Only basic E1 line settings are required. However, it is important that the E1 line number is the same as the E1 line number that is used in the timeslot configuration of OsmoBSC.

Example: Configure an E1 line

```
e1_input
 e1_line 0 driver dahdi
 e1_line 0 port 2
 no e1_line 0 keepalive
```

8.8 GPRS vs EGPRS considerations

8.8.1 Configuration

OsmoPCU can be configured to either:

- Allocate only GPRS TBFs to all MS (no EGPRS)
- Allocate EGPRS TBFs to EGPRS capable phones while still falling back to allocating GPRS TBFs on GPRS-only capable MS.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

These two different modes of operation are selected by properly configuring the Coding Schemes (see Section 8.1.2).

The first mode of operation (GPRS-only for all MS) can be accomplished configuring OsmoPCU so that the resulting MCS set is empty. This can be done in two ways:

- Announcing an empty MCS bit-mask over PCUIF to OsmoPCU: That's actually done automatically by OsmoBSC on BTS with VTY config set to `gprs mode gprs`.
- Configuring OsmoPCU to force an empty set by using VTY command `mcs max 0`.

Hence, if the resulting MCS bit-mask is not empty, (BSC configuring the BTS with `gprs mode egprs` and OsmoPCU VTY containing something other than *mcs max 0*), EGPRS TBFs will be allocated for all MS announcing EGPRS capabilities.

It is important to remark that in order to use MCS5-9, the BTS must support 8-PSK modulation. Nevertheless, in case 8-PSK is not supported by the BTS, one can still enable EGPRS and simply make sure 8-PSK MCS are never used by configuring OsmoPCU with `mcs max 4 4`.

Similarly, a BTS may support 8-PSK modulation only on downlink, since it is easier to implement than the uplink, together with the fact that higher downlink throughput is usually more interesting from user point of view. In this scenario, OsmoPCU can be configured to allow for full MCS range in downlink while still preventing use of 8-PSK on the uplink: `mcs max 9 4`.

Some other interesting configurations to control use of EGPRS in the network which lay outside OsmoPCU include:

- If `osmo-bts-trx` together with `osmo-trx` is used, remember to enable EGPRS support (OsmoTRX VTY `egprs enable`).
- It is possible to improve EGPRS performance (in particular, the TBF establishment timing) a bit by enabling 11-bit Access Burst support. This allows EGPRS capable phones to indicate their EGPRS capability, establishment cause, and multi-slot class directly in the Access Burst (OsmoTRX VTY `ext-rach enable`, OsmoBSC VTY `gprs egprs-packet-channel-reques`

Note
If you enable MCS5-9 you will also need an 8-PSK capable OsmoBTS+PHY, which means `osmo-bts-sysmo` or `osmo-bts-litecell15` with their associated PHY, or `osmo-bts-trx` with `osmo-trx` properly configured.

8.8.2 GPRS+EGPRS multiplexing

Both EGPRS and GPRS-only capable MS can be driven concurrently in the same PDCH timeslot by the PCU, hence no special configuration is required per timeslot regarding this topic; OsmoPCU scheduler takes care of the specific requirements when driving MS with different capabilities.

These specific requirements translate to some restrictions regarding which Coding Schemes can be used at given frame numbers, and hence which kind of RLCMAC blocks can be sent, which means serving a GPRS-only MS in a PDCH may end up affecting slightly the downlink throughput of EGPRS capable MS.

Throughput loss based on MS capabilities with TBF attached to a certain PDCH timeslot:

All UEs are EGPRS capable
No throughput loss, since all data is sent using EGPRS, and EGPRS control messages are used when appropriate.

All UEs are GPRS-only (doesn't support EGPRS)
No throughput loss, since all data and control blocks use GPRS.

Some UEs are GPRS-only, some EGPRS
In general EGPRS capable UEs will use EGPRS, and GPRS-only UEs will use GPRS, with some restrictions affecting throughput of EGPRS capable UEs:

- Whenever a GPRS-only MS is to be polled to send uplink data to PCU, then a downlink RLCMAC block modulated using GMSK must be sent, which means that if the scheduler selects a EGPRS MS for downlink on that block it will force sending of data with MCS1-4 (if it's new data, if it's a retransmission it cannot be selected since MCS from original message cannot be changed). In the worst case if no control block needs to be sent or no new data in MCS1-4 is available to send, then an RLCMAC Dummy Block is sent.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

- For synchronization purposes, each MS needs to receive an RLCMAC block which it can fully decode at least every 360ms, which means the scheduler must enforce a downlink block in CS1-4 every 360ms, that is, every 18th RLCMAC block. In general this is not a big issue since anyway all control RLCMAC blocks are encoded in CS1, so in case any control block is sent from time to time it's accomplished and there's no penalty. However, if only EGPRS downlink data is sent over that time frame, then the scheduler will force sending a RLCMAC Dummy Block.

8.9 Configuring GSMTAP tracing

In addition to being able to obtain pcap protocol traces of the NS/BSSGP communication and the text-based logging from the OsmoPCU software, there is also the capability of tracing all communication on the radio interface related to PS. To do so, OsmoPCU can encapsulate MAC blocks (23-155 byte messages at the L2-L1 interface depending on coding scheme) into *GSMTAP* and send them via UDP/IP. At that point, they can be captured with utilities like **tcpdump** or **tshark** for further analysis by the **wireshark** protocol analyzer.

In order to activate this feature, you first need to make sure to specify the remote address of *GSMTAP* host in the configuration file. In most cases, using 127.0.0.1 for passing the messages over the loopback (`lo`) device will be sufficient:

Example: Enabling GSMTAP Um-frame logging to localhost

```
pcu
 gsmtap-remote-host 127.0.0.1 ❶
```

❶ Destination address for *GSMTAP* Um-frames

Note
Changing this parameter at run-time will not affect the existing *GSMTAP* connection, full program restart is required.

Note
Command line parameters `-i` and `--gsmtap-ip` have been deprecated.

OsmoPCU can selectively trace such messages based on different categories, for both Ul and Dl. For a complete list of cateogry values, please refer to the *OsmoPCU VTY reference manual* [vty-ref-osmopcu].

For example, to enable GSMTAP tracing for all DL EGPRS rlcmac data blocks, you can use the `gsmtap-category dl-data-egpr` command at the `pcu` node of the OsmoPCU VTY.

Example: Enabling GSMTAP for for all DL EGPRS rlcmac data blocks

```
OsmoPCU> enable
OsmoPCU# configure terminal
OsmoPCU(config)# pcu
OsmoPCU(pcu)# gsmtap-category dl-data-egprs
OsmoPCU(trx)# write ❶
```

❶ the `write` command will make the configuration persistent in the configuration file. This is not required if you wish to enable GSMTAP only in the current session of OsmoPCU.

De-activation can be performed similarly by using the `no gsmtap-category dl-data-egprs` command at the `pcu` node of the OsmoPCU VTY.

It may be useful to enable all categories with a few exceptions, or vice versa disable everything using one command. For this purpose, the VTY provides `gsmtap-category enable-all` and `gsmtap-category disable-all` commands.

Example: Enabling all categoriess except dl-dummy

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

```
pcu
 gsmtap-category enable-all ❶
 no gsmtap-category dl-dummy ❷
```

❶ Enable all available SAPIs

❷ Exclude DL RLCMAC blocks

From the moment they are enabled via VTY, GSMTAP messages will be generated and sent in UDP encapsulation to the IANA-registered UDP port for GSMTAP (4729) of the specified remote address.

9 Counters

These counters and their description are based on OsmoPCU 0.9.0.244-de96 (OsmoPCU).

9.1 Rate Counters

Table 2: ns:nsvc - NSVC Peer Statistics

Name	Reference	Description
packets:in	[?]	Packets at NS Level (In)
packets:out	[?]	Packets at NS Level (Out)
packets:out:drop	[?]	Dropped Packets (Out)
bytes:in	[?]	Bytes at NS Level (In)
bytes:out	[?]	Bytes at NS Level (Out)
bytes:out:drop	[?]	Dropped Bytes (Out)
blocked	[?]	NS-VC Block count
unblocked	[?]	NS-VC Unblock count
dead	[?]	NS-VC gone dead count
replaced	[?]	NS-VC replaced other count
nsei-chg	[?]	NS-VC changed NSEI count
inv-nsvci	[?]	NS-VCI was invalid count
inv-nsei	[?]	NSEI was invalid count
lost:alive	[?]	ALIVE ACK missing count
lost:reset	[?]	RESET ACK missing count

Table 3: ns:nse - NSE Peer Statistics

Name	Reference	Description
packets:in	[?]	Packets at NS Level (In)
packets:out	[?]	Packets at NS Level (Out)
packets:out:drop	[?]	Dropped Packets (Out)
bytes:in	[?]	Bytes at NS Level (In)
bytes:out	[?]	Bytes at NS Level (Out)
bytes:out:drop	[?]	Dropped Bytes (Out)
blocked	[?]	NS-VC Block count
unblocked	[?]	NS-VC Unblock count
dead	[?]	NS-VC gone dead count
replaced	[?]	NS-VC replaced other count

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

Table 3: (continued)

Name	Reference	Description
nsei-chg	[?]	NS-VC changed NSEI count
inv-nsvci	[?]	NS-VCI was invalid count
inv-nsei	[?]	NSEI was invalid count
lost:alive	[?]	ALIVE ACK missing count
lost:reset	[?]	RESET ACK missing count

Table 4: pcu:sgsn - SGSN Statistics

Name	Reference	Description
rx_paging_cs	[?]	Amount of paging CS requests received
rx_paging_ps	[?]	Amount of paging PS requests received

Table 5: bssgp:bss_ctx - BSSGP Peer Statistics

Name	Reference	Description
packets:in	[?]	Packets at BSSGP Level (In)
packets:out	[?]	Packets at BSSGP Level (Out)
bytes:in	[?]	Bytes at BSSGP Level (In)
bytes:out	[?]	Bytes at BSSGP Level (Out)
blocked	[?]	BVC Blocking count
discarded	[?]	BVC LLC Discarded count
status	[?]	BVC Status count

Table 6: bts - BTS Statistics

Name	Reference	Description
tbf:dl:alloc	[?]	TBF DL Allocated
tbf:dl:freed	[?]	TBF DL Freed
tbf:dl:aborted	[?]	TBF DL Aborted
tbf:ul:alloc	[?]	TBF UL Allocated
tbf:ul:freed	[?]	TBF UL Freed
tbf:ul:aborted	[?]	TBF UL Aborted
tbf:reused	[?]	TBF Reused
tbf:alloc:algo-a	[?]	TBF Alloc Algo A
tbf:alloc:algo-b	[?]	TBF Alloc Algo B
tbf:alloc:failed	[?]	TBF Alloc Failure (any reason)
tbf:alloc:failed:no_tfi	[?]	TBF Alloc Failure (TFIs exhausted)
tbf:alloc:failed:no_usf	[?]	TBF Alloc Failure (USFs exhausted)
tbf:alloc:failed:no_slot_combi	[?]	TBF Alloc Failure (No valid UL/DL slot combination found)
tbf:alloc:failed:no_slot_avail	[?]	TBF Alloc Failure (No slot available)
rlc:sent	[?]	RLC Sent

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

Table 6: (continued)

Name	Reference	Description
rlc:resent	[?]	RLC Resent
rlc:restarted	[?]	RLC Restarted
rlc:stalled	[?]	RLC Stalled
rlc:nacked	[?]	RLC Nacked
rlc:final_block_resent	[?]	RLC Final Blk resent
rlc:ass:timedout	[?]	RLC Assign Timeout
rlc:ass:failed	[?]	RLC Assign Failed
rlc:ack:timedout	[?]	RLC Ack Timeout
rlc:ack:failed	[?]	RLC Ack Failed
rlc:rel:timedout	[?]	RLC Release Timeout
rlc:late-block	[?]	RLC Late Block
rlc:sent-dummy	[?]	RLC Sent Dummy
rlc:sent-control	[?]	RLC Sent Control
rlc:dl_bytes	[?]	RLC DL Bytes
rlc:dl_payload_bytes	[?]	RLC DL Payload Bytes
rlc:ul_bytes	[?]	RLC UL Bytes
rlc:ul_payload_bytes	[?]	RLC UL Payload Bytes
decode:errors	[?]	Decode Errors
sba:allocated	[?]	SBA Allocated
sba:freed	[?]	SBA Freed
sba:timedout	[?]	SBA Timeout
llc:timeout	[?]	Timedout Frames
llc:dropped	[?]	Dropped Frames
llc:scheduled	[?]	Scheduled Frames
llc:dl_bytes	[?]	RLC encapsulated PDUs
llc:ul_bytes	[?]	full PDUs received
pch:requests	[?]	PCH requests sent
pch:requests:timeout	[?]	PCH requests timeout
rach:requests	[?]	RACH requests received
rach:requests:11bit	[?]	11BIT_RACH requests received
rach:requests:one_phase	[?]	One phase packet access with request for single TS UL
rach:requests:two_phase	[?]	Single block packet request for two phase packet access
rach:requests:unexpected	[?]	RACH Request with unexpected content received
spb:uplink_first_segment	[?]	First seg of UL SPB
spb:uplink_second_segment	[?]	Second seg of UL SPB
spb:downlink_first_segment	[?]	First seg of DL SPB
spb:downlink_second_segment	[?]	Second seg of DL SPB
immediate:assignment_UL	[?]	Immediate Assign UL
immediate:assignment_ul:one_phase	[?]	Immediate Assign UL (one phase packet access)
immediate:assignment_ul:two_phase	[?]	Immediate Assign UL (two phase packet access)
immediate:assignment_ul:contention_resolution_success	[?]	First RLC Block (PDU) on the PDTCH from the MS received
immediate:assignment_rej	[?]	Immediate Assign Rej
immediate:assignment_DL	[?]	Immediate Assign DL
channel:request_description	[?]	Channel Request Desc
pkt:ul_assignment	[?]	Packet UL Assignment
pkt:access_reject	[?]	Packet Access Reject
pkt:dl_assignment	[?]	Packet DL Assignment
pkt:cell_chg_notification	[?]	Packet Cell Change Notification

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

Table 6: (continued)

Name	Reference	Description
pkt:cell_chg_continue	[?]	Packet Cell Change Continue
pkt:neigh_cell_data	[?]	Packet Neighbour Cell Data
ul:control	[?]	UL control Block
ul:assignment_poll_timeout	[?]	UL Assign Timeout
ul:assignment_failed	[?]	UL Assign Failed
dl:assignment_timeout	[?]	DL Assign Timeout
dl:assignment_failed	[?]	DL Assign Failed
pkt:ul_ack_nack_timeout	[?]	PUAN Poll Timeout
pkt:ul_ack_nack_failed	[?]	PUAN poll Failed
pkt:dl_ack_nack_timeout	[?]	PDAN poll Timeout
pkt:dl_ack_nack_failed	[?]	PDAN poll Failed
gprs:downlink_cs1	[?]	CS1 downlink
gprs:downlink_cs2	[?]	CS2 downlink
gprs:downlink_cs3	[?]	CS3 downlink
gprs:downlink_cs4	[?]	CS4 downlink
egprs:downlink_mcs1	[?]	MCS1 downlink
egprs:downlink_mcs2	[?]	MCS2 downlink
egprs:downlink_mcs3	[?]	MCS3 downlink
egprs:downlink_mcs4	[?]	MCS4 downlink
egprs:downlink_mcs5	[?]	MCS5 downlink
egprs:downlink_mcs6	[?]	MCS6 downlink
egprs:downlink_mcs7	[?]	MCS7 downlink
egprs:downlink_mcs8	[?]	MCS8 downlink
egprs:downlink_mcs9	[?]	MCS9 downlink
gprs:uplink_cs1	[?]	CS1 Uplink
gprs:uplink_cs2	[?]	CS2 Uplink
gprs:uplink_cs3	[?]	CS3 Uplink
gprs:uplink_cs4	[?]	CS4 Uplink
egprs:uplink_mcs1	[?]	MCS1 Uplink
egprs:uplink_mcs2	[?]	MCS2 Uplink
egprs:uplink_mcs3	[?]	MCS3 Uplink
egprs:uplink_mcs4	[?]	MCS4 Uplink
egprs:uplink_mcs5	[?]	MCS5 Uplink
egprs:uplink_mcs6	[?]	MCS6 Uplink
egprs:uplink_mcs7	[?]	MCS7 Uplink
egprs:uplink_mcs8	[?]	MCS8 Uplink
egprs:uplink_mcs9	[?]	MCS9 Uplink

9.2 Osmo Stat Items

NSVC Peer Statistics .ns.nsvc - NSVC Peer Statistics

Name	Reference	Description	Unit
alive.delay	[?]	ALIVE response time	ms

NS Bind Statistics .ns.bind - NS Bind Statistics

Name	Reference	Description	Unit
tx_backlog_length	[?]	Transmit backlog length	packets

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

BTS Statistics .bts - BTS Statistics

Name	Reference	Description	Unit
ms.present	[?]	MS Present	
pdch.available	[?]	PDCH available	
pdch.occupied	[?]	PDCH occupied (all)	
pdch.occupied.gprs	[?]	PDCH occupied (GPRS)	
pdch.occupied.egprs	[?]	PDCH occupied (EGPRS)	

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

10 Code Architecture

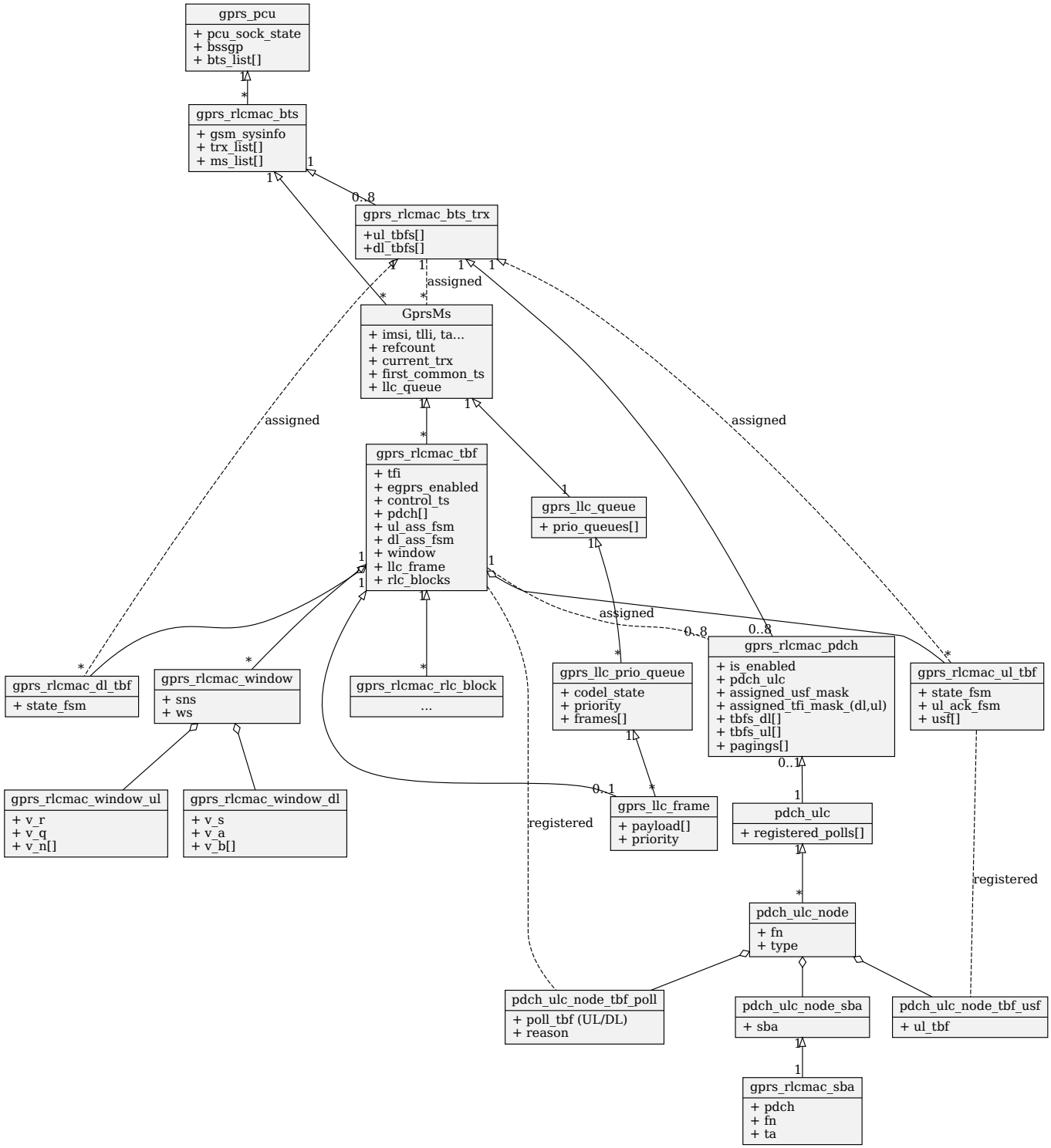


Figure 4: General overview of main OsmoTRX components

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

11 Gb interface using libosmogb

libosmogb is part of the libosmocore.git repository and implements the Gb interface protocol stack consisting of the NS and BSSGP layers. It is used in a variety of Osmocom project, including OsmoSGSN, OsmoGbProxy and OsmoPCU.

This section describes the configuration that libosmogb exposes via the VTY.

11.1 Gb interface configuration

11.1.1 NS-over-UDP configuration

The GPRS-NS protocol can be encapsulated in UDP/IP. This is the default encapsulation for IP based GPRS systems.

Example: GPRS NS-over-UDP configuration

```
OsmoSGSN(config-ns)# encapsulation udp local-ip 127.0.0.1 ❶
OsmoSGSN(config-ns)# encapsulation udp local-port 23000 ❷
```

The example above configures a libosmogb based application to listen for incoming connections from PCUs on the specified address and port.

❶ Set the local side IP address for NS-over-UDP

❷ Set the local side UDP port number for NS-over-UDP. 23000 is the default

11.1.2 NS-over-FR-GRE configuration

The GPRS-NS protocol can alternatively be encapsulated over Frame Relay (FR). Traditionally this is communicated over SDH/PDH media, which we don't support. However, we can encapsulate the FR in GRE, and then that in IP.

The resulting NS-FR-GRE-IP stack can be converted by an off-the-shelf router with FR and IP support.

Example: GPRS NS-over-FR-GRE configuration

```
OsmoSGSN(config-ns)# encapsulation framerelay-gre enabled 1 ❶
OsmoSGSN(config-ns)# encapsulation framerelay-gre local-ip 127.0.0.1 ❷
```

❶ Enable FR-GRE encapsulation

❷ Set the local side IP address for NS-over-FR-GRE

11.1.3 NS Timer configuration

The NS protocol features a number of configurable timers.

Table 7: List of configurable NS timers

tns-block	(un)blocking timer timeout (secs)
tns-block-retries	(un)blocking timer; number of retries
tns-reset	reset timer timeout (secs)
tns-reset-retries	reset timer; number of retries
tns-test	test timer timeout (secs)
tns-alive	alive timer timeout(secs)
tns-alive-retries	alive timer; number of retries

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

11.2 Examining Gb interface status

There are several commands that can help to inspect and analyze the currently running system status with respect to the Gb interfaces.

Example: Inspecting NS state

```
OsmoSGSN> show ns
Encapsulation NS-UDP-IP     Local IP: 127.0.0.1, UDP Port: 23000
Encapsulation NS-FR-GRE-IP  Local IP: 0.0.0.0
```

Example: Inspecting NS statistics

```
OsmoSGSN> show ns stats
Encapsulation NS-UDP-IP     Local IP: 10.9.1.198, UDP Port: 23000
Encapsulation NS-FR-GRE-IP  Local IP: 0.0.0.0
NSEI  101, NS-VC 101, Remote: BSS, ALIVE UNBLOCKED, UDP 10.9.1.119:23000
 NSVC Peer Statistics:
  Packets at NS Level  ( In):     1024 (2/s 123/m 911/h 0/d)
  Packets at NS Level  (Out):     1034 (0/s 151/m 894/h 0/d)
  Bytes at NS Level    ( In):   296638 (1066/s 22222/m 274244/h 0/d)
  Bytes at NS Level    (Out):   139788 (0/s 48225/m 91710/h 0/d)
  NS-VC Block count          :        0 (0/s 0/m 0/h 0/d)
  NS-VC gone dead count      :        0 (0/s 0/m 0/h 0/d)
  NS-VC replaced other count:        0 (0/s 0/m 0/h 0/d)
  NS-VC changed NSEI count   :        0 (0/s 0/m 0/h 0/d)
  NS-VCI was invalid count   :        0 (0/s 0/m 0/h 0/d)
  NSEI was invalid count     :        0 (0/s 0/m 0/h 0/d)
  ALIVE ACK missing count    :        0 (0/s 0/m 0/h 0/d)
  RESET ACK missing count    :        0 (0/s 0/m 0/h 0/d)
 NSVC Peer Statistics:
  ALIVE response time        :        0 ms
```

Example: Inspecting BSSGP state

```
OsmoSGSN> show bssgp
NSEI   101, BVCI     2, RA-ID: 1-2-1-0, CID: 0, STATE: UNBLOCKED
NSEI   101, BVCI     0, RA-ID: 0-0-0-0, CID: 0, STATE: UNBLOCKED
```

FIXME: show nse

11.3 FIXME

11.3.1 Blocking / Unblocking / Resetting NS Virtual Connections

The user can manually perform operations on individual NSVCs:

- blocking a NSVC
- unblocking a NSVC
- resetting a NSVC

The VTY command used for this is the `nsvc (nsei|nsvci) <0-65535> (block|unblock|reset)` command available from the ENABLE node.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

11.4 Gb interface logging filters

There are some Gb-interface specific filters for the libosmocore logging subsystem, which can help to reduce the logged output to messages pertaining to a certain NS or BSSGP connection only.

Example: enabling a log filter for a given NSEI

```
OsmoSGSN> logging filter nsvc nsei 23
```

Example: enabling a log filter for a given NSVCI

```
OsmoSGSN> logging filter nsvc nsvci 23
```

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 8: Linux kernel default DSCP/TOS → priority mapping

TOS (binary)	DSCP (binary)	Priority (decimal)
xxx0000x	xxx000	0
xxx0001x	xxx000	0
xxx0010x	xxx001	0
xxx0011x	xxx001	0
xxx0100x	xxx010	2
xxx0101x	xxx010	2
xxx0110x	xxx011	2
xxx0111x	xxx011	2
xxx1000x	xxx100	6

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

Table 8: (continued)

TOS (binary)	DSCP (binary)	Priority (decimal)
xxx1001x	xxx100	6
xxx1010x	xxx101	6
xxx1011x	xxx101	6
xxx1100x	xxx110	4
xxx1101x	xxx110	4
xxx1110x	xxx111	4
xxx1111x	xxx111	4

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.q 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/eth0.9❶
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 ❷
 EGRESS priority mappings: ❸
```

❶ make sure to specify your specific VLAN interface name here instead of `eth0.9`

❷ ingress priority mappings (all PCP values mapped to priority 0)

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

❸ 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 eth0.9❶ type vlan egress-qos-map 1:1 2:2 3:3 ❷
$ sudo cat /proc/net/vlan/eth0.9 ❸
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 ❹
```

❶ make sure to specify your specific VLAN interface name here instead of `eth0.9`

❷ command to define three new egress QoS maps

❸ command to re-display the current status

❹ 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 egress 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 egress QoS map to map from priority to PCP

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

12.4.1 Full example of QoS for osmo-pcu uplink QoS

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

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

Table 9: DSCP and PCP assignments for osmo-bts uplink traffic in this example

Traffic	DSCP	PCP
Gb (NS)	10	1

1. configure the osmocom program to set the DSCP value
   - osmo-pcu.cfg: `dscp 10` in `udp bind` vty node
2. configure an egrees QoS map to map from priority to PCP

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

```
...
pcu
 gb ip-dscp 10
 gb socket-priority 1
 ...
```

Example Step 2: egress QoS map to map from DSCP values to priority values

```
$ sudo ip link set dev eth0.9❶ type vlan egress-qos-map 0:0 1:1 5:5 6:6 7:7 ❷
```

❶ make sure to specify your specific VLAN interface name here instead of `eth0.9`.

❷ create a egress QoS map that maps the priority value 1:1 to the PCP. We also include the mappings for 5, 6, and 7 from the osmo-bts example here (see [userman-osmobts]).

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.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

13.1 Scheduling Policy

The scheduler to use as well as some of its properties (such as realtime priority) can be configured at any time for the entire process. This sort of functionality is useful in order to increase priority for processes running time-constrained procedures, such as those acting on the Um interface, like *osmo-trx* or *osmo-bts*, where use of this feature is highly recommended.

Example: Set process to use RR scheduler

```
cpu-sched
 policy rr 1 (1)
```

(1) Configure process to use *SCHED_RR* policy with real time priority 1

Note
Running a process under real time priority requires CAP_SYS_NICE capabilities. In order to avoid running the process as *root* user, you can either modify the binary permissions with `setcap "CAP_SYS_NICE+ep" /path/to/binary`, or allow the user running the program to set rtprio by adding a file */etc/security/limits.d/youruser_allow-rtprio.conf* with something like `youruser - rtprio 99`. See `man 7 sched` for more information on the topic.

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 (1)
```

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

Or otherwise:

```
cpu-sched
 cpu-affinity all 0x01 (1)
```

(1) 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

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

```
OsmoTRX> show cpu-sched threads
Thread list for PID 338609:
 TID: 338609, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338610, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338611, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338629, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338630, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338631, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338634, NAME: 'UHDAsyncEvent', cpu-affinity: 0x3
 TID: 338635, NAME: 'TxLower', cpu-affinity: 0x3
 TID: 338636, NAME: 'RxLower', cpu-affinity: 0x3
 TID: 338637, NAME: 'RxUpper0', cpu-affinity: 0x3
 TID: 338638, NAME: 'TxUpper0', cpu-affinity: 0x3
 TID: 338639, NAME: 'RxUpper1', cpu-affinity: 0x3
 TID: 338640, NAME: 'TxUpper1', cpu-affinity: 0x3
```

At runtime, one can change the cpu-affinity mask for a given thread identifying it by either TID or name:

Example: Set CPU-affinity from VTY telnet interface

```
OsmoTRX> cpu-affinity TxLower 0x02 ❶
OsmoTRX> cpu-affinity TxLower 0x03 ❷
```

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

❷ Allow with TID *338636* (*RxLower*) only on CPU 0 and 1

Since thread names are set dynamically by the process during startup or at a later point after creating the thread itself, One may need to specify in the config file that the mask must be applied by the thread itself once being configured rather than trying to apply it immediately. To specify so, the *delay* keyword is using when configuring in the VTY. If the *delay* keyword is not used, the VTY will report and error and fail at startup when trying to apply a cpu-affinity mask for a yet-to-be-created thread.

Example: Set CPU-affinity from VTY config file

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

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

14 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])

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

A3/A8
Algorithm 3 and 8; Authentication and key generation algorithm in GSM and GPRS, typically COMP128v1/v2/v3 or MILENAGE are typically used

A5
Algorithm 5; Air-interface encryption of GSM; currently only A5/0 (no encryption), A5/1 and A5/3 are in use

Abis Interface
Interface between BTS and BSC, traditionally over E1 (*3GPP TS 48.058* [3gpp-ts-48-058] and *3GPP TS 52.021* [3gpp-ts-52-021])

ACC
Access Control Class; every BTS broadcasts a bit-mask of permitted ACC, and only subscribers with a SIM of matching ACC are permitted to use that BTS

AGCH
Access Grant Channel on Um interface; used to assign a dedicated channel in response to RACH request

AGPL
GNU Affero General Public License, a copyleft-style Free Software License

AQPSK
Adaptive QPSK, a modulation scheme used by VAMOS channels on Downlink

ARFCN
Absolute Radio Frequency Channel Number; specifies a tuple of uplink and downlink frequencies

AUC
Authentication Center; central database of authentication key material for each subscriber

BCCH
Broadcast Control Channel on Um interface; used to broadcast information about Cell and its neighbors

BCC
Base Station Color Code; short identifier of BTS, lower part of BSIC

BTS
Base Transceiver Station

BSC
Base Station Controller

BSIC
Base Station Identity Code; 16bit identifier of BTS within location area

BSSGP
Base Station Subsystem Gateway Protocol (*3GPP TS 48.018* [3gpp-ts-48-018])

BVCI
BSSGP Virtual Circuit Identifier

CBC
Cell Broadcast Centre; central entity of Cell Broadcast service

CBCH
Cell Broadcast Channel; used to transmit Cell Broadcast SMS (SMS-CB)

CBS
Cell Broadcast Service

CBSP
Cell Broadcast Service Protocol (*3GPP TS 48.049* [3gpp-ts-48-049])

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

CC
Call Control; Part of the GSM Layer 3 Protocol

CCCH
Common Control Channel on Um interface; consists of RACH (uplink), BCCH, PCH, AGCH (all downlink)

Cell
A cell in a cellular network, served by a BTS

CEPT
Conférence européenne des administrations des postes et des télécommunications; European Conference of Postal and Telecommunications Administrations.

CGI
Cell Global Identifier comprised of MCC, MNC, LAC and BSIC

CSFB
Circiut-Switched Fall Back; Mechanism for switching from LTE/EUTRAN to UTRAN/GERAN when circuit-switched services such as voice telephony are required.

dB
deci-Bel; relative logarithmic unit

dBm
deci-Bel (milliwatt); unit of measurement for signal strength of radio signals

DHCP
Dynamic Host Configuration Protocol (*IETF RFC 2131* [ietf-rfc2131])

downlink
Direction of messages / signals from the network core towards the mobile phone

DSCP
Differentiated Services Code Point (*IETF RFC 2474* [ietf-rfc2474])

DSP
Digital Signal Processor

dvnixload
Tool to program UBL and the Bootloader on a sysmoBTS

EDGE
Enhanced Data rates for GPRS Evolution; Higher-speed improvement of GPRS; introduces 8PSK

EGPRS
Enhanced GPRS; the part of EDGE relating to GPRS services

EIR
Equipment Identity Register; core network element that stores and manages IMEI numbers

ESME
External SMS Entity; an external application interfacing with a SMSC over SMPP

ETSI
European Telecommunications Standardization Institute

FPGA
Field Programmable Gate Array; programmable digital logic hardware

Gb
Interface between PCU and SGSN in GPRS/EDGE network; uses NS, BSSGP, LLC

GERAN
GPRS/EDGE Radio Access Network

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

GFDL
GNU Free Documentation License; a copyleft-style Documentation License

GGSN
GPRS Gateway Support Node; gateway between GPRS and external (IP) network

GMSK
Gaussian Minimum Shift Keying; modulation used for GSM and GPRS

GPL
GNU General Public License, a copyleft-style Free Software License

Gp
Gp interface between SGSN and GGSN; uses GTP protocol

GPRS
General Packet Radio Service; the packet switched 2G technology

GPS
Global Positioning System; provides a highly accurate clock reference besides the global position

GSM
Global System for Mobile Communications. ETSI/3GPP Standard of a 2G digital cellular network

GSMTAP
GSM tap; pseudo standard for encapsulating GSM protocol layers over UDP/IP for analysis

GSUP
Generic Subscriber Update Protocol. Osmocom-specific alternative to TCAP/MAP

GT
Global Title; an address in SCCP

GTP
GPRS Tunnel Protocol; used between SGSN and GGSN

HLR
Home Location Register; central subscriber database of a GSM network

HNB-GW
Home NodeB Gateway. Entity between femtocells (Home NodeB) and CN in 3G/UMTS.

HPLMN
Home PLMN; the network that has issued the subscriber SIM and has his record in HLR

IE
Information Element

IMEI
International Mobile Equipment Identity; unique 14-digit decimal number to globally identify a mobile device, optionally with a 15th checksum digit

IMEISV
IMEI software version; unique 14-digit decimal number to globally identify a mobile device (same as IMEI) plus two software version digits (total digits: 16)

IMSI
International Mobile Subscriber Identity; 15-digit unique identifier for the subscriber/SIM; starts with MCC/MNC of issuing operator

IP
Internet Protocol (*IETF RFC 791* [ietf-rfc791])

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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])

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

syslog
System logging service of UNIX-like operating systems

System Information
A set of downlink messages on the BCCH and SACCH of the Um interface describing properties of the cell and network

TCH
Traffic Channel; used for circuit-switched user traffic (mostly voice) in GSM

TCP
Transmission Control Protocol; (*IETF RFC 793* [ietf-rfc793])

TFTP
Trivial File Transfer Protocol; (*IETF RFC 1350* [ietf-rfc1350])

TOS
Type Of Service; bit-field in IPv4 header, now re-used as DSCP (*IETF RFC 791* [ietf-rfc791])

TRX
Transceiver; element of a BTS serving a single carrier

TS
Technical Specification

u-Boot
Boot loader used in various embedded systems

UBI
An MTD wear leveling system to deal with NAND flash in Linux

UBL
Initial bootloader loaded by the TI Davinci SoC

UDP
User Datagram Protocol (*IETF RFC 768* [ietf-rfc768])

UICC
Universal Integrated Chip Card; A smart card according to *ETSI TR 102 216* [etsi-tr102216]

Um interface
U mobile; Radio interface between MS and BTS

uplink
Direction of messages: Signals from the mobile phone towards the network

USIM
Universal Subscriber Identity Module; application running on a UICC to provide subscriber identity for UMTS and GSM networks

USSD
Unstructured Supplementary Service Data; textual dialog between subscriber and core network, e.g. **100 → Your extension is 1234*

VAMOS
Voice services over Adaptive Multi-user channels on One Slot; an optional extension for GSM specified in Release 9 of 3GPP GERAN specifications (*3GPP TS 48.018* [3gpp-ts-48-018]) allowing two independent UEs to transmit and receive simultaneously on traffic channels

VCTCXO
Voltage Controlled, Temperature Compensated Crystal Oscillator; a precision oscillator, superior to a classic crystal oscillator, but inferior to an OCXO

VLAN
Virtual LAN in the context of Ethernet (*IEEE 802.1Q* [ieee-802.1q])

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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 10: TCP/UDP port numbers

L4 Protocol	Port Number	Purpose	Software
UDP	1984	Osmux	osmo-mgw, osmo-bts
UDP	2427	MGCP GW	osmo-bsc_mgcp, osmo-mgw
TCP	2775	SMPP (SMS interface for external programs)	osmo-nitb
TCP	3002	A-bis/IP OML	osmo-bts, osmo-bsc, osmo-nitb
TCP	3003	A-bis/IP RSL	osmo-bts, osmo-bsc, osmo-nitb
TCP	4227	telnet (VTY)	osmo-pcap-client
TCP	4228	telnet (VTY)	osmo-pcap-server
TCP	4236	Control Interface	osmo-trx
TCP	4237	telnet (VTY)	osmo-trx
TCP	4238	Control Interface	osmo-bts
TCP	4239	telnet (VTY)	osmo-stp
TCP	4240	telnet (VTY)	osmo-pcu
TCP	4241	telnet (VTY)	osmo-bts
TCP	4242	telnet (VTY)	osmo-nitb, osmo-bsc, cellmgr-ng
TCP	4243	telnet (VTY)	osmo-bsc_mgcp, osmo-mgw
TCP	4244	telnet (VTY)	osmo-bsc_nat
TCP	4245	telnet (VTY)	osmo-sgsn
TCP	4246	telnet (VTY)	osmo-gbproxy
TCP	4247	telnet (VTY)	OsmocomBB
TCP	4249	Control Interface	osmo-nitb, osmo-bsc
TCP	4250	Control Interface	osmo-bsc_nat
TCP	4251	Control Interface	osmo-sgsn
TCP	4252	telnet (VTY)	sysmobts-mgr
TCP	4253	telnet (VTY)	osmo-gtphub
TCP	4254	telnet (VTY)	osmo-msc
TCP	4255	Control Interface	osmo-msc
TCP	4256	telnet (VTY)	osmo-sip-connector
TCP	4257	Control Interface	osmo-ggsn, ggsn (OpenGGSN)
TCP	4258	telnet (VTY)	osmo-hlr
TCP	4259	Control Interface	osmo-hlr
TCP	4260	telnet (VTY)	osmo-ggsn
TCP	4261	telnet (VTY)	osmo-hnbgw
TCP	4262	Control Interface	osmo-hnbgw
TCP	4263	Control Interface	osmo-gbproxy
TCP	4264	telnet (VTY)	osmo-cbc
TCP	4265	Control Interface	osmo-cbc

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

Table 10: (continued)

L4 Protocol	Port Number	Purpose	Software
TCP	4266	D-GSM MS Lookup: mDNS serve	osmo-hlr
TCP	4267	Control Interface	osmo-mgw
TCP	4268	telnet (VTY)	osmo-uecups
SCTP	4268	UECUPS	osmo-uecups
TCP	4269	telnet (VTY)	osmo-e1d
TCP	4270	telnet (VTY)	osmo-isdntap
TCP	4271	telnet (VTY)	osmo-smlc
TCP	4272	Control Interface	osmo-smlc
TCP	4273	telnet (VTY)	osmo-hnodeb
TCP	4274	Control Interface	osmo-hnodeb
TCP	4275	telnet (VTY)	osmo-upf
TCP	4276	Control Interface	osmo-upf
TCP	4277	telnet (VTY)	osmo-pfcp-tool
TCP	4278	Control Interface	osmo-pfcp-tool
UDP	4729	GSMTAP	Almost every osmocom project
TCP	5000	A/IP	osmo-bsc, osmo-bsc_nat
UDP	23000	GPRS-NS over IP default port	osmo-pcu, osmo-sgsn, osmo-gbproxy
TCP	48049	BSC-CBC (CBSP) default port	osmo-bsc, osmo-cbc

B Bibliography / References

B.0.0.0.1 References

[1] [userman-ice1usb] Osmocom Project: icE1usb User Manual.

[2] [userman-ogt] Pau Espin: osmo-gsm-tester User Manual.

[3] [userman-remsim] Harald Welte: osmo-remsim User Manual.

[4] [osmobts-abis-spec] Neels Hofmeyr & Harald Welte. OsmoBTS Abis Protocol Specification. https://ftp.osmocom.org/docs/latest/osmobts-abis.pdf

[5] [userman-osmobsc] Osmocom Project: OsmoBSC User Manual. https://ftp.osmocom.org/docs/latest/osmobsc-usermanual.pdf

[6] [vty-ref-osmobsc] Osmocom Project: OsmoBSC VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmobsc-vty-reference.pdf

[7] [userman-osmobts] Osmocom Project: OsmoBTS User Manual. https://ftp.osmocom.org/docs/latest/osmobts-usermanual.pdf

[8] [vty-ref-osmobts] Osmocom Project: OsmoBTS VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmobts-trx-vty-reference.pdf https://ftp.osmocom.org/docs/latest/osmobts-sysmo-vty-reference.pdf https://ftp.osmocom.org/docs/latest/osmobts-lc15-vty-reference.pdf https://ftp.osmocom.org/docs/latest/-osmobts-oc2g-vty-reference.pdf https://ftp.osmocom.org/docs/latest/osmobts-octphy-vty-reference.pdf https://ftp.osmocom.org/docs/latest/osmobts-virtual-vty-reference.pdf

[9] [userman-osmocbc] Osmocom Project: OsmoCBC User Manual. https://ftp.osmocom.org/docs/latest/-osmocbc-usermanual.pdf

[10] [vty-ref-osmocbc] Osmocom Project: OsmoCBC VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmocbc-vty-reference.pdf

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

[11] [userman-osmogbproxy] Osmocom Project: OsmoGBProxy User Manual. https://ftp.osmocom.org/docs/-latest/osmogbproxy-usermanual.pdf

[12] [vty-ref-osmogbproxy] Osmocom Project: OsmoGBPRoxy VTY Reference Manual. https://ftp.osmocom.org/-docs/latest/osmogbproxy-vty-reference.pdf

[13] [userman-osmoggsn] Osmocom Project: OpenGGSN User Manual. https://ftp.osmocom.org/docs/latest/-osmoggsn-usermanual.pdf

[14] [vty-ref-osmoggsn] Osmocom Project: OsmoGGSN VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmoggsn-vty-reference.pdf

[15] [userman-osmohlr] Osmocom Project: OsmoHLR User Manual. https://ftp.osmocom.org/docs/latest/osmohlr-usermanual.pdf

[16] [vty-ref-osmohlr] Osmocom Project: OsmoHLR VTY Reference Manual. https://ftp.osmocom.org/docs/latest/-osmohlr-vty-reference.pdf

[17] [userman-osmohnbgw] Osmocom Project: OsmoHNBGW User Manual. https://ftp.osmocom.org/docs/latest/-osmohnbgw-usermanual.pdf

[18] [vty-ref-osmohnbgw] Osmocom Project: OsmoHNBGW VTY Reference Manual. https://ftp.osmocom.org/-docs/latest/osmohnbgw-vty-reference.pdf

[19] [userman-osmomgw] Osmocom Project: OsmoMGW User Manual. https://ftp.osmocom.org/docs/latest/-osmomgw-usermanual.pdf

[20] [vty-ref-osmomgw] Osmocom Project: OsmoMGW VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmomgw-vty-reference.pdf

[21] [userman-osmomsc] Osmocom Project: OsmoMSC User Manual. https://ftp.osmocom.org/docs/latest/-osmomsc-usermanual.pdf

[22] [vty-ref-osmomsc] Osmocom Project: OsmoMSC VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmomsc-vty-reference.pdf

[23] [userman-osmonitb] Osmocom Project: OsmoNITB User Manual. https://ftp.osmocom.org/docs/latest/-osmonitb-usermanual.pdf

[24] [vty-ref-osmonitb] Osmocom Project: OsmoNITB VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmonitb-vty-reference.pdf

[25] [userman-osmopcu] Osmocom Project: OsmoPCU User Manual. https://ftp.osmocom.org/docs/latest/-osmopcu-usermanual.pdf

[26] [vty-ref-osmopcu] Osmocom Project: OsmoPCU VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmopcu-vty-reference.pdf

[27] [userman-osmosgsn] Osmocom Project: OsmoSGSN User Manual. https://ftp.osmocom.org/docs/latest/-osmosgsn-usermanual.pdf

[28] [vty-ref-osmosgsn] Osmocom Project: OsmoSGSN VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmosgsn-vty-reference.pdf

[29] [userman-osmosipconnector] Osmocom Project: OsmoSIPconnector User Manual. https://ftp.osmocom.org/-docs/latest/osmosipconnector-usermanual.pdf

[30] [vty-ref-osmosipconnector] Osmocom Project: OsmoSIPconnector VTY Reference Manual. https://ftp.osmocom.org/docs/latest/osmosipconnector-vty-reference.pdf

[31] [userman-osmosmlc] Osmocom Project: OsmoSMLC User Manual. https://ftp.osmocom.org/docs/latest/-osmosmlc-usermanual.pdf

[32] [vty-ref-osmosmlc] Osmocom Project: OsmoSMLC VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmosmlc-vty-reference.pdf

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

[33] [userman-osmostp] Osmocom Project: OsmoSTP User Manual. https://ftp.osmocom.org/docs/latest/osmostp-usermanual.pdf

[34] [vty-ref-osmostp] Osmocom Project: OsmoSTP VTY Reference Manual. https://ftp.osmocom.org/docs/latest/-osmostp-vty-reference.pdf

[35] [userman-osmotrx] Osmocom Project: OsmoTRX User Manual. https://ftp.osmocom.org/docs/latest/osmotrx-usermanual.pdf

[36] [vty-ref-osmotrx] Osmocom Project: OsmoTRX VTY Reference Manual. https://ftp.osmocom.org/docs/-latest/osmotrx-uhd-vty-reference.pdf https://ftp.osmocom.org/docs/latest/osmotrx-lms-vty-reference.pdf https://ftp.osmocom.org/docs/latest/osmotrx-ipc-vty-reference.pdf https://ftp.osmocom.org/docs/latest/-osmotrx-usrp1-vty-reference.pdf

[37] [3gpp-ts-23-041] 3GPP TS 23.041: Technical realization of Cell Broadcast Service (CBS)

[38] [3gpp-ts-23-048] 3GPP TS 23.048: Security mechanisms for the (U)SIM application toolkit; Stage 2 https://www.3gpp.org/DynaReport/23048.htm

[39] [3gpp-ts-23-236] 3GPP TS 23.236: Intra-domain connection of Radio Access Network (RAN) nodes to multiple Core Network (CN) nodes https://www.3gpp.org/DynaReport/23236.htm

[40] [3gpp-ts-24-007] 3GPP TS 24.007: Mobile radio interface signalling layer 3; General Aspects https://www.3gpp.org/DynaReport/24007.htm

[41] [3gpp-ts-24-008] 3GPP TS 24.008: Mobile radio interface Layer 3 specification; Core network protocols; Stage 3. https://www.3gpp.org/dynareport/24008.htm

[42] [3gpp-ts-31-101] 3GPP TS 31.101: UICC-terminal interface; Physical and logical characteristics https://www.3gpp.org/DynaReport/31101.htm

[43] [3gpp-ts-31-102] 3GPP TS 31.102: Characteristics of the Universal Subscriber Identity Module (USIM) application https://www.3gpp.org/DynaReport/31102.htm

[44] [3gpp-ts-31-103] 3GPP TS 31.103: Characteristics of the IMS Subscriber Identity Module (ISIM) application https://www.3gpp.org/DynaReport/31103.htm

[45] [3gpp-ts-31-111] 3GPP TS 31.111: Universal Subscriber Identity Module (USIM) Application Toolkit (USAT) https://www.3gpp.org/DynaReport/31111.htm

[46] [3gpp-ts-31-115] 3GPP TS 31.115: Secured packet structure for (Universal) Subscriber Identity Module (U)SIM Toolkit applications https://www.3gpp.org/DynaReport/31115.htm

[47] [3gpp-ts-31-116] 3GPP TS 31.116: Remote APDU Structure for (U)SIM Toolkit applications https://www.3gpp.org/DynaReport/31116.htm

[48] [3gpp-ts-35-205] 3GPP TS 35.205: 3G Security; Specification of the MILENAGE algorithm set: General

[49] [3gpp-ts-35-206] 3GPP TS 35.206: 3G Security; Specification of the MILENAGE algorithm set: Algorithm specification https://www.3gpp.org/DynaReport/35206.htm

[50] [3gpp-ts-44-006] 3GPP TS 44.006: Mobile Station - Base Station System (MS - BSS) interface; Data Link (DL) layer specification https://www.3gpp.org/DynaReport/44006.htm

[51] [3gpp-ts-44-018] 3GPP TS 44.018: Mobile radio interface layer 3 specification; Radio Resource Control (RRC) protocol https://www.3gpp.org/DynaReport/44018.htm

[52] [3gpp-ts-44-064] 3GPP TS 44.064: Mobile Station - Serving GPRS Support Node (MS-SGSN); Logical Link Control (LLC) Layer Specification https://www.3gpp.org/DynaReport/44064.htm

[53] [3gpp-ts-45-002] 3GPP TS 45.002: Digital cellular telecommunications system (Phase 2+) (GSM); GSM/EDGE Multiplexing and multiple access on the radio path https://www.3gpp.org/DynaReport/45002.htm

[54] [3gpp-ts-48-008] 3GPP TS 48.008: Mobile Switching Centre - Base Station system (MSC-BSS) interface; Layer 3 specification https://www.3gpp.org/DynaReport/48008.htm

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

[55] [3gpp-ts-48-016] 3GPP TS 48.016: General Packet Radio Service (GPRS); Base Station System (BSS) - Serving GPRS Support Node (SGSN) interface; Network service https://www.3gpp.org/DynaReport/48016.htm

[56] [3gpp-ts-48-018] 3GPP TS 48.018: General Packet Radio Service (GPRS); Base Station System (BSS) - Serving GPRS Support Node (SGSN); BSS GPRS protocol (BSSGP) https://www.3gpp.org/DynaReport/48018.htm

[57] [3gpp-ts-48-049] 3GPP TS 48.049: Digital cellular communications system; Base Station Controller - Cell Broadcast Centre (BSC-CBC) interface specification; Cell Broadcast Service Protocol (CBSP) https://www.3gpp.org/DynaReport/48049.htm

[58] [3gpp-ts-48-056] 3GPP TS 48.056: Base Station Controller - Base Transceiver Station (BSC - BTS) interface; Layer 2 specification https://www.3gpp.org/DynaReport/48056.htm

[59] [3gpp-ts-48-058] 3GPP TS 48.058: Base Station Controller - Base Transceiver Station (BSC - BTS) Interface; Layer 3 specification https://www.3gpp.org/DynaReport/48058.htm

[60] [3gpp-ts-51-011] 3GPP TS 51.011: Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface

[61] [3gpp-ts-51-014] 3GPP TS 51.014: Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface https://www.3gpp.org/DynaReport/51014.htm

[62] [3gpp-ts-52-021] 3GPP TS 52.021: Network Management (NM) procedures and messages on the A-bis interface https://www.3gpp.org/DynaReport/52021.htm

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

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

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

[66] [etsi-ts102671] ETSI TS 102 671: Smart Cards; Machine to Machine UICC; Physical and logical characteristics https://www.etsi.org/deliver/etsi_ts/102600_102699/102671/18.01.00_60/ts_102671v180100p.pdf

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

[68] [ietf-rfc768] IETF RFC 768: User Datagram Protocol https://tools.ietf.org/html/rfc768

[69] [ietf-rfc791] IETF RFC 791: Internet Protocol https://tools.ietf.org/html/rfc791

[70] [ietf-rfc793] IETF RFC 793: Transmission Control Protocol https://tools.ietf.org/html/rfc793

[71] [ietf-rfc1035] IETF RFC 1035: Domain Names - Implementation and Specification https://tools.ietf.org/html/rfc1035

[72] [ietf-rfc1350] IETF RFC 1350: Trivial File Transfer Protocol https://tools.ietf.org/html/rfc1350

[73] [ietf-rfc2131] IETF RFC 2131: Dynamic Host Configuration Protocol https://tools.ietf.org/html/rfc2131

[74] [ietf-rfc2474] IETF RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv44 and IPv6 Headers https://tools.ietf.org/html/rfc2474

[75] [ietf-rfc2719] IETF RFC 2719: Signal Transport over IP https://tools.ietf.org/html/rfc2719

[76] [ietf-rfc3331] IETF RFC 3331: Message Transfer Part 2 User Adaptation Layer https://tools.ietf.org/html/rfc3331

[77] [ietf-rfc3550] IETF RFC 3550: RTP: A Transport protocol for Real-Time Applications https://tools.ietf.org/html/rfc3550

[78] [ietf-rfc3596] IETF RFC 3596: DNS Extensions to Support IP Version 6 https://tools.ietf.org/html/rfc3596

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

[79] [ietf-rfc3868] IETF RFC 3868: SCCP User Adaptation Layer https://tools.ietf.org/html/rfc3868

[80] [ietf-rfc4165] IETF RFC 4165: Message Transfer Part 2 Peer-to-Peer Adaptation Layer https://tools.ietf.org/-html/rfc4165

[81] [ietf-rfc4251] IETF RFC 4251: The Secure Shell (SSH) Protocol Architecture https://tools.ietf.org/html/-rfc4251

[82] [ietf-rfc4666] IETF RFC 4666: Message Transfer Part 3 User Adaptation Layer https://tools.ietf.org/html/-rfc4666

[83] [ietf-rfc5771] IETF RFC 5771: IANA Guidelines for IPv4 Multicast Address Assignments https://tools.ietf.org/html/rfc5771

[84] [itu-t-q701] ITU-T Q.701: Functional Description of the Message Transfer Part (MTP) https://www.itu.int/rec/-T-REC-Q.701/en/

[85] [itu-t-q711] ITU-T Q.711: Functional Description of the Signalling Connection Control Part https://www.itu.int/rec/T-REC-Q.711/en/

[86] [itu-t-q713] ITU-T Q.713: Signalling connection control part formats and codes https://www.itu.int/rec/T-REC-Q.713/en/

[87] [itu-t-q714] ITU-T Q.714: Signalling connection control part procedures https://www.itu.int/rec/T-REC-Q.714/en/

[88] [itu-t-q921] ITU-T Q.921: ISDN user-network interface - Data link layer specification https://www.itu.int/rec/-T-REC-Q.921/en

[89] [smpp-34] SMPP Develoepers Forum. Short Message Peer-to-Peer Protocol Specification v3.4 https://docs.nimta.com/SMPP_v3_4_Issue1_2.pdf

[90] [gnu-agplv3] Free Software Foundation. GNU Affero General Public License. https://www.gnu.org/licenses/-agpl-3.0.en.html

[91] [freeswitch_pbx] FreeSWITCH SIP PBX https://freeswitch.org

[92] [tw-ts-001] TW-TS-001: Enhanced RTP transport of FR and EFR codec frames in an IP-based GSM RAN https://www.freecalypso.org/specs/tw-ts-001-v010100.txt

[93] [tw-ts-002] TW-TS-002: Enhanced RTP transport of HRv1 codec frames in an IP-based GSM RAN https://www.freecalypso.org/specs/tw-ts-002-v010100.txt

[94] [tw-ts-003] TW-TS-003: BSSMAP extension for selection of enhanced RTP transport formats https://www.freecalypso.org/specs/tw-ts-003-v010002.txt

C GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. http://fsf.org/

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

C.1 PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

C.2 APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Section whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not Transparent is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, Title Page means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

c. State on the Title Page the name of the publisher of the Modified Version, as the publisher.

d. Preserve all the copyright notices of the Document.

e. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

f. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

g. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

h. Include an unaltered copy of this License.

i. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

j. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

k. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

l. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

m. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

n. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Sections.

o. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Section and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

C.6 COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH

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.

Copyright © 2013-2021 sysmocom - s.f.m.c. GmbH