sysmocom

sysmocom - s.f.m.c. GmbH

OsmoUPF User Manual

by Neels Janosch Hofmeyr

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

HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
1	December 2021	Initial OsmoUPF manual	NJH

Contents

1 Foreword

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

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

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

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

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

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

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

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

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

1.1 Acknowledgements

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

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

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

May the source be with you!

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

1.2 Endorsements

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

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

2 Preface

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

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

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

2.1 FOSS lives by contribution!

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

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

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

We’re looking forward to receiving your contributions.

2.2 Osmocom and sysmocom

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

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

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

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

2.3 Corrections

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

2.4 Legal disclaimers

2.4.1 Spectrum License

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

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

2.4.2 Software License

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

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

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

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

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

2.4.3 Trademarks

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

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

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.

OsmoUPF

4 Overview

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

4.1 About OsmoUPF

OsmoUPF is the Osmocom implementation of a User Plane Function for 2G, 3G, 4G and 5G mobile network packet switched user data management. It implements:

- A Packet Forwarding Control Protocol (PFCP) entity to manage the GTP user plane of mobile subscribers.
- GTP forwarding as well as encapsulation/decapsulation of user traffic, using the Linux mainline kernel GTP module.

The aim is to provide:

- 1000 modifications of tunnel state per second (add/remove/modify),
- 4-8 Gbps throughput,
- 100-125k concurrent GTP tunnels.

A typical network scenario using OsmoUPF is illustrated in the following diagram:

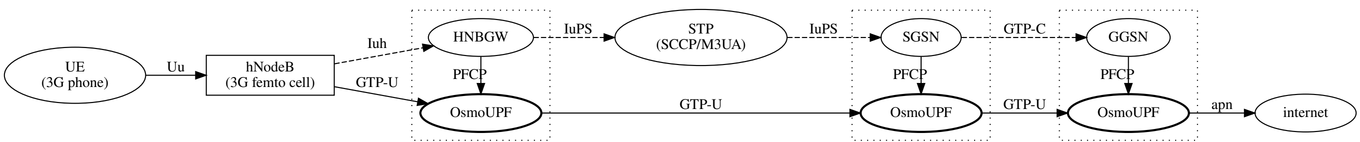

Figure 1: Typical network architecture used with OsmoUPF

Note
at the time of writing this section, the only Osmocom component providing a PFCP CPF interface is OsmoHNBGW. PFCP support has not yet made its way into OsmoSGSN nor OsmoGGSN.

4.2 the PFCP interface

PFCP is specified by 3GPP TS 29.244.

OsmoUPF implements a PFCP User Plane Function interface, listening for PFCP requests from PFCP Control Plane Function clients, to carry out proxy-relaying and encapsulation/decapsulation of GTP tunnels.

OsmoUPF does not support the complete PFCP feature set. It detects exactly two use cases that will provide service of actual GTP tunnels:

tunend use case

```
Access                  osmo-upf              Core
 PGW                        |              PDN/internet
  |                PDR1:  > FAR1:               |
  |                IP/GTP | IP                  |
  |        ------> F-TEID |            -----> |
  |                       |                     |
  |                FAR2:  < PDR2:               |
  |                IP/GTP | IP                  |
  | F-TEID <------        | UE IP addr <----- |
```

- `tunend`: GTP tunnel encapsulation/decapsulation:
  - One Packet Detection Rule (PDR) accepts a GTP tunnel from the Access side with an Outer Header Removal.
  - This PDR uses a Forwarding Action Rule (FAR) for plain IP towards Core.
  - Another PDR accepts plain IP on a specific IP address from Core.
  - The second PDR uses a FAR towards Access with Outer Header Creation for GTP.

tunmap use case

```
Access                  osmo-upf                 Core
 PGW                        |                    PGW
  |                PDR1:  > FAR1:                  |
  |                IP/GTP | IP/GTP                 |
  |        ------> F-TEID |        -----> F-TEID |
  |                       |                        |
  |                FAR2:  < PDR2:                  |
  |                IP/GTP | IP/GTP                 |
  | F-TEID <------        | F-TEID <-----          |
```

- `tunmap`: GTP tunnel forwarding:
  - One Packet Detection Rule (PDR) accepts a GTP tunnel from the Access side with an Outer Header Removal.
  - This PDR uses a Forwarding Action Rule (FAR) towards Core with an Outer Header Creation for GTP.
  - A second PDR+FAR pair like above, with Access and Core swapped.

Access and Core must be indicated by the Source Interface IE (PDR) and Destination Interface IE (FAR) in PFCP.

Any set of rules only partially or not at all matching the above PDR and FAR rules will not result in any actions on the GTP user plane, but will still return a successful outcome in the PFCP messages.

For example, a rule set using a Source Interface other than "Access" or "Core" results in a PFCP no-op, returning PFCP responses with successful outcome, but not providing any GTP-U service.

This is a direct result of:

- allowing PFCP rule sets to be setup incrementally by several subsequent PFCP messages, and of

- OsmoUPF using Linux kernel features for the GTP user plane, where there is either a full bidirectional GTP tunnel in place or none at all.

For example, for `tunmap`, a typical CPF will establish a PFCP session in two steps: first request a local F-TEID from the UPF before passing on a data service request from Access to Core. When the Core side has responded with its GTP details, the PFCP session at the UPF is updated (Session Modifification), to form a complete PFCP rule set.

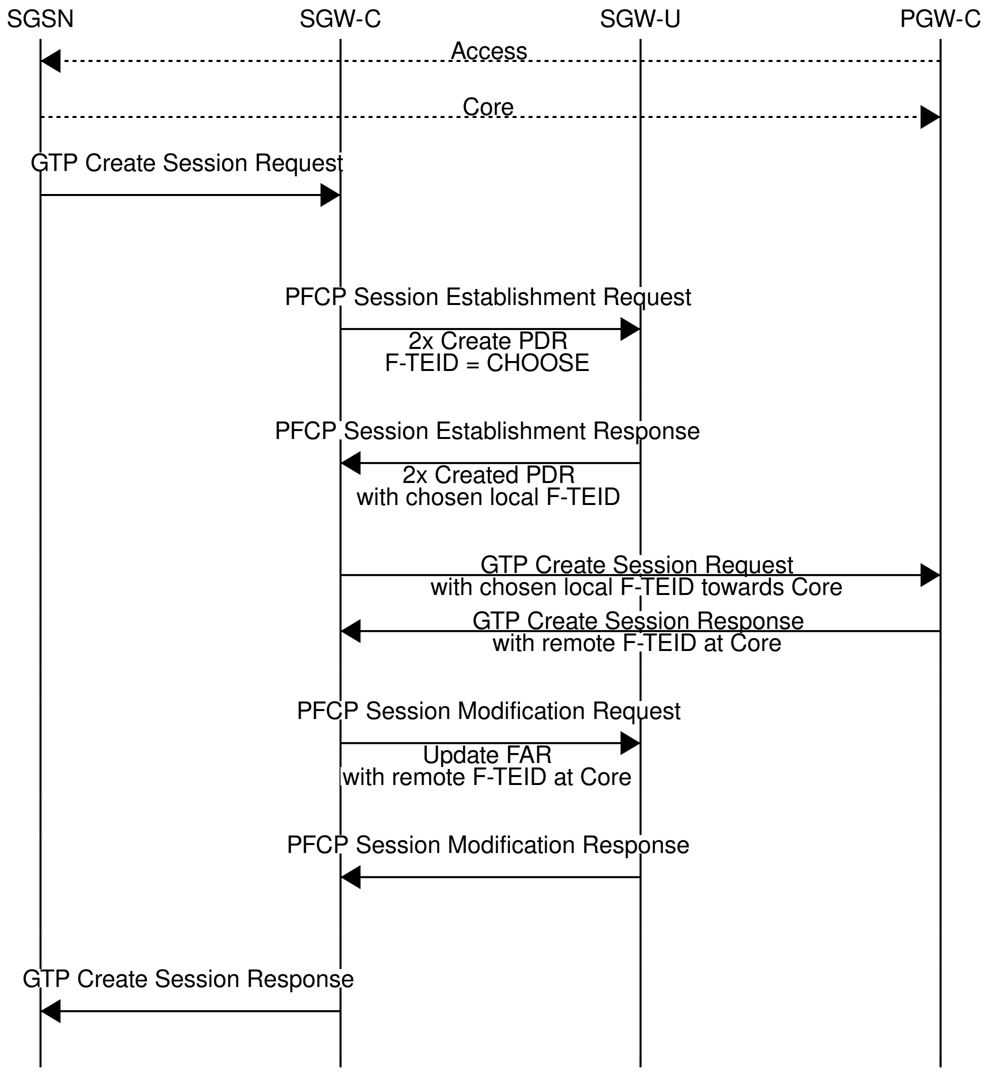

Figure 2: Typical sequence of establishing a GTP-U tunnel relay

The OsmoUPF logging as well as the VTY interface yield information on whether a ruleset results in an actual bidirectional GTP tunnel being set up.

4.3 the GTP interface

OsmoUPF requires the following Linux kernel features to provide the GTP user plane functionality:

- the Linux kernel GTP module for encapsulation/decapsulation between GTP and plain IP.
- the Linux netfilter nftables feature for relaying GTP, i.e. forwarding between two GTP tunnels.

Tunnel relaying with netfilter requires at least Linux kernel 5.17.

To be able to interact with these Linux kernel features, the osmo-upf binary needs cap_net_admin privileges, as in:

```
sudo setcap cap_net_admin+pe /usr/bin/osmo-upf
```

Without above Linux kernel features, or when no cap_net_admin is available, OsmoUPF is only useful for testing PFCP clients: the GTP features may be run in mockup mode, so that OsmoUPF serves as a "dry run" PFCP server.

5 Running OsmoUPF

The OsmoUPF executable (`osmo-upf`) offers the following command-line arguments:

5.1 SYNOPSIS

osmo-upf [-h|-V] [-D] [-c *CONFIGFILE*]

5.2 OPTIONS

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

-V, --version
: Print the compile-time version number of the OsmoHNBGW 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-upf.cfg` in the current working directory.

5.3 Multiple instances

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

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

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

The PFCP port is specified to be fixed as port 8805. Hence, each osmo-upf process needs to run on a distinct local interface:

```
pfcp
  local-addr 10.9.0.2
```

For GTP encapsulation/decapsulation and GTP tunnel relaying, osmo-upf depends on the IP addresses configured at the Linux kernel GTP module, and the IP addresses negotiated within PFCP by the control plane function.

If multiple `osmo-upf` processes are running on the same Linux kernel, each `osmo-upf` needs to be configured with a distinct netfilter table name, so that naming of individual tunnel rulesets does not collide:

```
tunmap
 table-name osmo-upf-2
```

5.4 Configure PFCP Server

The following example configures OsmoUPF to listen for PFCP association requests from Control Plane Function entities on local interface 10.9.8.7, port 8805:

```
pfcp
 local-addr 10.9.8.7
```

3GPP TS 29.244 4.2.2 specifies that PFCP Request messages shall be sent to UDP port 8805, i.e. the PFCP port is fixed as 8805 and currently not configurable in osmo-upf.

Setting a *local-addr* is required: the PFCP protocol features a Node ID, which uniquely identifies PFCP peers across different interfaces. According to the PFCP specification, the Node ID can be a fully-qualified domain name (FQDN) or an IP address. Currently, osmo-upf has no support for using an FQDN as Node ID, and so far uses the *local-addr* as local Node ID — hence the *local-addr* must not be "0.0.0.0", which is an unfortunate consequence. This is likely to improve in the future, see https://osmocom.org/issues/5682 .

5.5 Linux Kernel Features

OsmoUPF uses two distinct Linux kernel features:

- The GTP module is used for `tunend`: GTP encapsulation/decapsulation from/to "the internet".
- The netfilter framework and nftables are used for `tunmap`: GTP tunnel proxying, also known as tunnel forwarding or tunnel mapping.

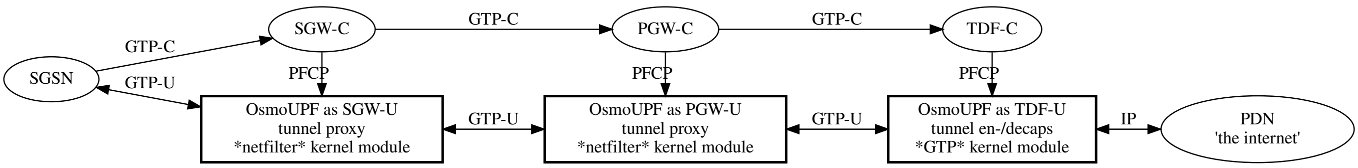

Figure 3: Linux kernel feature usage

GTP kernel module configuration in the `tunend` section can be omitted for sites that serve only as GTP forwarding proxy, without encapsulation/decapsulation of GTP payloads — except to provide GTP Echo service, see Section 5.7.1.

Netfilter configuration in the `tunmap` section can be omitted for sites only serving as GTP tunnel endpoint.

5.6 Configure Linux Kernel GTP Module for `tunend`

The Linux kernel GTP module is used for the `tunend` use case, i.e. GTP encapsulation/decapsulation from/to "the internet".

To use the GTP kernel module, OsmoUPF requires a GTP device, which is a dedicated network device provided by the Linux kernel, serving as GTP tunnel endpoint. It is typically named like "apn0".

`osmo-upf` can either create a GTP device on startup, or use a pre-existing GTP device. To en/decapsulate GTP, the APN device needs to be assigned an IP address range that matches the UE IP addresses that are configured in GTP-C / PFCP.

The following configuration placed in `osmo-upf.cfg` creates a GTP device called `apn23` on startup of osmo-upf, which is destroyed on program exit. It listens for GTP on local IP address `1.2.3.4`:

```
tunend
 dev create apn23 1.2.3.4
```

TODO
`osmo-upf` is not yet able to configure this network device's IP address range, MTU etc.

The following configuration placed in `osmo-upf.cfg` uses a pre-existing device called `apn42`:

```
tunend
 dev use apn42 2.3.4.5
```

GTP kernel devices can be managed manually using the `gtp-link` program available from the *libgtpnl* project:

```
# gtp-link add apn42
(keep this process running)
# ip addr add dev apn42 192.168.42.1/24

$ osmo-upf -c osmo-upf.cfg
```

It is possible to configure multiple GTP devices in `osmo-upf.cfg`. Depending on the Network Instance name, osmo-upf creates tunnel endpoints on the GTP device with a matching IP address:

- The Network Instance IE in the PDR on the Access side determines the local IP address to use, see Section 6.
- This local IP address in turn determines the GTP device to use.

It is possible for a GTP device to listen on ANY — just omit the IP address in the `dev` config. In this case, all Network Instance names will be served by this GTP device. When using ANY, there should be exactly one GTP dev configured.

5.7 Configure Linux netfilter for `tunmap`

The Linux kernel netfilter module is used for GTP tunnel proxying, also known as tunnel forwarding or tunnel mapping.

When using the netfilter module, you may set up `osmo-upf.cfg` for: - GTP Echo (required) - nft table name (optional)

5.7.1 GTP Echo

You need to ensure that OsmoUPF responds to GTP Echo requests. - A GTP device configured for `tunend` implicitly includes a GTP Echo service. - For `tunmap`, no GTP Echo mechanism is implemented.

So, when your use case is `tunmap` only, you should still add a GTP device as for `tunend`, only to provide the GTP Echo service.

Here are some options to do so:

If you have no GTP devices configured in `osmo-upf.cfg` yet, you can add a single GTP device without a specific IP address, in order to respond to GTP-U Echo requests on all interfaces to anyone that is asking:

```
tunend
 dev create gtp-echo
```

Note that `gtp-echo` is just an arbitrary GTP device name, choose any string that makes a valid network device name and is still available, as in the `dev` argument in the `ip addr show dev` command on Linux.

This will bind osmo-upf on 0.0.0.0:2152 to respond to GTP Echo requests.

If you would like to limit GTP Echo responses to specific network interfaces, you need to add a separate GTP device per local IP address:

```
tunend
 dev create gtp-echo1 192.168.0.23
 dev create gtp-echo2 10.9.8.17
```

This will bind osmo-upf only on 192.168.0.23:2152 and 10.9.8.17:2152 to respond to GTP Echo requests.

For creating and manipulating a GTP device in more versatile ways, see Section 5.6.

5.7.2 nft Table Name

For `tunmap`, `osmo-upf` creates a new nft table, under which it submits rule sets for GTP tunnel proxying. This table name defaults to `osmo-upf`. A custom table name can be configured in `osmo-upf.cfg` like this:

```
tunmap
 table-name my-table-name
```

When running more than one osmo-upf process on a system, pick distinct table names to avoid name collisions in the nftables rulesets.

5.8 IP Forwarding

In order to allow forwarding GTP payloads, the Linux operating system must be configured to allow IP forwarding.

Note that there are many distribution-specific ways to configure this, and there might be higher-level firewall rule management software available like `ufw`. You should configure firewall rules matching your distribution and setup.

To allow IP forwarding from and to all interfaces globally in a reboot-safe way, you may put a line like this in /etc/sysctl.conf:

```
net.ipv4.ip_forward=1
```

To do the same in an ad-hoc way that is not reboot safe but takes effect immediately:

```
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

It is also possible to instruct the firewall to allow IP forwarding for specific network devices only. For example, on a Debian based system, place an nft ruleset like this in `/etc/nftables.conf`:

```
define gtp_netdevs = { eth0, eth23 };

table inet filter {
	chain forward {
		type filter hook forward priority filter; policy drop;
		iifname $gtp_netdevs oifname $gtp_netdevs udp dport 2152 accept
	}
}
```

This ruleset allows IP forwarding, but limited to the GTP-U port 2152, and to two specific network devices eth0 and eth23.

6 Local GTP Addresses / Network Instance

PFCP features optional Network Instance IEs, in which the CPF may tell the UPF which local network interface to use for a PDR and/or a FAR.

NOTE

osmo-upf only evaluates the Network Instances configured in PDRs. Since osmo-upf always pairs a PDR+FAR with another PDR+FAR in reverse direction, each side's PDR is sufficient.

Network Instance IEs affect both the tunend and the tunmap use cases, as well as which local IP address is returned in the PFCP response

1. Look up Network Instance name in the osmo-upf.cfg `netinst` section, to obtain a local IP address.
2. Depending on use case:
   - tunend: create the tunnel on a GTP device matching the local IP address, see Section 5.6.

- tunmap: use the local IP address in the netfilter ruleset, see Section 5.7.

3. Usually, return the chosen local IP address in the F-TEID IE of the Created PDR IE in the PFCP response.

Network Instance configuration consists of {name, IP address} pairs.

NOTE
As soon as a `netinst` configuration is nonempty, receiving an undefined Network Instance name results in a PFCP Reject response, and a log message on cateogry `session`, level `NOTICE`. To make the PFCP return success, add the failing name to the `netinst` config.

6.1 netinst for tunend

The following configuration sets up two GTP devices for tunend, expecting Network Instance names `access1` or `access2`:

```
tunend
 dev create apn1 10.0.0.1
 dev create apn2 10.0.0.2
netinst
 add access1 10.0.0.1
 add access2 10.0.0.2
```

For example, if a Create PDR IE indicates Network Instance = `access1`, a GTP tunnel is set up in GTP kernel device `apn1`. For `access2`, use `apn2`.

6.2 netinst for tunmap

For the tunmap use case, it is sufficient to configure `netinst` entries, without any addition to the `tunmap` section. The following example configures various interfaces for tunmap, to match Network Instance names received in PFCP:

```
tunmap
 table-name osmo-upf
netinst
 add access1 10.0.0.1
 add access2 10.0.0.2
 add core1 9.0.0.1
 add core2 9.0.0.2
```

For example, a Create PDR indicating a Network Instance of `core1` will result in an nftables rule that receives packets on local address `9.0.0.1`.

7 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
`<0-10>`	`5`	A number from a range

7.1 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.

7.2 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.

7.3 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.

7.3.1 The question-mark (?) command

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

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

Example: Typing ? at start of OsmoMSC prompt

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

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

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

Example: Typing ? after a partial command

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

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

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

Example: Typing ? after show bsc

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

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

7.3.2 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.

7.3.3 The `list` command

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

Example: Typing list at start of OsmoMSC VIEW node prompt

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



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

Tip

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

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

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

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

7.3.4 The attribute system

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

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

Example: Typing show vty-attributes at the VTY prompt

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

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

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

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

Example: Typing list with-flags at the VTY prompt

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

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

❶ This command has no attributes assigned.

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

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

❹, ❺ This command applies immediately.

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

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

7.3.5 The expert mode

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

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

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

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

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

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

❶ This command enables the *expert* mode.

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

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

8 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.

8.1 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 ?`

8.2 Log levels

For each of the log categories (see Section 8.1), 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.

8.3 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.

8.4 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`

8.5 Log targets

Each of the log targets represent certain destination for log messages. It can be configured independently by selecting levels (see Section 8.2) for categories (see Section 8.1) as well as filtering (see Section 8.4) and other options like `logging timestamp` for example.

8.5.1 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 8.1 for more details on categories and Section 8.2 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 8.4.

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`

8.5.2 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.

8.5.3 Logging via gsmtap

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

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

It's configured as follows:

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

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

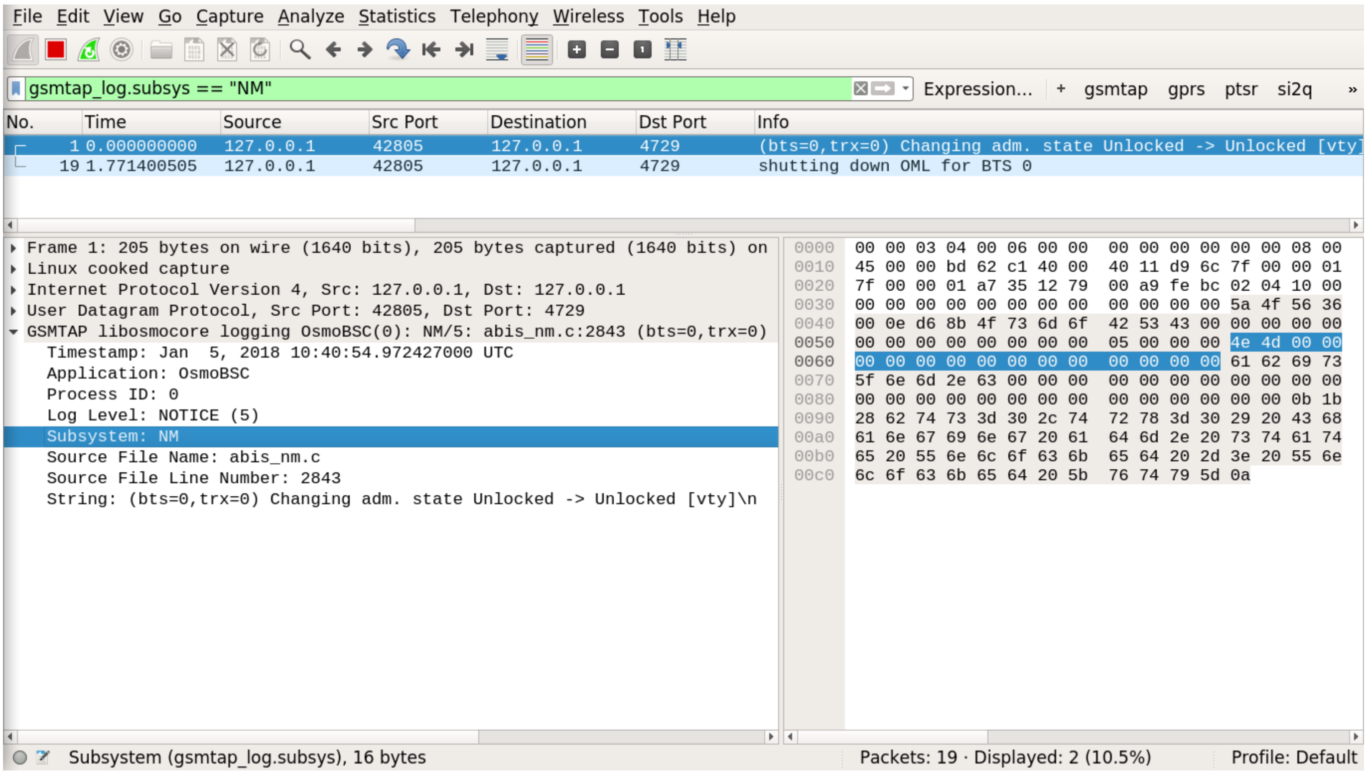

Figure 4: Wireshark with logs delivered over GSMTAP

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

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


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

8.5.4 Logging to a file

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

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

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

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

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

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

8.5.5 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.

8.5.6 Logging to systemd-journal

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

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

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

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

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

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

You can configure systemd based logging in two ways:

Example: systemd-journal target with offloaded rendering

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

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

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

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

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

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

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

Example: systemd-journal target with libosmocore based rendering

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

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

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

8.5.7 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)#
```

9 Osmocom Counters

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

9.1 Osmo Counters (deprecated)

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

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

9.2 Rate Counters

Rate counters count rates of events.

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

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

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

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

9.3 Stat Item

Stat items are a grouped replacement for osmo counters.

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

9.4 Statistic Levels

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

9.4.1 Global

These are global statistics.

9.4.2 Peer

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

This level also includes reporting global statistics.

9.4.3 Subscriber

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

This level also includes global and peer-based statistics.

9.5 Stats Reporter

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

9.5.1 Configuring a stats reporter

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

Example 9.1 Log statistics

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

❶ The interval determines how often the statistics are reported.

❷ Write the statistic information to any configured log target.

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

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

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

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

Example 9.2 Report statistics to statsd

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

❶ Configure the statsd reporter.

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

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

❹ IP address of the statsd server.

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

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

9.6 Socket stats

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

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

9.6.1 Configuration

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

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

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

Example 9.3 Report statistics to statsd

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

❶ Set the gathering interval (sec.)

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

9.6.2 Generated stats items

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

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

Example 9.4 VTY output of a stats item group of a TCP connection

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

10 Osmocom Control Interface

The VTY interface as described in Section 7 is aimed at human interaction with the respective Osmocom program.

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

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

10.1 Control Interface Protocol

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

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

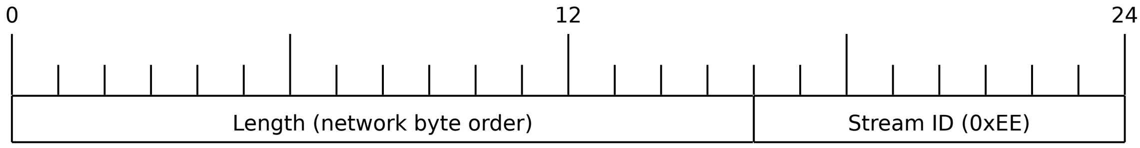

Figure 5: IPA header for control protocol

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

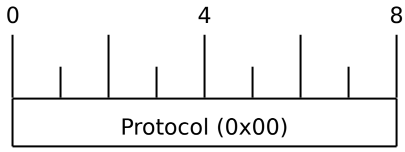

Figure 6: IPA extension header for control protocol

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

The fields specified below follow the following meaning:

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

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

<val>
The value of the variable / field

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

10.1.1 GET operation

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

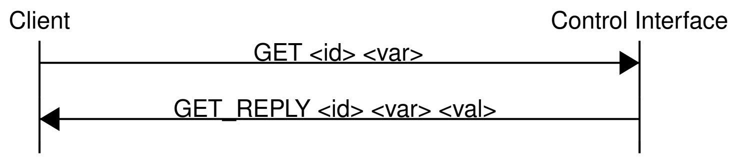

Figure 7: Control Interface GET operation (successful outcome)

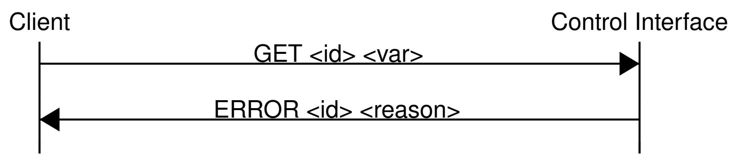

Figure 8: Control Interface GET operation (unsuccessful outcome)

10.1.2 SET operation

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

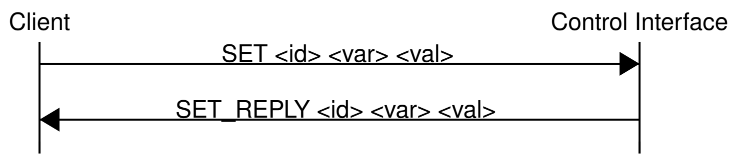

Figure 9: Control Interface SET operation (successful outcome)

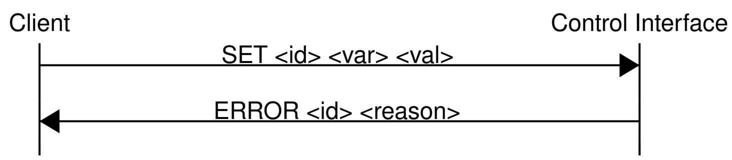

Figure 10: Control Interface SET operation (unsuccessful outcome)

10.1.3 TRAP operation

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

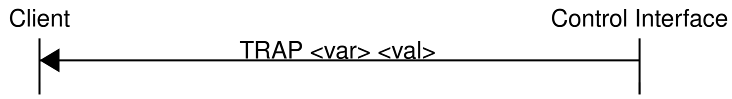

Figure 11: Control Interface TRAP operation

10.2 Common variables

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

Table 2: Variables available over control interface

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

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

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

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

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

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

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

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

10.3 Control Interface python examples

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

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

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

Internally it uses "`rate_ctr.*`" variable described in Section 10.2 to get the list of counter groups and than request all the counters in each group. Applications interested in individual metrics can request it directly using `rate_ctr2csv.py` as an example.

10.3.1 Getting rate counters

Example: Use rate_ctr2csv.py to get rate counters from OsmoBSC

```
$ ./scripts/osmo_rate_ctr2csv.py --header
Connecting to localhost:4249...
Getting rate counter groups info...
"group","counter","absolute","second","minute","hour","day"
"e1inp.0","hdlc:abort","0","0","0","0","0"
"e1inp.0","hdlc:bad_fcs","0","0","0","0","0"
"e1inp.0","hdlc:overrun","0","0","0","0","0"
"e1inp.0","alarm","0","0","0","0","0"
"e1inp.0","removed","0","0","0","0","0"
"bsc.0","chreq:total","0","0","0","0","0"
"bsc.0","chreq:no_channel","0","0","0","0","0"
...
"msc.0","call:active","0","0","0","0","0"
"msc.0","call:complete","0","0","0","0","0"
"msc.0","call:incomplete","0","0","0","0","0"
Completed: 44 counters from 3 groups received.
```

10.3.2 Setting a value

Example: Use osmo_ctrl.py to set the short network name of OsmoBSC

```
$ ./osmo_ctrl.py -d localhost -s short-name 32C3
Got message: SET_REPLY 1 short-name 32C3
```

10.3.3 Getting a value

Example: Use osmo_ctrl.py to get the mnc of OsmoBSC

```
$ ./osmo_ctrl.py -d localhost -g mnc
Got message: GET_REPLY 1 mnc 262
```

10.3.4 Listening for traps

You can use `osmo_ctrl.py` to listen for traps the following way:

Example: Using osmo_ctrl.py to listen for traps:

```
$ ./osmo_ctrl.py -d localhost -m
❶
```

❶ the command will not return and wait for any TRAP messages to arrive

11 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.

11.1 Scheduling Policy

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

Example: Set process to use RR scheduler

```
cpu-sched
 policy rr 1 ❶
```

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

11.2 CPU-Affinity Mask

Most operating systems allow for some sort of configuration on restricting the amount of CPUs a given process or thread can run on. The procedure is sometimes called as *cpu-pinning* since it allows to keep different processes pinned on a subset of CPUs to make sure the scheduler won't run two CPU-hungry processes on the same CPU.

The set of CPUs where each thread is allowed to run on is expressed by means of a bitmask in hexadecimal representation, where the right most bit relates to CPU 0, and the Nth most significant bit relates to CPU *N-1*. Setting the bit means the process is allowed to run on that CPU, while clearing it means the process is forbidden to run on that CPU.

Hence, for instance a cpu-affinity mask of *0x00* means the thread is not allowed on any CPU, which will cause the thread to stall until a new value is applied. A mask of *0x01* means the thread is only allowed to run on the 1st CPU (CPU 0). A mask of *0xff00* means CPUs 8-15 are allowed, while 0-7 are not.

For single-threaded processes (most of Osmocom are), it is usually enough to set this line in VTY config file as follows:

```
cpu-sched
 cpu-affinity self 0x01 ❶
```

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

Or otherwise:

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

❶ Allow all threads only on CPU 0

For multi-threaded processes, it may be desired to run some threads on a subset of CPUs while another subset may run on another one. In order to identify threads, one can either use the TID of the thread (each thread has its own PID in Linux), or its specific Thread Name in case it has been set by the application.

The related information on all threads available in the process can be listed through VTY. This allows identifying quickly the different threads, its current cpu-affinity mask, etc.

Example: Get osmo-trx Thread list information from VTY

```
OsmoTRX> show cpu-sched threads
Thread list for PID 338609:
 TID: 338609, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338610, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338611, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338629, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338630, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338631, NAME: 'osmo-trx-uhd', cpu-affinity: 0x3
 TID: 338634, NAME: 'UHDAsyncEvent', cpu-affinity: 0x3
```

```
 TID: 338635, NAME: 'TxLower', cpu-affinity: 0x3
 TID: 338636, NAME: 'RxLower', cpu-affinity: 0x3
 TID: 338637, NAME: 'RxUpper0', cpu-affinity: 0x3
 TID: 338638, NAME: 'TxUpper0', cpu-affinity: 0x3
 TID: 338639, NAME: 'RxUpper1', cpu-affinity: 0x3
 TID: 338640, NAME: 'TxUpper1', cpu-affinity: 0x3
```

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

Example: Set CPU-affinity from VTY telnet interface

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

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

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

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

Example: Set CPU-affinity from VTY config file

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

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

12 Glossary

2FF
: 2nd Generation Form Factor; the so-called plug-in SIM form factor

3FF
: 3rd Generation Form Factor; the so-called microSIM form factor

3GPP
: 3rd Generation Partnership Project

4FF
: 4th Generation Form Factor; the so-called nanoSIM form factor

A Interface
: Interface between BTS and BSC, traditionally over E1 (*3GPP TS 48.008* [3gpp-ts-48-008])

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

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

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

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

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

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

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

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

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

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

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

BTS
: Base Transceiver Station

BSC
: Base Station Controller

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

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

BVCI
: BSSGP Virtual Circuit Identifier

CBC
: Cell Broadcast Centre; central entity of Cell Broadcast service

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

CBS
: Cell Broadcast Service

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

CC
: Call Control; Part of the GSM Layer 3 Protocol

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

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

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

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

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

dB
: deci-Bel; relative logarithmic unit

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

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

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

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

DSP
: Digital Signal Processor

dvnixload
: Tool to program UBL and the Bootloader on a sysmoBTS

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

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

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

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

ETSI
: European Telecommunications Standardization Institute

FPGA
: Field Programmable Gate Array; programmable digital logic hardware

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

GERAN
: GPRS/EDGE Radio Access Network

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

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

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

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

Gp
: Gp interface between SGSN and GGSN; uses GTP protocol

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

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

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

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

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

GT
: Global Title; an address in SCCP

GTP
: GPRS Tunnel Protocol; used between SGSN and GGSN

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

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

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

IE
: Information Element

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

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

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

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

IPA
: *ip.access GSM over IP* protocol; used to multiplex a single TCP connection

Iu
: Interface in 3G/UMTS between RAN and CN

IuCS
: Iu interface for circuit-switched domain. Used in 3G/UMTS between RAN and MSC

IuPS
: Iu interface for packet-switched domain. Used in 3G/UMTS between RAN and SGSN

LAC
Location Area Code; 16bit identifier of Location Area within network

LAPD
Link Access Protocol, D-Channel (*ITU-T Q.921* [itu-t-q921])

LAPDm
Link Access Protocol Mobile (*3GPP TS 44.006* [3gpp-ts-44-006])

LLC
Logical Link Control; GPRS protocol between MS and SGSN (*3GPP TS 44.064* [3gpp-ts-44-064])

Location Area
Location Area; a geographic area containing multiple BTS

LU
Location Updating; can be of type IMSI-Attach or Periodic. Procedure that indicates a subscriber's physical presence in a given radio cell.

M2PA
MTP2 Peer-to-Peer Adaptation; a SIGTRAN Variant (*RFC 4165* [ietf-rfc4165])

M2UA
MTP2 User Adaptation; a SIGTRAN Variant (*RFC 3331* [ietf-rfc3331])

M3UA
MTP3 User Adaptation; a SIGTRAN Variant (*RFC 4666* [ietf-rfc4666])

MCC
Mobile Country Code; unique identifier of a country, e.g. 262 for Germany

MFF
Machine-to-Machine Form Factor; a SIM chip package that is soldered permanently onto M2M device circuit boards.

MGW
Media Gateway

MM
Mobility Management; part of the GSM Layer 3 Protocol

MNC
Mobile Network Code; identifies network within a country; assigned by national regulator

MNCC
Mobile Network Call Control; Unix domain socket based Interface between MSC and external call control entity like osmo-sip-connector

MNO
Mobile Network Operator; operator with physical radio network under his MCC/MNC

MO
Mobile Originated. Direction from Mobile (MS/UE) to Network

MS
Mobile Station; a mobile phone / GSM Modem

MSC
Mobile Switching Center; network element in the circuit-switched core network

MSC pool
A number of redundant MSCs serving the same core network, which a BSC / RNC distributes load across; see also the "MSC Pooling" chapter in OsmoBSC's user manual [userman-osmobsc] and *3GPP TS 23.236* [3gpp-ts-23-236]

MSISDN
: Mobile Subscriber ISDN Number; telephone number of the subscriber

MT
: Mobile Terminated. Direction from Network to Mobile (MS/UE)

MTP
: Message Transfer Part; SS7 signaling protocol (*ITU-T Q.701* [itu-t-q701])

MVNO
: Mobile Virtual Network Operator; Operator without physical radio network

NCC
: Network Color Code; assigned by national regulator

NITB
: Network In The Box; combines functionality traditionally provided by BSC, MSC, VLR, HLR, SMSC functions; see OsmoNITB

NRI
: Network Resource Indicator, typically 10 bits of a TMSI indicating which MSC of an MSC pool attached the subscriber; see also the "MSC Pooling" chapter in OsmoBSC's user manual [userman-osmobsc] and *3GPP TS 23.236* [3gpp-ts-23-236]

NSEI
: NS Entity Identifier

NVCI
: NS Virtual Circuit Identifier

NWL
: Network Listen; ability of some BTS to receive downlink from other BTSs

NS
: Network Service; protocol on Gb interface (*3GPP TS 48.016* [3gpp-ts-48-016])

OCXO
: Oven Controlled Crystal Oscillator; very high precision oscillator, superior to a VCTCXO

OML
: Operation & Maintenance Link (ETSI/*3GPP TS 52.021* [3gpp-ts-52-021])

OpenBSC
: Open Source implementation of GSM network elements, specifically OsmoBSC, OsmoNITB, OsmoSGSN

OpenGGSN
: Open Source implementation of a GPRS Packet Control Unit

OpenVPN
: Open-Source Virtual Private Network; software employed to establish encrypted private networks over untrusted public networks

Osmocom
: Open Source MObile COMmunications; collaborative community for implementing communications protocols and systems, including GSM, GPRS, TETRA, DECT, GMR and others

OsmoBSC
: Open Source implementation of a GSM Base Station Controller

OsmoNITB
: Open Source implementation of a GSM Network In The Box, combines functionality traditionally provided by BSC, MSC, VLR, HLR, AUC, SMSC

OsmoSGSN
Open Source implementation of a Serving GPRS Support Node

OsmoPCU
Open Source implementation of a GPRS Packet Control Unit

OTA
Over-The-Air; Capability of operators to remotely reconfigure/reprogram ISM/USIM cards

PC
Point Code; an address in MTP

PCH
Paging Channel on downlink Um interface; used by network to page an MS

PCP
Priority Code Point (*IEEE 802.1Q* [?])

PCU
Packet Control Unit; used to manage Layer 2 of the GPRS radio interface

PDCH
Packet Data Channel on Um interface; used for GPRS/EDGE signalling + user data

PIN
Personal Identification Number; a number by which the user authenticates to a SIM/USIM or other smart card

PLMN
Public Land Mobile Network; specification language for a single GSM network

PUK
PIN Unblocking Code; used to unblock a blocked PIN (after too many wrong PIN attempts)

RAC
Routing Area Code; 16bit identifier for a Routing Area within a Location Area

RACH
Random Access Channel on uplink Um interface; used by MS to request establishment of a dedicated channel

RAM
Remote Application Management; Ability to remotely manage (install, remove) Java Applications on SIM/USIM Card

RF
Radio Frequency

RFM
Remote File Management; Ability to remotely manage (write, read) files on a SIM/USIM card

Roaming
Procedure in which a subscriber of one network is using the radio network of another network, often in different countries; in some countries national roaming exists

Routing Area
Routing Area; GPRS specific sub-division of Location Area

RR
Radio Resources; Part of the GSM Layer 3 Protocol

RSL
Radio Signalling Link (*3GPP TS 48.058* [3gpp-ts-48-058])

RTP
Real-Time Transport Protocol (*IETF RFC 3550* [ietf-rfc3550]); Used to transport audio/video streams over UDP/IP

SACCH
: Slow Associate Control Channel on Um interface; bundled to a TCH or SDCCH, used for signalling in parallel to active dedicated channel

SCCP
: Signaling Connection Control Part; SS7 signaling protocol (*ITU-T Q.711* [itu-t-q711])

SDCCH
: Slow Dedicated Control Channel on Um interface; used for signalling and SMS transport in GSM

SDK
: Software Development Kit

SGs
: Interface between MSC (GSM/UMTS) and MME (LTE/EPC) to facilitate CSFB and SMS.

SGSN
: Serving GPRS Support Node; Core network element for packet-switched services in GSM and UMTS.

SIGTRAN
: Signaling Transport over IP (*IETF RFC 2719* [ietf-rfc2719])

SIM
: Subscriber Identity Module; small chip card storing subscriber identity

Site
: A site is a location where one or more BTSs are installed, typically three BTSs for three sectors

SMPP
: Short Message Peer-to-Peer; TCP based protocol to interface external entities with an SMSC

SMSC
: Short Message Service Center; store-and-forward relay for short messages

SS7
: Signaling System No. 7; Classic digital telephony signaling system

SS
: Supplementary Services; query and set various service parameters between subscriber and core network (e.g. USSD, 3rd-party calls, hold/retrieve, advice-of-charge, call deflection)

SSH
: Secure Shell; *IETF RFC 4250* [ietf-rfc4251] to 4254

SSN
: Sub-System Number; identifies a given SCCP Service such as MSC, HLR

STP
: Signaling Transfer Point; A Router in SS7 Networks

SUA
: SCCP User Adaptation; a SIGTRAN Variant (*RFC 3868* [ietf-rfc3868])

syslog
: System logging service of UNIX-like operating systems

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

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

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

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

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

TRX
Transceiver; element of a BTS serving a single carrier

TS
Technical Specification

u-Boot
Boot loader used in various embedded systems

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

UBL
Initial bootloader loaded by the TI Davinci SoC

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

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

Um interface
U mobile; Radio interface between MS and BTS

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

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

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

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

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

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

VLR
Visitor Location Register; volatile storage of attached subscribers in the MSC

VPLMN
Visited PLMN; the network in which the subscriber is currently registered; may differ from HPLMN when on roaming

VTY
Virtual TeletYpe; a textual command-line interface for configuration and introspection, e.g. the OsmoBSC configuration file as well as its telnet link on port 4242

A Osmocom TCP/UDP Port Numbers

The Osmocom GSM system utilizes a variety of TCP/IP based protocols. The table below provides a reference as to which port numbers are used by which protocol / interface.

Table 3: 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
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

Table 3: (continued)

L4 Protocol	Port Number	Purpose	Software
TCP	4275	telnet (VTY)	osmo-upf
TCP	4276	Control Interface	osmo-upf
TCP	4277	telnet (VTY)	osmo-pfcp-tool
TCP	4278	Control Interface	osmo-pfcp-tool
UDP	4729	GSMTAP	Almost every osmocom project
TCP	5000	A/IP	osmo-bsc, osmo-bsc_nat
UDP	23000	GPRS-NS over IP default port	osmo-pcu, osmo-sgsn, osmo-gbproxy
TCP	48049	BSC-CBC (CBSP) default port	osmo-bsc, osmo-cbc

B Bibliography / References

B.0.0.0.1 References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[73] [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

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

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

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

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

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

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

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

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

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

[83] [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/

[84] [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/

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

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

[87] [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

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

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

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

C GNU Free Documentation License

Version 1.3, 3 November 2008

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

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

C.1 PREAMBLE

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

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

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

C.2 APPLICABILITY AND DEFINITIONS

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

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

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

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

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

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

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

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

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

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

C.3 VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section Section C.4.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

C.4 COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

C.5 MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

a. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

b. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

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

d. Preserve all the copyright notices of the Document.

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

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

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

h. Include an unaltered copy of this License.

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

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

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

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

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

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

o. Preserve any Warranty Disclaimers.

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

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

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

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

C.6 COMBINING DOCUMENTS

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

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

C.7 COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

C.8 AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

C.9 TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

C.10 TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

C.11 FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

C.12 RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

C.13 ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with… Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.