# Contents

1 General Information ............................................. 7
   1.1 Scope and Audience ........................................ 8
   1.2 License and Copyright ...................................... 8
   1.3 Disclaimers .................................................. 8
   1.4 Source Code Availability ................................... 10
   1.5 Abbreviations ............................................... 11
   1.6 References .................................................. 12
   1.7 Contact Information & Support .............................. 13

2 Introduction to OpenBTS Application Suite ..................... 14
   2.1 Key Programs ............................................... 15
   2.2 Network Organization ..................................... 16

3 Getting to Know Your OpenBTS System ........................... 19
   3.1 Accessing the System ....................................... 19
   3.2 Starting and Stopping Applications ......................... 20
   3.3 OpenBTS Command Line Interface (CLI) ..................... 20
   3.4 Using the OpenRANUI ....................................... 24
   3.5 Databases ................................................... 25
   3.6 Folder Structure ............................................ 25
   3.7 Logging ...................................................... 26

4 OpenBTS Data Tables and Structures ............................ 27
   4.1 Manipulating OpenBTS Databases ........................... 27
   4.2 The Configuration Table ................................... 28
## Contents

### 11 PowerScanner

#### Appendices

**A** OpenBTS Implementation of GSM & 3GPP Specifications and IETF Standards

- A.1 The OpenBTS GSM Air Interface .................................................. 64
- A.2 Text Messaging (SMS) ................................................................. 72
- A.3 Short Message Service Cell Broadcast (SMSCB) ................................. 76
- A.4 General Packet Radio Service (GPRS) ............................................ 77

**B** Configuration Parameters

- B.1 OpenBTS Parameters ................................................................. 85
- B.2 Smqueue Parameters .................................................................. 98
- B.3 SIPAuthServe Parameters .......................................................... 101

**C** The Command Line Interface (CLI) Reference

- C.1 “alarms” Command .................................................................... 103
- C.2 “audit” Command ....................................................................... 103
- C.3 “calls” Command ....................................................................... 103
- C.4 “cellid” Command ....................................................................... 104
- C.5 “chans” Command ....................................................................... 104
- C.6 “config” Commands .................................................................... 105
- C.7 “devconfig” Command ................................................................. 106
- C.8 “endcall” Command ................................................................... 106
- C.9 “freqcorr” Command ................................................................... 107
- C.10 “gprs” Command ....................................................................... 107
- C.11 “handover” Command ................................................................. 108
- C.12 “help” Command ...................................................................... 108
- C.13 “load” Command ...................................................................... 108
- C.14 “memstat” Command .................................................................. 109
- C.15 “neighbors” Command ............................................................... 109
- C.16 “noise” Command ..................................................................... 109
- C.17 “notices” Command .................................................................. 109
## CONTENTS

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D Logging in OpenBTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>116</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E Multi-BTS Networks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>118</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E.1 How Mobility Works</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>118</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E.2 Example of Mobility Configuration, Simple Case</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E.3 Example of Mobility Configuration, More Advanced Case</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>122</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E.4 Handover</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>123</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E.5 Remote Logging</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>126</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F Test Procedures</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>130</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F.1 Test SIM Procedures</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>130</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F.2 Testing with Open Registration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>132</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## FOSS Licenses

| G.1 | Creative Commons License | 134 |
| G.2 | GNU General Public License, v.1 | 139 |
| G.3 | GNU General Public License, v.2 | 144 |
| G.4 | GNU General Public License, v.3 | 152 |
| G.5 | GNU Lesser General Public License, v.2.1 | 166 |
| G.6 | GNU Lesser General Public License, v.3 | 176 |
| G.7 | BSD Licenses | 178 |
Chapter 1

General Information

1.1 Scope and Audience ................................. 8
1.2 License and Copyright ................................ 8
1.3 Disclaimers ......................................... 8
  1.3.1 Warranty ......................................... 8
  1.3.2 Accuracy ........................................ 8
  1.3.3 Patent Laws ...................................... 8
  1.3.4 Trademarks ....................................... 9
  1.3.5 Telecom and Radio Spectrum Laws .......... 9
  1.3.6 FOSS License Compliance ..................... 9
1.4 Source Code Availability ............................ 10
1.5 Abbreviations ....................................... 11
1.6 References ......................................... 12
  1.6.1 References to ETSI Documents ............... 12
  1.6.2 ETSI/3GPP ...................................... 12
  1.6.3 IETF ............................................ 12
1.7 Contact Information & Support ...................... 13
  1.7.1 Direct Contact .................................. 13
  1.7.2 Online Resources ............................... 13

The OpenBTS Release 4.0 Public is distributed publicly under the AGPLv3 license. The OpenBTS Release 4.0 Commercial is distributed only to commercial customers, normally under a binary license. The commercial release provides the following improvements over the public release:

- Prioritization of emergency calls and IMS-compliant emergency call headers;
- All applications in the OpenBTS application suite are configured as upstart services. They start during system initialization, and restart automatically should either one crash or exit;
Remote management interface (OpenRANUI) over http, integrating subscriber provisioning, SIM card creation, configuration and monitoring tools for all major components of the application suite;

No AGPLv3 license restrictions.

1.1 Scope and Audience

This document describes the organization and operation of the OpenBTS Release 4.0 series GSM access point software. It is intended for use by software developers and network operators.

1.2 License and Copyright

Copyrights to this document are held by Range Networks, Inc. It is distributed under a Creative Commons Attribution-ShareAlike 3.0 Unported License. See the Appendix "Creative Commons License" for more information.

1.3 Disclaimers

1.3.1 Warranty

The OpenBTS software and its associated documentation are provided with NO WARRANTY OF ANY KIND. Although the information in this handbook has been carefully checked for accuracy, and is believed to be correct and current, no warranty, either express or implied, is made as to either its applicability to, or its compatibility with, specific requirements; nor does Range Networks, Inc. assume any responsibility for correctness of this information, nor for damages consequent to its use. All design characteristics and specifications are subject to change without notice.

Range Networks, Inc. welcomes any reports of software failures or documentation errors and will make reasonable efforts to correct these in future releases.

1.3.2 Accuracy

This manual is a description of the OpenBTS software, not a specification. In any discrepancy between the software and this manual, the software source code is authoritative.

1.3.3 Patent Laws

The use of this software to provide GSM services may result in the use of patented technologies. The user of this software is required to take whatever actions are necessary to avoid patent infringement.
1.3.4 Trademarks

“OpenBTS” is a registered trademark of Range Networks, Inc. (Range), a US corporation. Range reserves the right to control the use of this trademark. Do not use this trademark in commerce without permission and do not rebrand OpenBTS under a different trademark.

“Asterisk” is a trademark of Digium, Inc.

1.3.5 Telecom and Radio Spectrum Laws

Users of this software are expected to comply with all applicable local, national and international regulations.

1.3.6 FOSS License Compliance

OpenBTS Release 4.0 Commercial is distributed in binary form under a commercial license, with copyrights assigned to Range Networks, Inc., protected under copyright law.

OpenBTS Release 4.0 Public is distributed in source code form under APGLv3, protected under copyright law. This software is not ”public domain”. AGPLv3 is a commercial software license, enforceable like any other.

This section lists exceptions to the Range copyright acquired under FOSS licenses and describes OpenBTS compliance with those licenses.

LGPL Compliance

The following publicly-available packages are linked dynamically by applications in the OpenBTS suite under LGPL licenses:

- UnixODBC. Public distribution available from unixodbc.org. License is GNU Lesser General Public License, v.2.1.
- libc6. Public distribution available from . License is GNU Lesser General Public License, v.2.1.
- libusb-1.0-0-dev. Public distribution available from libusb.org. License is GNU Lesser General Public License, v.2.1.
- libzmq. Public distribution available from zeromq.org. License is GNU Lesser General Public License, v.3.

In each case, OpenBTS uses the public distribution without modification.

GPL Compliance

The following applications are distributed in Range Networks products and used in association with OpenBTS under GPL licenses:
• Asterisk 11.7.0. Public distribution available from asterisk.org. License is GNU General Public License, v.2.

• DataTables plug-in. Public distribution available from datatables.net. The plug-in is dual licensed under the GPLv2 license and a BSD 3-clause license.

• libreadline-dev. Public distribution available from ftp.gnu.org. License is GNU General Public License, v.3.

• PNotify plug-in. Public distribution available from pinesframework.org/pnotify/. The plug-in is triple licensed under the GPL, LGPL and MPL.

• pySim. Public distribution available from . License is GNU General Public License, v.2.

• SMQueue. Source code available from Range Networks upon request. License is GNU General Public License, v.3.

• Ubuntu Linux distribution v. 12.04. Public distribution available from ubuntu.com. The Ubuntu distribution includes a Linux kernel and a large number of utility applications distributed under a range of GPL licenses. OpenBTS and its related applications use the following components: the Linux kernel, apache/httpd, cat, erlang, ifconfig, iptables, logrotate, lshw, killall, ntp, openssl, rm, rsyslogd, screen, sh, ssh, wget. This list may not be exhaustive.

**BSD-style License Compliance**

The following software components are used in OpenBTS under BSD Licenses:


• Sparkline plug-in. Public distribution available from omnipotent.net/jquery.sparkline/. The plug-in is licensed under the 3-clause BSD License.

**MIT License Compliance**

The following software components are used in OpenBTS under the MIT license:

• jQuery library. Public distribution available from jquery.com.


### 1.4 Source Code Availability

The source-code of Release 4.0 Public is available at github.com/RangeNetworks/dev/releases/tag/v4.0.0.
1.5 Abbreviations

- APDU – application protocol data unit
- APN – Access Point Name
- ARFCN – absolute radio frequency channel number
- BTS – base transceiver subsystem
- dB – Decibels
- dBm – Decibel milliwatts
- FEC – forward error correction
- GPL – General Public License
- GPRS – General Packet Radio Service
- kHz – kilohertz
- LGPL – Lesser General Public License
- MOC – mobile-originated call
- MO-SMS – mobile-originated SMS
- MTC – mobile-terminated call
- MT-SMS – mobile-terminated SMS
- MS – mobile station
- PDU – protocol data unit
- RF – radio frequency
- RPDU – relay-layer protocol data unit
- RRLP – radio resource location protocol
- SIP – Session Initiation Protocol
- SMS – Short Message Service
- TDM – time-division multiplexing
- TPDU – transfer-layer protocol data unit
- USSD – unstructured supplementary service data
- W – Watts
CHAPTER 1. GENERAL INFORMATION

1.6 References

1.6.1 References to ETSI Documents

References to the Phase 2+ GSM specification series are given as “GSM xx.yy” where xx.yy is the specification number within the series. References to the 3GPP specification series are similarly given as “3GPP xx.yyy”. References to IETF specifications are given as “RFC-xxxx”. References to ITU-T specifications are given as “ITU-T xx.y”.

1.6.2 ETSI/3GPP

This document references the following GSM and 3GPP specifications, which can be downloaded for free from the etsi.org web site.

- GSM 03.38: “Digital cellular telecommunications system (Phase 2+); Alphabets and language-specific information”
- GSM 03.41: “Digital cellular telecommunications system (Phase 2+); Technical realization of Cell Broadcast Service (CBS)”
- GSM 04.06: “Digital cellular telecommunications system (Phase 2+); Mobile Station - Base Station System (MS - BSS) interface; Data Link (DL) layer specification”
- GSM 04.08: “Digital cellular telecommunications system (Phase 2+); Mobile radio interface layer 3 specification”
- GSM 05.02: “Digital cellular telecommunications system (Phase 2+); Multiplexing and multiple access on the radio path”
- GSM 05.03: “Digital cellular telecommunications system (Phase 2+); Channel coding”
- GSM 05.05: “Digital cellular telecommunications system (Phase 2+); Radio transmission and reception”
- GSM 05.08: “Digital cellular telecommunications system (Phase 2+); Radio subsystem link control”
- GSM 05.10: “Digital cellular telecommunications system (Phase 2+); Radio subsystem synchronization”
- 3GPP TS 24.008: “Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Mobile radio interface Layer 3 specification; Core network protocols; Stage 3”
- 3GPP 24.229: “Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3”

1.6.3 IETF

This document references the following IETF standards, which can be downloaded for free from the ietf.org web site.
1.7. CONTACT INFORMATION & SUPPORT

- RFC-2833: “RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals”
- RFC-2976: “The SIP INFO Method”
- RFC-3325: “Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks”
- RFC-3455: “Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP)”
- RFC-4119: “A Presence-based GEOPRIV Location Object Format”

1.7 Contact Information & Support

1.7.1 Direct Contact

For additional information or paid technical support for OpenBTS, please contact:

Range Networks, Inc.
560 Brannan Street
San Francisco, California 94107
United States of America

telephone +1 415-778-8700

e-mail support@rangenetworks.com

1.7.2 Online Resources

Customer Support System

A support agreement with Range Networks includes a subscription to the customer support web site, wiki and forum. For more information, contact support@rangenetworks.com.

Free Support

Free support is available through the OpenBTS public mailing list. See the openbts.org web site for more information.
Chapter 2

Introduction to OpenBTS Application Suite

2.1 Key Programs

2.1.1 OpenBTS

2.1.2 Transceiver

2.1.3 SMQueue

2.1.4 SIP router/PBX

2.1.5 SIPAuthServe

2.2 Network Organization

A complete OpenBTS installation comprises several distinct applications:

- **OpenBTS** – The actual OpenBTS application, containing most of the GSM stack above the radiomodem.
- **Transceiver** – The software radiomodem and hardware control interface.
- **SMQueue** – The RFC-3428 store-and-forward server for text messaging.
- **Asterisk** – The VoIP PBX or “softswitch”.
- **SIPAuthServe** – An application managing the database of subscriber information.
- **Other Services** – Optional services supported through external servers, interfaced to OpenBTS through various protocols. In OpenBTS these services are:
  - NodeManager, the JSON management API to allow third-party applications to manage the essential OpenBTS suite components (OpenBTS, SMQueue and SIPAuthServe). The OpenRANUI web interface provided with the Commercial release of OpenBTS uses NodeManager;
  - SMSCB, low-rate data service.

In Range preconfigured systems, all suite executables besides Asterisk are located in “/OpenBTS” folder on the root partition.
2.1 Key Programs

2.1.1 OpenBTS

The OpenBTS application contains:

- L1 TDM functions (GSM 05.02)
- L1 FEC functions (GSM 05.03)
- L1 closed loop power and timing controls (GSM 05.08 and 05.10)
- L2 LAPDm (GSM 04.06)
- L3 radio resource management functions (GSM 04.08)
- L3 GSM-SIP gateway for mobility management
- L3 GSM-SIP gateway for call control
- L4 GSM-SIP gateway for text messaging

The general design approach of OpenBTS is not to implement any function above L3 or L4, so at L3 or L4 every subprotocol of GSM is either terminated locally or translated through a gateway to some other protocol for handling by an external application. Similarly, OpenBTS itself does not contain any speech transcoding functions above the L1 FEC parts.

2.1.2 Transceiver

The transceiver application performs the radiomodem functions of GSM 05.05 and manages the USB interface to the radio hardware. The functions of the transceiver are described in Section A.1.1.

2.1.3 SMQueue

SMQueue is an RFC-3428 store-and-forward server that is used for text messaging in the OpenBTS system. SMQueue is required to send a text message from one MS to another, or to provide reliable delivery of text messages to an MS from any source. See Chapter 7 for more information.

2.1.4 SIP router/PBX

OpenBTS uses a SIP router or PBX to perform the call control functions that would normally be performed by the mobile switching center in a conventional GSM network, although in most network configurations this switching function is distributed over multiple switches. These switches also provide transcoding services.

As of OpenBTS Release 4.0, the standard SIP router is Asterisk 11. OpenBTS has been used with VoIP PBX and switch applications other than Asterisk, however Range does not normally support these configurations.

See Chapter 6 for information about integration between OpenBTS and Asterisk.

### 2.1.5 SIPAuthServe

An application that implements Subscriber Registry, the database of subscriber information that replaces both the Asterisk SIP registry and the GSM Home Location Register (HLR) found in a conventional GSM network. OpenBTS also relies on Asterisk for any transcoding functions. See Chapter 6 for information about SIPAuthServe, Subscriber Registry and their integration with OpenBTS and Asterisk.

### 2.2 Network Organization

In the simplest network, with a single access point, all of the applications in the suite run inside the access point on the same embedded computer. This is shown in Figure 2.1.

In a slightly larger network, with a small number of access points and good IP connectivity between them, one access point can be designated as a master and provide servers for the rest of the network. Figure 2.2 shows an example of a network of two access points, with one access point hosting the server for both. This is a simple configuration for small multi-BTS networks. It is also a preferred configuration for cell sites that co-locate multiple access points sharing a common network connection.

The OpenBTS and Transceiver applications must run inside each GSM/SIP access point. The Asterisk and the Subscriber Registry application (SIPAuthServe) communicate through the filesystem and therefore must run on the same computer, but that computer can be remote from the access point. SMQueue and the other servers can run anywhere and may have multiple instances.
2.2. NETWORK ORGANIZATION

Figure 2.1: Components of the OpenBTS application suite and their communication channels as installed in each access point. Sharp-cornered boxes are hardware components. Round-cornered boxes are software components.
CHAPTER 2. INTRODUCTION TO OPENBTS APPLICATION SUITE

"Transcevier"
Radiomodem

"OpenBTS"
GSM/SIP
Protocol Processor

smqueue
RFC-3428
SMS Processor

subscriber registry
Database/Server

SIP
HTTP/S
SIP/RTP
IAX

SIP/IAX
Softswitch

SIP
SMTP
SIP
HTTP/S
SIP/RTP

UDP
SIP
SIP/SQL
SIP/RTP
SIP/HTTP/S

"Transcevier"
Radiomodem

"OpenBTS"
GSM/SIP
Protocol Processor

Figure 2.2: Two access points with unit #1 providing servers for both.
Chapter 3

Getting to Know Your OpenBTS System

3.1 Accessing the System ............................................................... 19
3.2 Starting and Stopping Applications ............................................. 20
3.3 OpenBTS Command Line Interface (CLI) ....................................... 20
   3.3.1 Starting and Stopping the CLI .............................................. 20
   3.3.2 Using the CLI ................................................................. 21
   3.3.3 Using the CLI to Access a Remote Node ................................ 23
   3.3.4 Using the CLI in Development ............................................ 23
3.4 Using the OpenRANUI .............................................................. 24
   3.4.1 Starting and Stopping Applications of OpenBTS Suite .............. 24
   3.4.2 Configuring Applications of OpenBTS Suite .......................... 24
   3.4.3 Managing Subscribers ..................................................... 25
   3.4.4 Monitoring the Node ....................................................... 25
3.5 Databases ................................................................................. 25
3.6 Folder Structure ....................................................................... 25
3.7 Logging .................................................................................. 26

3.1 Accessing the System

All Range installations of OpenBTS Release 4.0 run Ubuntu Linux v12.04 operating system. The primary
interface to the system is the UNIX shell, accessed via ssh from the Ethernet connector on the front panel. The
system’s Ethernet interface is auto-sensing, requiring no hub or crossover cable. It is shipped with the default
IP address of “192.168.0.21”.

Once the unit is connected to another computer or a network on the 192.168.0.x subnet, it can be accessed via
ssh. The account is as follows:
CHAPTER 3. GETTING TO KNOW YOUR OPENBTS SYSTEM

Login: openbts
Password: openbts

The account is super-user (sudo) enabled. The command to issue on any UNIX system, including OS X, is:

```bash
ssh openbts@192.168.0.21
```

From a Windows machine an SSH client (like PuTTY) can be used.

**Note:** Make sure to change the password for user “openbts” before you connect the system to the public network.

### 3.2 Starting and Stopping Applications

In embedded configurations used in Range systems, all applications in the OpenBTS Application Suite are configured as upstart services (visit upstart.ubuntu.com for further details) and start during system initialization. Should either one crash or exit, it will automatically restart.

To manually start an application, log into the system and type at the command line:

```bash
sudo start <application_name>
```

To stop an application:

```bash
sudo stop <application_name>
```

To restart OpenBTS application, type:

```bash
sudo stop openbts
sudo start openbts
```

**Note:** This behavior is an enhancement over the previous versions of OpenBTS. The “runloop.sh” script is no longer in place.

### 3.3 OpenBTS Command Line Interface (CLI)

The OpenBTS console application is called the “command line interface” or CLI. The CLI allows you to monitor system status and change many operating parameters of OpenBTS and the Transceiver in real time.

#### 3.3.1 Starting and Stopping the CLI

Its executable is located at `/OpenBTS/OpenBTSCLI` on the file system. To launch the CLI, log into your access point via ssh and type on the Unix command line:
In order for the CLI to run, OpenBTS application must also be running. Once the CLI launches you will see the welcome notice and the OpenBTS command prompt:

OpenBTS Command Line Interface (CLI) utility
Copyright 2012-2014 Range Networks, Inc.
Licensed under GPLv2.
Includes libreadline, GPLv2.
Connecting to 127.0.0.1:49300...
Remote Interface Ready.
Type:
"help" to see commands,
"version" for version information,
"notices" for licensing information,
"quit" to exit console interface.
OpenBTS>

To exit the CLI, type the “quit” command and press Enter. Doing so does not stop the OpenBTS application. The CLI at this time does not have a command to stop OpenBTS, but it can restart OpenBTS with the “shutdown” command. Once the OpenBTS application shuts down, it starts automatically (provided it is configured as an upstart service), but loses connection to the CLI. Restart the CLI to connect to OpenBTS again.

3.3.2 Using the CLI

The CLI interface is simple: write a command string at the CLI prompt and press Enter, then read back the result string(s).

For a list of available commands type “help” at the CLI prompt and press Enter. Use “help” followed by a command name to get a description of a specific command:

OpenBTS> help rxgain
rxgain [newRxgain] -- get/set the RX gain in dB

OpenBTS>

A detailed list of CLI commands is provided in Appendix C.

Using the CLI to Configure the OpenBTS

The command to inspect and modify the OpenBTS configuration parameters is “config”. The parameters are case-sensitive. They are stored in the configuration table (see section 4.2). The complete list of configuration parameters and their meanings is provided in Appendix B.
The “config” command followed by a configuration parameter name provides full details on that parameter, including the current value, the default value, the range of acceptable values and whether the parameter is static or dynamic. For example:

```
OpenBTS> config GSM.Identity.ShortName
GSM.Identity.ShortName Range [default]
- description: Network short name, displayed on some phones. Optional but must be defined if you also want the network to send time-of-day.
- type: string
- default value: Range
- visibility level: customer site - these values are different for each BTS and should not be left default
- static: 0
- valid val regex: ^[0-9a-zA-Z]+$  
- scope: value must be the same across all nodes
```

The “config” command followed by a partial parameter name gives a list of all parameters with names containing that partial name. For example:

```
OpenBTS> config Identity
GSM.Identity.BSIC.BCC 1
GSM.Identity.BSIC.NCC 3
GSM.Identity.CI 10 [default]
GSM.Identity.LAC 1010
GSM.Identity.MCC 001 [default]
GSM.Identity.MNC 05
GSM.Identity.ShortName Range [default]
```

To set a parameter to a certain value, provide the parameter name followed by the new value to the “config” command like this:

```
OpenBTS> config GSM.Identity.ShortName MyCellularNetwork
GSM.Identity.ShortName changed from "Range" to "MyCellularNetwork"
```

To revert a parameter to the default value, use the “rmconfig” command like this:

```
OpenBTS> rmconfig GSM.Identity.ShortName
GSM.Identity.ShortName set back to its default value
```

OpenBTS> config GSM.Identity.ShortName
3.3. OPENBTS COMMAND LINE INTERFACE (CLI)

GSM.Identity.ShortName Range [default]
- description: Network short name, displayed on some phones. Optional but must be defined.
- type: string
- default value: Range
- visibility level: customer site - these values are different for each BTS and should not be left default
- static: 0
- valid val regex: ^[0-9a-zA-Z]+$ 
- scope: value must be the same across all nodes

OpenBTS>

To set a configuration parameter to an empty value, use the “unconfig” command. For example:

OpenBTS> unconfig Control.LUR.OpenRegistration
Control.LUR.OpenRegistration disabled

OpenBTS>

3.3.3 Using the CLI to Access a Remote Node

By default the OpenBTS CLI operates over a TCP socket on port 49300 of the 127.0.0.1 interface. These
defaults can be changed with the “CLI.Port” and “CLI.Interface” parameters using the “devconfig”
command to allow network access.

OpenBTS> devconfig CLI.Interface 192.168.0.42
CLI.Interface changed from "127.0.0.1" to "192.168.0.42"

OpenBTS>

The CLI can connect to OpenBTS running on a different IP and/or port. Provide the “-p” option to specify
an alternative port and the “-t” option to specify the IP address of a target system. For example:

> /OpenBTS/OpenBTSCLI -t 192.168.0.42 -p 49333

3.3.4 Using the CLI in Development

The CLI interface is simple: it reads a command string at the CLI socket and returns the result string(s).
Any number of outside applications can access the CLI at the same time. It can also execute commands in
non-interactive mode, making it possible to use the CLI capabilities in third party tools (e.g. shell scripts).

Two options offer slightly different modes of operation: the “-c” option followed by a command makes the
CLI execute the command, print the result to STDOUT and exit. No banners/welcome messages appear in the output.

> /OpenBTS/OpenBTSCLI -c cellid
MCC=001 MNC=01 LAC=1010 CI=10
Alternatively, the “-d” option starts the CLI and awaits a command on the STDIN, executes it, prints the result to STDOUT and exits.

**Note:** The “-d” option repeats the functionality of the “OpenBTSDo” script from the earlier versions of OpenBTS. The script is still available in the current release for backward compatibility.

### 3.4 Using the OpenRANUI

The commercial release of OpenBTS includes a web-enabled administrative interface called OpenRANUI. It allows you to monitor and configure your OpenBTS application suite. It is available via HTTP interface at the following location (provided that the BTS operates at the default “192.168.0.21” IP address), and can be accessed using any modern-time web browser:

```
http://192.168.0.21/ranui/
```

The login to access the OpenRANUI is “openbts” and the password matches the system password for the same account (“openbts” by default).

#### 3.4.1 Starting and Stopping Applications of OpenBTS Suite

The “Services” panel provides the system health information and allows to start and stop individual services. Click on the “manage node” button to reveal the version information for every component running on a node. The corresponding slides let you enable and disable the components. Click on “Save” to confirm your action. To minimize the node panel click on “done”.

#### 3.4.2 Configuring Applications of OpenBTS Suite

The “Configuration” panel provides easy access to the primary configuration parameters of the OpenBTS suite components, namely OpenBTS, SMQueue, SIPAuthServe and RangeSIMd. For ease of use they are grouped into “basic” and “advanced”.

**Note:** The OpenRANUI does not provide access to development, factory or custom parameters. Use the CLI to review and/or change them.

Indicators on the left show the status of these parameters:

- white indicator means that the parameter value is valid,
- yellow indicator means that the parameter has been modified, but is not saved yet,
- red indicator means an incorrect value (incorrect values cannot be saved), and
- grey indicator means that the parameter is set to the default value.

To reset a parameter to its default value, move your mouse over the default area and click on it.
3.5. DATABASES

The “audit” panel of the OpenRANUI provides an audit of the current OpenBTS configuration. It can help detect problems in the setup, though not all warning messages would necessarily point to a problem. In a multi-node configuration the “audit” panel would also do a cross-node check of values that need to be coordinated in the network.

3.4.3 Managing Subscribers

The “Subscribers” panel provides a searchable list of subscribers registered with the network, including their names and phone numbers. These parameters are tied to an IMSI, and can be changed as needed.

You can also manually add or remove a subscriber. Click on “add subscriber” and provide the necessary details, then click on “add” to save them.

If you have a Range Networks SIM-card writer (part #8425-xxx series) attached to your node, you can use it to generate COMP128v1 SIMs with known IMSIs and known Ki, supporting full RAND-SRES authentication. Click on “write SIM” and provide the necessary details, then click on “write”. The corresponding subscriber data will be added to the subscriber registry automatically.

3.4.4 Monitoring the Node

The “Monitor” panel provides the graphs representing the SDCCH and TCH channel loads, AGCH and PCH queue sizes, and the level of noise. With the “Monitor” panel selected, turn the toggle on the right to the “Enabled” position to initialize the monitors. Each graph covers the period of 5 minutes.

Finally, the “Databases” panel provides easy, searchable access to the run-time databases, discussed in full detail in Chapter 4.

3.5 Databases

OpenBTS uses a set of sqlite3 database files to store its configuration parameters and run-time data. For more information on sqlite3, see the www.sqlite.org web site.

Chapter 4 discusses the data structures and their use in full detail.

Note: Depending on your configuration, some or all OpenBTS databases will have write-ahead logging enabled.

3.6 Folder Structure

OpenBTS configuration databases are located in “/etc/OpenBTS”. Asterisk configuration files are located separately according to Asterisk setup (“/etc/asterisk” by default), and the Subscriber Registry database is located at “/var/lib/asterisk/sqlite3dir/sqlite3.db”.

OpenBTS run-time data tables are located in “/var/run/OpenBTS”. Their locations are defined in the configuration table and can be changed to meet the requirements of your project.
3.7 Logging

The OpenBTS logging destination is controlled by the logging mechanism called rsyslogd which records all OpenBTS log messages to “/var/log/OpenBTS.log”. It is configured with “/etc/rsyslog.d/OpenBTS.conf” as follows:

```
local7.* /var/log/OpenBTS.log
```

The rsyslogd mechanism offers a range of powerful archival, reporting and notification mechanisms, including email notifications and routing of log messages to remote sites for network monitoring. See the rsyslogd Unix manual pages for more information on these features.

OpenBTS logging level can be configured to provide the desired amount of detail, including on per-file basis. See Appendix D for further details.
Chapter 4

OpenBTS Data Tables and Structures

4.1 Manipulating OpenBTS Databases

The following methods can be used to view and modify the OpenBTS database files:

- **OpenBTS Command-Line Interface (CLI).** The OpenBTS CLI “config” and “unconfig” commands (Section 3.3) can be used to edit the OpenBTS configuration table (Section 4.2) in real time. Configuration changes from the CLI are written back to the “OpenBTS.db” database and are persistent. Some changes may require a restart of OpenBTS before they take effect.

- **OpenRANUI web interface.** This tool is only available in the Commercial version. All customer tuneable parameters can be modified via the web-interface from the “Configuration” panel. Additionally, the “Databases” panel provides read-only access with sorting and search capabilities to all major run-time tables.

The above-mentioned methods of manipulating the databases ensure the integrity of the OpenBTS-related databases and should be considered the methods of choice. In the laboratory or development environment, it might be occasionally useful to access the databases directly.

- **The sqlite3 tool.** The sqlite3 command line tool can be used to inspect and modify these databases using SQL syntax.
− The database can be manipulated directly using SQL syntax in real time.
− For offline editing, sqlite3 can export SQL code to a text file with the "dump" command. The text can then be edited and reimported with "read". Refer to sqlite3 manual for further details.

Please note, however, that this method of manipulating databases is not safe and may corrupt your databases. Understanding of OpenBTS data structures, sqlite3 database journaling and general SQL principles is required. It is recommended that a reserve copy of a configuration database is made prior to editing.

4.2 The Configuration Table

The parameters that control the OpenBTS application are stored in a database table called the configuration table, located at "/etc/OpenBTS/OpenBTS.db".

Some parameters are dynamic, meaning that a parameter change will have an immediate effect. Some of these parameters are static and changes to them do not take effect until OpenBTS is restarted. Some of these static parameters are matched to the hardware of a specific implementation and should not be changed at all. Comments within the configuration database describe each parameter and under what conditions it can be changed. Flags within the database schema indicate which parameters are static. The schema for the configuration table is:

CREATE TABLE CONFIG (
  KEYSTRING TEXT UNIQUE NOT NULL,
  VALUESTRING TEXT,
  STATIC INTEGER DEFAULT 0,
  OPTIONAL INTEGER DEFAULT 0,
  COMMENTS TEXT DEFAULT ''
)

Note that the database itself contains comments, available to the operator at all times and repeated in this manual.

To change a dynamic configuration parameter in real time, edit the table using one of the methods described at the beginning of this chapter. The effect will be immediate, although in-progress transactions may continue to use the parameters with which they started.

E.g.: A change in "SIP.Proxy.Speech" will not affect in-progress telephone calls, but any new calls will use the new proxy.

When changing a static configuration parameter, the OpenBTS application must be restarted (see section 3.2 for instructions) in order for the change to take effect.

For optional parameters an empty string "" is a valid value and normally means that the parameter has been disabled or is configured automatically. To set a parameter to an empty value, use the "unconfig" command in the CLI.

The complete list of configuration parameters is provided in Appendix B. The configuration parameters of particular importance are covered in the sections below.
4.3 TMSI Table

To reduce dependence on a backhaul link, OpenBTS tracks TMSIs internally on a per-cell basis. OpenBTS allocates a TMSI in the TMSI table for every MS that sends a Location Updating Request, whether the MS is allowed to register or not.¹

To accomplish this, OpenBTS tracks TMSI-IMSI relationships in an sqlite3 database table called the **TMSI table**, with the default location at “/var/run/OpenBTS/TMSITable.db” which is set in the “Control.Reporting.TMSITable” configuration parameter. The TMSI table is treated as read-write by OpenBTS but should be treated as read-only by other applications. In flash-based systems, this table should be stored in a ramdisk partition. The schema is:

```sql
CREATE TABLE IF NOT EXISTS TMSI_TABLE (  
TMSI INTEGER PRIMARY KEY, -- this value is used as the TMSI  
IMSI TEXT UNIQUE NOT NULL, -- IMSI of the SIM  
IMEI TEXT, -- IMEI of the MS, if requested  
CREATED INTEGER NOT NULL, -- Unix time (seconds) of record creation  
ACCESSED INTEGER NOT NULL, -- Unix time (seconds) of last encounter  
A5_SUPPORT INTEGER, -- encryption support in the MS, if requested  
POWER_CLASS INTEGER, -- power class of the MS, if requested  
OLD_TMSI INTEGER, -- previous TMSI from another cell or network  
OLD_MCC INTEGER, -- previous network MCC  
OLD_MNC INTEGER, -- previous network MNC  
OLD_LAC INTEGER, -- previous network LAC  
kcc varchar(33) default '', -- returned by the Registrar, needed for ciphering  
RRLP_STATUS INTEGER DEFAULT 0, -- whether or not MS supports RRLP  
DEG_LAT FLOAT, -- cached RRLP result  
DEG_LONG FLOAT, -- cached RRLP result  
ASSOCIATED_URI text default '', -- Saved from the SIP REGISTER message and inserted into MOC SIP INVITE  
ASSERTED_IDENTITY text default '', -- Saved from the SIP REGISTER message and inserted into MOC SIP INVITE  
WELCOME_SENT INTEGER DEFAULT 0, -- 0 == welcome message not sent yet; 1 == sent by us; 2 == sent by someone else  
AUTH INTEGER DEFAULT 0, -- Authorization result, 0 == unauthorized  
AUTH_EXPIRY INTEGER DEFAULT 0, -- Absolute time (seconds) when authorization expires or 0 for single-use  
REJECT_CODE INTEGER DEFAULT 0, -- Reject code, or 0 if authorized  
TMSI_ASSIGNED INTEGER DEFAULT 0, -- Set when the TMSI has been successfully assigned to the MS, ie, the MS knows it  
PTMSI_ASSIGNED INTEGER DEFAULT 0 -- Set when the P-TMSI has been successfully assigned to the MS by
```

¹In order for this scheme to work correctly, each BTS in a multi-BTS network must have a unique location area code.
the SGSN, i.e., the MS knows it

4.4 Transaction Table

OpenBTS reports in-progress calls and SMS transfers to an external sqlite3 database table called \textit{Transaction table}, with the default location at “/var/run/OpenBTS/TransactionTable.db” which is set in the “Control.Reporting.TransactionTable” configuration parameter.

This table is treated as write-only by OpenBTS but should be treated as read-only by other applications. The SQL schema is:

\begin{verbatim}
CREATE TABLE IF NOT EXISTS TRANSACTION_TABLE (  
  ID INTEGER PRIMARY KEY, -- internal transaction ID  
  CHANNEL TEXT DEFAULT NULL, -- channel description string (cross-referenced CHANNEL_TABLE)  
  CREATED INTEGER NOT NULL, -- Unix time of record creation  
  CHANGED INTEGER NOT NULL, -- Unix time of last state change  
  TYPE TEXT, -- transaction type  
  SUBSCRIBER TEXT, -- IMSI, if known  
  L3TI INTEGER, -- GSM L3 transaction ID, +8 if generated by MS  
  SIP_CALLID TEXT, -- SIP-side call id tag  
  SIP_PROXY TEXT, -- SIP proxy IP  
  CALLED TEXT, -- called party number, if known  
  CALLING TEXT, -- calling party number, if known  
  GSMSTATE TEXT, -- current GSM/Q.931 state  
  SIPSTATE TEXT -- current SIP state
)
\end{verbatim}

The “CHANNEL” column uses the same encoding as the “CN_TN_TYPE_AND_OFFSET” column in the Channel Table and can be used to cross-reference the two tables. It is a channel description string of the form:

$C<n>T<n> \langle\text{channelType}\rangle-\langle\text{subchannelIndex}\rangle$

For example:

- “C0T1 TCH/F” is a full-rate traffic channel on timeslot 1 of the C0 ARFCN, and
- “C0T0 SDCCH-0/4” is the #0 SDCCH (of 4 available) on C0T0.

4.5 Channel Table

OpenBTS reports real-time physical status information for active dedicated channels to an external sqlite3 database table called \textit{Channel Table} or \textit{PHYSTATUS}. The entry for a channel is updated every time a Measurement Report message is received on the channel’s associated SACCH. The table’s default location
is "/var/run/OpenBTS/ChannelTable.db" which is set in the "Control.Reporting.PhysStatusTable" configuration parameter.

This table is treated as write-only by OpenBTS but should be treated as read-only by other applications. The schema is:

```
CREATE TABLE IF NOT EXISTS PHYSTATUS (  
    CN_TN_TYPE_AND_OFFSET STRING PRIMARY KEY,  
    -- cross-refs TRANSACTION_TABLE; strings of the same format are used  
    -- in the CHANNEL column of the Transaction table to allow  
    -- cross-referencing,  
    ARFCN INTEGER DEFAULT NULL, -- the ARFCN of this channel  
    ACCESSED INTEGER DEFAULT 0, -- Unix time of the most recent update of the record  
    RXLEV_FULL_SERVING_CELL INTEGER DEFAULT NULL,  
    -- downlink RSSI in dBm as observed by the handset, averaged over  
    -- all timeslots on this ARFCN  
    RXLEV_SUB_SERVING_CELL INTEGER DEFAULT NULL,  
    -- downlink RSSI in dBm as observed by the handset for this timeslot  
    RXQUAL_FULL_SERVING_CELL_BER FLOAT DEFAULT NULL,  
    -- downlink BER in percent as observed by the handset, averaged  
    -- over all timeslots on this ARFCN  
    RXQUAL_SUB_SERVING_CELL_BER FLOAT DEFAULT NULL,  
    -- downlink BER in percent as observed by the handset for this timeslot  
    RSSI FLOAT DEFAULT NULL,  
    -- uplink RSSI at the BTS receiver, expressed in dB relative to full  
    -- scale input; this is normally within a few dB of the  
    -- GSM.Radio.RSSITarget parameter  
    TIME_ERR FLOAT DEFAULT NULL,  
    -- timing advance error in symbol periods; this is normally accurate  
    -- to better than 1/20 of a symbol period, reported at a resolution  
    -- of 1/256 of a symbol period.  
    TRANS_PWR INTEGER DEFAULT NULL,  
    -- MS transmitter power in dBm  
    TIME_ADVC INTEGER DEFAULT NULL,  
    -- MS timing advance in whole symbol periods  
    FER FLOAT DEFAULT NULL,  
    -- observed uplink FER for the channel  
    NCELL_ARFCN INTEGER DEFAULT NULL,  
    -- ARFCN of the strongest neighbor  
    NCELL_RSSI INTEGER DEFAULT NULL  
    -- RSSI measurement of the strongest neighbor)
```
Chapter 5

OpenBTS and the Transceiver

5.1 Downlink Power and Congestion Management

5.1.1 T3122 Exponential Back-Off

5.1.2 Physical Measurements

5.2 Uplink Power and Timing Control

5.2.1 Uplink Power Control

5.2.2 Uplink Timing Control

Please refer to Appendix section A.1 for in-depth description of the OpenBTS implementation of GSM air interface, “Um”, including references to relevant GSM specifications.

5.1 Downlink Power and Congestion Management

OpenBTS can automatically adjust its downlink power to limit loads and prevent congestion. This feature is especially useful for graceful power-up in areas with very high subscriber density and load-shedding in the event of sudden failure of a neighboring cell or even the failure of a nearby cell of a different operator. This congestion management feature works in conjunction with the T3122 adaptation loop described in Section 5.1.1. The practical result of the automatic power adjustment is to limit the service area of the BTS to a population of nearby phones that it can actually serve.

The configuration parameters associated with this mechanism are:

- GSM.Radio.PowerManager.TargetT3122 – This is the acceptable value of T3122 that the power management loop attempts to achieve. If the actual value of T3122 is larger than this, the BTS will reduce its output power. If the actual value of T3122 is small than this, the BTS will increase its output power (if it is not already maximized). It is critical that this target value be within the bounds set by GSM.Timer.T3122Max and GSM.Timer.T3122Min, as described in Section 5.1.1.
5.1. DOWNLINK POWER AND CONGESTION MANAGEMENT

- **GSM.Radio.PowerManager.Period** – This is the adaptation time constant in milliseconds.

- **GSM.Radio.PowerManager.MaxAttenDB** – The maximum allowed attenuation, in dB relative to full power, which determines the minimum output level. This is also the initial attenuation level.

- **GSM.Power.PowerManager.MinAttenDB** – The minimum allowed attenuation, in dB relative to full scale, which determines the maximum output level. This value is normally zero, allowing the BTS to operate at the maximum power level supported by the hardware.

To disable the automatic power control feature, set the minimum and maximum attenuation levels (MaxAttenDB and MinAttenDB) to the same value, usually zero for maximum power at all times. The CLI “power” command, described in Section C.19, can be used to monitor this mechanism or to control upper and lower power bounds as a pair.

5.1.1 T3122 Exponential Back-Off

When too many MSs make simultaneous access attempts to the BTS, resulting in channel exhaustion, the BTS can respond on the CCCH with an Immediate Assignment Reject message, as defined in GSM 04.08 Section 9.1.20. This message carries a value, T3122, that dictates how long the rejected MS must wait before making another access attempt. (Emergency call attempts are not subject to T3122 waiting.)

OpenBTS implements an exponential back-off algorithm that causes T3122 to grow exponentially whenever channel exhaustion occurs. The bounds for T3122 are set with the configuration parameters GSM.Timer.T3122Max and GSM.Timer.T3122Min, given in milliseconds. To disable the exponential back-off, set these two bounds to the same value.

T3122 back-off is connected to downlink power adaptation, described in Section 5.1.

5.1.2 Physical Measurements

**Downlink Path Loss** in dB can be estimated as \(P_r - P_t\), where \(P_r\) is the RSSI in dBm and \(P_t\) is the output power of the BTS in dBm. For example, if the BTS output is 10 Watts (40 dBm) and the RSSI is reported as -90 dBm, then the total path loss is \(-90 - 40 = -130\) dB. This total loss includes all cable losses and antenna gains.

**Handset Distance.** Round-trip propagation delay is directly proportional to handset’s distance from the BTS. That distance is approximately 535 meters per symbol period of round-trip delay. The round-trip delay reported in the Channel table is in two parts:

- **Timing Advance** – This is a clock offset inside the handset controlled by the BTS to compensate for propagation delay. It is in integer symbol periods, allowing the BTS to adjust timing to \(\pm \frac{1}{2}\) symbol.

- **Timing Error** – This is a timing error measured by the BTS on the arriving signal. Because timing advance is controlled to just \(\pm \frac{1}{2}\) symbol, errors in this range are normal. Positive values mean that the signal is arriving later than ideal. Negative values mean that the signal is arriving early.

The total roundtrip delay is timing advance + timing error, in symbol periods. That sum, multiplied by 535 is an estimate of the handset distance in meters.
CHAPTER 5. OPENBTS AND THE TRANSCEIVER

5.2 Uplink Power and Timing Control

5.2.1 Uplink Power Control

GSM uses a closed-loop uplink power control, described in GSM 05.08 Sections 4.1-4.2 and GSM 05.05 Section 4.1.1. The available maximum power levels of GSM MSs are given in Table 5.1. A multi-band MS can (and typically will) have different power classes in each supported band. The lowest available power output in any band is 5 dBm. The power control range is set with the configuration parameters GSM.MS.Power.Max and GSM.MS.Power.Min, both expressed in dBm. These are normally set to 5 and 39, respectively. These are global settings, applied to all MSs equally. For example, the effect of setting GSM.MS.Power.Max to something less than 39 in a GSM900 unit is to remove any range advantage that might be had by MSs power class 2. If an MS receives a power command that falls outside of its available power range, that MS will set its output power to the closest level available, maximum or minimum. So there is no risk in setting these bounds wider than what the MS can actually support. It may be desirable, though, in some installations, to limit MS power to prevent interference to other cell sites in the area.

Table 5.1: Maximum output power levels for GSM MSs. From GSM 05.05 Section 4.1.1.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N/A</td>
<td>1 W (30 dBm)</td>
<td>1 W (30 dBm)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8 W (39 dBm)</td>
<td>0.25 W (24 dBm)</td>
<td>0.25 W (30 dBm)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5 W (33 dBm)</td>
<td>4 W (36 dBm)</td>
<td>2 W (33 dBm)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2 W (33 dBm)</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.8 W (29 dBm)</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

5.2.2 Uplink Timing Control

GSM uses closed-loop timing advance control, described in GSM 05.10 Section 6. The configuration parameter “GSM.MS.TA.Max” sets a limit on MS timing advance and can be used to deliberately limit the range of service. The value is expressed in symbol periods of round-trip delay, at about 550 meters per step. The normal value of this parameter is 63, which is also the maximum allowed value and corresponds to a maximum range of 35 km.
Chapter 6

SIPAuthServe, Subscriber Registry and Asterisk

6.1 Subscriber Registry ........................................................... 36
   6.1.1 Configuring the Subscriber Registry ................................. 37
   6.1.2 Accessing the Subscriber Registry (Commercial Release Only) 37
   6.1.3 SIPAuthServe Authentication Interface ............................. 37
   6.1.4 Configuring Authentication ............................................ 38
   6.1.5 Intercepting Authentication .......................................... 38

6.2 Provisioning New Subscribers ............................................. 38
   6.2.1 Using Pre-existing SIMs ................................................. 39
   6.2.2 Whitelisting ............................................................... 40
   6.2.3 Open Registration ....................................................... 40
   6.2.4 Generating Custom SIMs ............................................... 42
   6.2.5 Roaming SIMs ............................................................ 42
   6.2.6 The Security of SIMs ................................................... 42

6.3 Emergency Calls .............................................................. 43
   6.3.1 What the User Dials .................................................. 43
   6.3.2 Configuring OpenBTS to Support Emergency Calls ............. 44

6.4 Connecting to a VoIP Carrier .............................................. 44

6.5 Hybrid GSM/SIP Transactions ............................................. 45
   6.5.1 Registration ("Location Updating") .................................. 45
   6.5.2 Call Control ............................................................. 45

6.6 Backhaul Capacity Considerations ....................................... 45
   6.6.1 Available Codecs ..................................................... 47
   6.6.2 RTP, IAX, Overhead and Trunking ................................ 47
One of the distinctive features of OpenBTS is the use of a generic VoIP switch in place of a GSM mobile switching center (MSC). In the standard OpenBTS deployment, the default VoIP switch is Asterisk 11.

The key concept in understanding OpenBTS-SIP integration is that each GSM MS in communication with the BTS unit appears to the VoIP network as a SIP endpoint with the username “IMSIxxxxxxxxxxxxxxx”, where xxxxxxxxxxxxxxx is a 14- or 15-digit IMSI from the MS’s SIM. The IP address of a SIP user is the IP address of its serving BTS. OpenBTS itself is invisible to the VoIP network. It is simply a conduit for the MSs.

### 6.1 Subscriber Registry

Commercial configurations of the OpenBTS Application Suite use a so-called “real time” Asterisk configuration, where Asterisk depends on an external sqlite3 database for its SIP registry and parts of its dialplan. In OpenBTS, the registry database is a part of an application called **SIPAuthServe**, referred to as the **Subscriber Registry**.

The Subscriber Registry database is an Asterisk SIP registry, following the standard `sip_buddies` format, with the following fields added:

- `username varchar(80),`  
  -- this is the original SIP username, but will be "IMSI..." in OpenBTS
- `WhiteListFlag timestamp not null default '0',`  
  -- true if MS is white-listed
- `WhiteListCode varchar(8) not null default '0',`  
  -- white-listing access code
- `rand varchar(33) default '',`  
  -- cached authentication token
- `sres varchar(33) default '',`  
  -- cached authentication token
- `ki varchar(33) default '',`  
  -- Ki, the SIM secret key
- `kc varchar(33) default '',`  
  -- Kc, the session encryption key (not yet used)
- `prepaid int(1) default 0 not null,`  
  -- '1' for pre-paid subscribers (not yet fully supported)
- `calllimit int(5) default 1,`  
  `account_balance int(9) default 0 not null,`  
  `RRLPSupported int(1) default 1 not null,`  
  `hardware varchar(20),`  
  `regTime integer default 0 not null,`  
  `a3_a8 varchar(45) default null`

For Asterisk, the Subscriber Registry database forms part of the SIP registry and dialplan. For other applications in the OpenBTS suite, the registry fills the role normally filled by the HLR and VLR in a conventional GSM net-
work. Like an HLR, the registry stores subscriber information and also provides A3 RAND-SRES authentication (Section A.1.3) for subscribers with known Ki.

Because Asterisk and SIPAuthServe access the sqlite3 database through a file interface, they must be running on the same physical server. The path to this database is stored in the “SubscriberRegistry.db” parameter of the SIPAuthServe configuration.

The default location is “/var/lib/asterisk/sqlite3dir/sqlite3.db”. Asterisk expects to find the database file in this location, so this value should not be changed.

6.1.1 Configuring the Subscriber Registry

The Subscriber Registry configuration parameters are stored in the database located at “/etc/OpenBTS/sipauthserve.db”. They can be edited, if needed, using the OpenRANUI web interface, by selecting the “sipauthserve” and “advanced” tabs in the “Configuration” panel (in commercial installations), or by editing the configuration table with sqlite3 (the only option currently applicable to the public release).

6.1.2 Accessing the Subscriber Registry (Commercial Release Only)

In the commercial release the Subscriber Registry database can be accessed using OpenRANUI web interface. Select the “Databases” panel, pick “SIP buddies” from the list of available databases and click on “(re)load” to print the raw data.

OpenRANUI is also used to manage subscribers. See Section 6.2.1 for details.

6.1.3 SIPAuthServe Authentication Interface

SIPAuthServe supports direct SIP authentication on the Subscriber Registry, performing a transaction modeled after IMS authentication and SIP digest authentication (Figure 6.2), but using:

- the RAND value as the nonce,
- an operator-specified A3/A8 instead of MD5 as the hash function.

SIPAuthServe supports two forms of RAND-SRES authentication on each interface:

- **Full Authentication** – This is the standard A3 authentication, used when Ki is known in the registry. OpenBTS includes support for COMP128v1 as A3. The registry invokes the A3 algorithm as an external Unix application with a standard calling interface, making it easy to add support for other A3 algorithms if implementations are available.

- **Cached Authentication** – This type of authentication uses the same protocol steps on Um as the standard authentication, but the behavior of the registry is different and A3 is not actually run. This authentication is weaker than full authentication, but can but used when Ki is not known.
  - On the first encounter with an MS, the registry generates a RAND value, sends it to the MS and saves both the RAND and resulting SRES in the database. The MS receives a successful authentication response for any correctly-formatted SRES.
– On subsequent authentications, the registry sends the RAND value cached in the database and checks to see if the returned SRES matches the cached value.

SIPAuthServe automatically selects the authentication type based on the availability of Ki. If Ki is known, it uses full authentication. Otherwise it uses cached authentication. The type of registration actually used can be determined by looking into the “sip_buddies” database table to see if a Ki is present.

**Note:** The GSM challenge-response (RAND-SRES) procedure is described in Section A.1.3.

### 6.1.4 Configuring Authentication

OpenBTS performs a SIP challenge-response authentication with SIPAuthServe, which updates the “sip_buddies” table. Asterisk picks up the registration IP address as needed through an ODBC interface to the “sip_buddies” table. This is a typical configuration for a public network with the real time Asterisk configuration. The OpenBTS parameter setting this configuration is stored in:

- **SIP.Proxy.Registration** – This parameter is set to the IP address and port of SIPAuthServe.

### 6.1.5 Intercepting Authentication

For some applications, it may be useful for a network operator to take control of authentication decisions to apply rules that are more complex than simply authenticating individual users against the contents of the “sip_buddies” table. For example, a carrier may be implementing complex whitelist and blacklist operations. OpenBTS developers could try to predict these various custom applications and create a whole new family of configuration parameters. A better approach, though, is for operators to insert their own custom software into the authentication process to “short-circuit” the normal authentication mechanism, or to modify the AIPAuthServe application, which is provided to Range customers under a GPL license.

On the SIP interface, the operator’s custom software is a very simple SIP proxy that accepts and processes the SIP REGISTER method. The OpenBTS “SIP.Proxy.Registration” parameter points to this custom SIP proxy and the custom SIP proxy relays SIP requests and responses as needed between OpenBTS and the Subscriber Registry.

### 6.2 Provisioning New Subscribers

“Provisioning” is a process of creating new subscriber accounts. OpenBTS subscribers are provisioned like any other SIP subscribers in an Asterisk system, with the following constraints:

- The SIP username is always “IMSI” followed by the digits of the IMSI.
- To support full authentication, the subscriber requires a Ki value. If no Ki value is provided, OpenBTS cannot use A3 authentication and will use cached authentication instead. (See Section 6.1.3.)
6.2. PROVISIONING NEW SUBSCRIBERS

6.2.1 Using Pre-existing SIMs

OpenBTS systems can use pre-existing SIMs even in the absence of a carrier roaming agreement, although full RAND-SRES authentication cannot be used because Ki is not known.

Manually

The key to manual provisioning is to determine the IMSI of the SIM used in the MS. Some possible ways to do this are:

1. Locate the IMSI in the TMSI table of the serving BTS with the “tmsis” CLI command or using the OpenRANUI web interface (select the “Databases” panel and pick the “TMSI” table from the drop-down list of available databases).

2. Enable the “Control.LUR.FailedRegistration.Message” feature using the corresponding OpenBTS configuration parameter to deliver a “welcome message” to unprovisioned MSs. The message is automatically appended with the IMSI digits. A typical message might be “To activate service, bring this code to our office: “. See Section 4.2 for more information.

Once the IMSI is known, the operator can generate an entry in the Subscriber Registry and assign a phone number to the subscriber in the OpenBTS network. To do so in the commercial release, use the “Subscribers” panel of the OpenRANUI web interface. Click on “add subscriber” button and provide the subscriber’s name, phone number and IMSI.

Because there is no roaming relationship, the number assigned to the SIM in the OpenBTS network is independent of the number assigned in any other cellular network, although it may be convenient for the two numbers to be the same.

**Note:** It is possible to provision a handset with no telephone number at all, in which case the provisioned MS cannot accept inbound calls but can still place outbound calls.

Interactive via SMS

The SMQueue SMS server and the SIPAuthServe can be used together to provide an interactive autoprovisioning system based on SMS. The configuration parameters are:

- In SMQueue:
  - The “SC.Register.*” parameter set. See Section 7.3.1

- In OpenBTS:
  - “Control.LUR.OpenRegistration” parameter must be defined to accept the intended handsets and the “Control.LUR.OpenRegistration.*” parameters must be defined. See Section 6.2.3.

The autoprovisioning process is:

1. The MS attempts a location updating request. Even though the MS is not provisioned, the network will accept the request.
2. While the MS still has an open dedicated channel to OpenBTS, OpenBTS sends it the open registration welcome message, defined in Control.LUR.OpenRegistration.Message. This message is usually something like, “Please respond to this message with your telephone number to receive service.” The return address for this message, the OpenBTS configuration parameter “Control.LUR.OpenRegistration.ShortCode” must match the address of the ‘register’ short code function, defined in the SMQueue configuration parameter “SC.Register.Code”.

3. The user responds to the text message with a telephone number.

4. The SMS response is transferred from the MS to OpenBTS to SMQueue where it is delivered to the “register” short code function.

5. The “register” short code function updates the Subscriber Registry to provision the new user.

6. The “register” short code function generates an SMS confirmation (or error) message to the user, delivered by SMQueue to OpenBTS to the MS.

6.2.2 Whitelisting

OpenBTS provides a whitelisting system based on SMS. This mechanism can be used to restrict access to the cellular network to a trusted group, even when using automatic provisioning. When a new subscriber entry is created in the Subscriber Registry, it can be designated as on or off the while list. If an MS is on the whitelist, it will register and authenticate normally. If an MS is not on the whitelist, it receives a text message containing an access code, but is otherwise denied service. The new MS can now be added to the whitelist and granted service through this process:

1. Any other MS that is already on the whitelist and has service can send the new MS’s access code to a short code application (Section 7.3) in SMQueue via SMS.

2. This application interacts with the SIPAuthServe (Section 6.1) and adds the new MS to the whitelist.

3. The new subscriber power-cycles the new MS to induce a new location update.

4. The new MS now has service.

6.2.3 Open Registration

Open registration is a mode where all MSs are accepted for registration, regardless of their authentication or provisioning status. Depending on the configuration of Asterisk (Chapter 6) and SMQueue (Chapter 7) these unprovisioned MSs may be able to make telephone calls and send text messages\(^1\). To enable open registration, set “Control.LUR.OpenRegistration” to a POSIX regular expression matching the IMSIs that are to be accepted. See examples below.

There is a second open registration parameter called “Control.LUR.OpenRegistration.Reject” for rejecting IMSIs. If open registration is not used (“Control.LUR.OpenRegistration” is an empty string set with “unconfig Control.LUR.OpenRegistration”), then this parameter is ignored. If open registration

\(^1\) Obviously, such handsets cannot receive calls or text messages because until they are provisioned, they have no assigned telephone numbers
is on, this parameter can be used to reject IMSIs even if they match the regular expression specified in "Control.LUR.OpenRegistration".

Here are some sample regular expressions for "Control.LUR.OpenRegistration" and "Control.LUR.OpenRegistration.Reject" and their open registration effects:

- "001" Match any IMSI starting with "001", the MCC for test networks.
- "00105" Match any IMSI from the test network with MCC=001 and MNC=05
- "001050000000042" Match only IMSI "001050000000042"
- "0" Match any IMSI containing a "0"
- "1" Match any IMSI containing a "1"
- "1024$" Match any IMSI ending in "1024"

The logic for processing registration requests is:

1. Check the IMSI against the Subscriber Registry. If it is not found, proceed to Step 2, otherwise:
   (a) Attempt to authenticate the handset.
   (b) If the handset passes authentication, accept it. Done.
   (c) If the handset fails authentication, send it an authentication failure message, which also implies a rejection. Done.

2. If "Control.LUR.OpenRegistration" is defined:
   (a) If the IMSI does not match "Control.LUR.OpenRegistration", reject the handset with cause "Control.LUR.UnprovisionedRejectCause". Done.
   (b) If the IMSI matches "Control.LUR.OpenRegistration.Reject", reject the handset with cause "Control.LUR.UnprovisionedRejectCause". Done.
   (c) Accept the handset. Done.

Here are some open registration configuration examples:

1. Accept only handsets with test SIMs (MCC=001):
   - Control.LUR.OpenRegistration is "001".
   - Control.LUR.OpenRegistration.Reject is an empty string.

2. Accept only handsets with test SIMs having IMSIs that do not end in "0":
   - Control.LUR.OpenRegistration is "001".
   - Control.LUR.OpenRegistration.Reject is "0$".

3. Accept all handsets except those from the US:
   - Control.LUR.OpenRegistration is ".*".
   - Control.LUR.OpenRegistration.Reject is "~310".

4. Accept all US handsets except those from AT&T:
   - Control.LUR.OpenRegistration is "~310".
   - Control.LUR.OpenRegistration.Reject is "~310410".

---

6.2. PROVISIONING NEW SUBSCRIBERS
6.2.4 Generating Custom SIMs

Range offers writable SIMs (part #8430-xxx series) and SIM writers (part #8425-xxx series) that allow an operator to generate COMP128v1 SIMs with known IMSI’s and known Ki’s, supporting full RAND-SRES authentication.

The OpenBTS application suite includes the software to automate the provisioning process for SIM cards generated using the SIM writer. In the commercial release the RangeSIMd’ application is integrated with the OpenRANUI web interface.

With the SIM writer attached to the BTS and RangeSIMd running, access the web UI, and select the “Subscribers” panel. Click on “write SIM”, verify that the SIM writer status is “ready”, specify the subscriber’s name and phone number, and click on “write”. The record will automatically be added to the Subscriber Registry.

Users of the public release can download and install application called “PySIM”, which programs SIMs from the command-line. Additional information available in the OpenBTS wiki.

6.2.5 Roaming SIMs

For information on roaming support, please contact Range Networks (Section 1.7).

6.2.6 The Security of SIMs

The integrity of the GSM challenge-response authentication is dependent on the secrecy of Ki, which in OpenBTS systems should be known only within the SIM and the Subscriber Registry. In a specification-compliant SIM, Ki cannot be read directly, only tested through the RAND-SRES process. If Ki cannot be extracted from the SIM, then the SIM cannot be fully cloned, since a clone with the wrong Ki will generate the wrong SRES for a given RAND. At least, this is how it is supposed to work.

COMP128v1 and SIM Cloning

The GSM specifications do not define specific algorithms for A3 & A8, but early specifications did offer an example of a combined A3/A8 algorithm called COMP128, since renamed to COMP128v1. As a result of the publication of this example, COMP128v1 was used in most early SIMs, even though that was not the intention of the specification authors.

Since its original publication, COMP128v1 has been compromised. It is well known that it is possible to compute Ki from a COMP128v1-based SIM, given several thousand RAND-SRES pairs. This means that it is possible to build a SIM-cloning system that extracts the IMSI and Ki from a COMP128v1 SIM by running several thousand cycles of COMP128v1 with different RAND inputs and then programming the IMSI and extracted Ki into a blank SIM. The cracking process takes a few hours, but once Ki is known new clones can be produced in a few seconds. Since the development of this cloning technique, most SIM manufacturers have taken at least one of these steps to prevent cloning:

- Use a better algorithm than COMP128v1 for A3/A8, one that is not compromised and not likely to be. The industry standard is currently COMP128v3.
• Design SIMs to shut down or self-destruct if too many A3/A8 calculations are requested too quickly. (With this precaution, even COMP128v1 SIMs are reasonably secure against cloning.)

Avoiding Cloned SIMs

Once a SIMs is cloned, it is impossible to distinguish the clones from the original or from each other. The IMSI in a cloned SIM becomes useless as a subscriber identity, so the proper approach for dealing with cloned SIMs is to detect them through conventional fraud detection techniques, unprovision them and maintain a blacklist of their IMSIs. In a multi-BTS network, there is no practical way for a single BTS unit to detect cloned SIMs, therefore clone detection is not part of OpenBTS itself, but must be performed by the core network. The main sign of a cloned SIM is that the subscriber appears to be moving frequently between geographically disjoint cells or attempts to make multiple simultaneous calls, especially from different cells.

6.3 Emergency Calls

This section applies to the commercial release only.

In GSM, the emergency call is a special transaction, distinct from ordinary mobile-originated call setup. The commercial release of OpenBTS supports the emergency call transaction and presents the call to a SIP router at a configurable extension.

For speed and reliability, OpenBTS always uses very early assignment (VEA) for emergency call establishment in Um, regardless of the setting of the “Control.VEA” configuration parameter in OpenBTS.

6.3.1 What the User Dials

There are several standard “emergency numbers” used in different parts of the world: 911, 112, 999, etc. When a mobile phone user enters one of these dialing codes into an MS, it is not treated as a dialed telephone number. Instead, it is a special code that puts the MS into a special emergency call mode. Most GSM MSs will recognize the dial strings 911, 999 or 112 as an emergency call, regardless of where the MS was sold or where it is being used. When the emergency call is delivered to the BTS, the actual number dialed by the user is not reported by the MS, only the fact that an emergency call has been requested. The routing of an emergency call is configured into the network and has nothing to do with the number actually dialed by the user, as long as the user dials a recognizable emergency number.

When OpenBTS receives an emergency call setup request, it presents the inbound call to its designated emergency call proxy, switch or PBX, which may be different from the proxy, switch or PBX used for normal speech calls. The SIP message headers for the INVITE message are formatted according to 3GPP 24.229 Section 4 and include an encoding of the full cell identity and, optionally, the geocoordinates of the BTS site.

Most call setup operations in OpenBTS are non-queuing and calls are rejected immediately if no channel is available. Emergency calls are an exception to this behavior and may be queued for a few seconds waiting for resources to come available. Furthermore, if an emergency call is placed in a congested cell, OpenBTS will terminate the longest-running non-emergency call in the cell to free a channel for the emergency call.
6.3.2 Configuring OpenBTS to Support Emergency Calls

To support emergency calls in OpenBTS, the following must be configured:

- **“GSM.RACH.AC”** – Bit 10 of GSM.RACH.AC must be cleared to indicate that the network supports emergency calls.
- **“SIP.Proxy.Emergency”** – This parameter specifies the IP address and port of the proxy, switch or PBX used for emergency calls.
- **“Control.Emergency.*”** – All of the parameters in this group must be defined, with the exception of Control.Emergency.Geolocation, which is optional.

See Section 4.2 for details about these parameters. All of these parameters are dynamic and can be set or changed at any time without disrupting service.

6.4 Connecting to a VoIP Carrier

In many VoIP installations, the operator will use a commercial VoIP carrier to route calls to and from the PSTN. (In some cases the cellular operator may also own and operate the VoIP carrier.) In this example, we create a SIP user corresponding to the VoIP carrier and a dialplan context called “from-trunk” where inbound calls from that VoIP carrier are evaluated and routed to an MS.

First, the SIP user representing the VoIP carrier:

```
[my-US-voip-carrier]
context=from-trunk
type=friend
host=my-US-voip-carrier.com
username=myVoIPCarrierAccountUsername
secret=myVoIPCarrierAccountPassword
canreinvite=no
nat=no
insecure=port,invite
qualify=5000
dtmfmode=auto
disallow=all
allow=ulaw
```

Most of these parameters are provided by the carrier. The one to note is the “context” parameter, which we are defining as “from-trunk”. The meaning of this is that inbound calls from the VoIP carrier will be evaluated for routing in the from-trunk context of the dialplan.

Here is the dialplan entry from extensions.conf:

```
[from-trunk]
; route incoming calls from the PSTN
```
6.5 Hybrid GSM/SIP Transactions

6.5.1 Registration (“Location Updating”)

When an MS enters a new “location area” in a GSM network, it performs a “location update request” (LUR). The network can also instruct the MS to perform the LUR periodically on a timer. The LUR operation is the GSM analog to a SIP REGISTER, and OpenBTS maps the LUR to a SIP REGISTER as shown in Figure 6.1. Figure 6.2 shows a more advanced example, including challenge-response authentication. Asterisk uses the simple form with OpenBTS because of differences between GSM and SIP authentication. SIPAuthServe uses the challenge-response form because it is specifically designed to work with GSM authentication.

![GSM location update mapped to a SIP REGISTER (non-authenticating case).](image)

6.5.2 Call Control

Figures 6.3 and 6.4 show the mobile-originated and mobile-terminated call setup cases, using very early assignment for simplicity. In both cases, once the channel is established, the transaction ladder is essentially that of a SIP-ISDN gateway.

6.6 Backhaul Capacity Considerations

In remote areas with poor network connectivity, backhaul bandwidth is often limited or expensive. This section describes the effect of Asterisk configuration on backhaul datarate.
Figure 6.2: GSM location update mapped to a SIP REGISTER (with authentication).

Figure 6.3: A GSM-SIP mobile-terminated call, VEA, normal case.
6.6. BACKHAUL CAPACITY CONSIDERATIONS

6.6.1 Available Codecs

Currently, OpenBTS only supports the GSM full rate codec (GSM-FR), a 13 kbit/sec codec with good speech quality. This codec is also supported by Asterisk and has a moderate level of support among commercial VoIP carriers. The other codecs supported by Asterisk and of potential interest to OpenBTS operators are:

- **G.711 (a-law or mu-law)** – This 64 kbit/sec codec offers very good speech quality and is the most widely used in the PSTN and supported by nearly all VoIP carriers.

- **G.729** – This is an 8 kbit/sec codec with fair speech quality and reasonably well-supported by VoIP carriers. The main drawbacks of G.729 are high computational complexity and a licensing fee of about $10/line/year.

- **Speex** – This codec can operate at rates as low as 4 kbit/sec and has speech quality and computational complexity similar to G.729 at 8 kbit/sec. The advantages of Speex are greater configuration flexibility and freedom from licensing fees. Support for Speex is growing among VoIP carriers.

- **LPC-10** – This is a low-complexity 2.4 kbit/sec codec. The speech is understandable, but often unnatural sounding.

Asterisk will automatically transcode as needed between GSM-FR and any of these codecs.

6.6.2 RTP, IAX, Overhead and Trunking

The media protocol most commonly used with SIP is RTP (IETF RFC-3550). When any codec is run over RTP, there is an additional overhead of roughly 17 kbit/sec per call when used with a 20 ms codec frame. (Yes, for
most codecs, the overhead of RTP is greater than the bandwidth requirement of the codec itself.

Asterisk also supports a combined signaling-and-media protocol called IAX (Inter-Asterisk eXchange). IAX has an overhead of roughly 20 kbit/sec with a codec frame size of 20 ms, but unlike RTP, IAX can distribute this overhead over many calls through a technique called “IAX trunking”. Trunking can be applied to allow any set of calls between a pair of switches to share a common IAX channel and amortize their overhead.

Table 6.1: Backhaul bandwidth for various codec/trunking configurations. All rates in kbit/sec and assuming 20 ms framing.

<table>
<thead>
<tr>
<th>Codec</th>
<th>per call raw rate</th>
<th>per call over RTP</th>
<th>7 calls over RTP</th>
<th>7 calls IAX trunking</th>
<th>speech quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.711</td>
<td>64</td>
<td>81</td>
<td>567</td>
<td>468</td>
<td>toll-quality</td>
</tr>
<tr>
<td>GSM-FR</td>
<td>13</td>
<td>30</td>
<td>210</td>
<td>124</td>
<td>toll-quality</td>
</tr>
<tr>
<td>G.729</td>
<td>8</td>
<td>25</td>
<td>175</td>
<td>97</td>
<td>near-toll-quality</td>
</tr>
<tr>
<td>Speex</td>
<td>8</td>
<td>25</td>
<td>175</td>
<td>97</td>
<td>near-toll-quality</td>
</tr>
<tr>
<td>Speex</td>
<td>4</td>
<td>21</td>
<td>147</td>
<td>60</td>
<td>not toll-quality</td>
</tr>
<tr>
<td>LPC-10</td>
<td>2.4</td>
<td>20</td>
<td>136</td>
<td>37</td>
<td>not toll-quality</td>
</tr>
</tbody>
</table>

Table 6.1 shows the required bandwidth for common combinations of codecs and protocols. Codecs with toll-quality speech are suitable for use in public networks under any conditions. Codecs with near-toll-quality speech are suitable for public networks in remote areas where the only other option would be no service at all. Codecs that are not toll-quality are not suitable for use in public networks, but may be useful in private networks in remote areas where backhaul costs are high. The importance of IAX trunking in reducing bandwidth is obvious, regardless of codec type.

**Satellite Backhaul**

Satellite-based sites are exquisitely sensitive to bandwidth requirements. The specific recommendation for satellite-based OpenBTS sites is for all of the BTS units at the site to route all non-local calls through a pair of Asterisk servers as shown in Figure 6.5. The recommended codec types would be GSM-FR, Speex or LPC-10 depending on budgets and user expectations; the important part is to use IAX trunking.
Figure 6.5: Paired Asterisk servers for IAX trunking in satellite-based applications.
Chapter 7

SMQueue

The delivery of each text message depends on a store-and-forward facility in the network. SMQueue provides this facility. The SMQueue source code is publicly available under AGPLv3.

See Appendix section A.2 for details on SMS implementation in OpenBTS and SMQueue, and references to the relevant GSM standards.

7.1  Design and Operation of SMQueue

The core of SMQueue is a queue of messages awaiting delivery. Messages wait in this queue, potentially through multiple delivery attempts, until delivery is confirmed or until the message is determined to be undeliverable. The operation is similar to that of an email server.

7.1.1  Addressing in SMQueue

SMQueue recognizes two kinds of addresses: ISDN/E.164 numeric addresses and SIP usernames. Any all-numeric address is assumed to be an ISDN/E.164 address and SMQueue will attempt to resolve it to a SIP username using the subscriber registry. Any address that is not all-numeric is assumed to be a SIP username in the operator’s network.
7.2 Configuring SMQueue

In the commercial release SMQueue comes preconfigured. The configuration is stored in an sqlite3 database table located at “/etc/OpenBTS/smqueue.db”. It uses the same schema as the OpenBTS configuration table. OpenRANUI web interface is used to configure SMQueue in commercial installations. Select the “smqueue” tab in the “Configuration” panel. SMQueue parameters are grouped into “basic” and “advanced”.

Currently, the configuration can be changed only by restarting SMQueue after changing the database. Some configuration parameters of note are:

- **SIP.myIP** – The IP address of the machine running SMQueue as seen by the Subscriber Registry server.
- **SIP.myIP2** – The IP address of the machine running SMQueue as seen by remote gateways.
- **SIP.GlobalRelay.*** – The IP address, port and other parameters of a remote RFC-3428 server for delivery of non-local messages. If no such gateway is available, these parameters should be left empty. This set of parameters can be used to build a hierarchy of SMQueue servers for large networks.
- **BounceMessage.*** – A set of error messages sent back to MSs when submitted messages are undeliverable.
- **SC.*** – Configuration parameters for specific short code functions, not for SMQueue itself. See Section 7.3.
- **Log.Level, Log.Level.*** – These are logging controls that behave the same way as those in OpenBTS.config. See Appendix D for more information.

7.3 Short Code Applications

Short codes are local addresses within SMQueue that terminate in local application code. A message sent to a short code becomes an input argument to a short code handler function, instead of being delivered to another user. Short code functions provide a means of writing interactive applications based on text messaging. A typical SMS-based application would normally comprise several short code addresses and handlers sharing common data.

7.3.1 Existing Short Code Applications

There are a few interesting short code applications built into the standard release of SMQueue, although they are all simple applications each requiring only a single handler function.

To disable any short code function, set its short code address to an empty string in the configuration table using “unconfig SC.<FunctionName>.Code” command in the OpenBTS CLI.

Not all short codes are documented here. Undocumented short codes are left disabled in the default configuration. Refer to the “smqueue/smcommands.cpp” source code for information on undocumented short codes.
Autoprovisioning ("Register")

The autoprovisioning application allows OpenBTS users to create new entries in the subscriber registry via SMS. The user sends his desired telephone number in a text message to the autoprovisioning short code address. If the requested number has an acceptable number of digits and is not already assigned to a user, the autoprovisioning handler function will perform the steps described in Section 6.2.1 to provision the new user into the subscriber registry. The autoprovisioning short code handler function is called "shortcode_register" and is configured through the "SC.Register.*" parameters in the SMQueue configuration table. The short code for this function is stored in the "SC.Register.Code" parameter.

For autoprovisioning to work, OpenBTS must also be configured with open registration enabled so that unprovisioned MSs will show service and be capable of sending text messages. The open registration welcome message can be a powerful way to advertise this feature, especially if the return short code of the welcome message is the same as the short code of the autoprovisioning function. See Section 6.2.3 for more information.

Info

This short code handler generates a brief report of SMQueue status, returned in a text message. The implementing function is called "shortcode_four_one_one". This short code handler can be configured with the "SC.Info.*" parameters.

Debug Dump

This short code is tied to an application which dumps debug information to the log. It is defined in the "SC.DebugDump.Code" parameter and is intended for administrative/development use.

Quick Check

This short code is tied to an application which tells the sender how many messages are currently queued. It is defined in the "SC.QuickChk.Code" parameter and is intended for administrative/development use.

7.3.2 Short Code Implementation

The short code implementation in OpenBTS is primitive but functional. Each short code handler is a C++ function coded directly into "smqueue/smcommands.cpp". The arguments to a short code handler are the source IMSI of the message, the message text and a short_code_params data structure into which any reply message can be written. The return value from a short code handler is a status code called short_code_action. See "smqueue/smcommands.cpp" for examples.

Once a short code handler function is defined, it must also be registered at a numeric address. This happens in SMqueue::init_smcommands(), also defined in "smqueue/smcommands.cpp".
Chapter 8

Other GSM Services

8.1 Short Message Service Cell Broadcast (SMSCB) ........................................... 53
8.2 Radio Resource Location Protocol (RRLP) ...................................................... 53

This chapter covers the configuration of 2G GSM services beyond speech and SMS text messaging.

8.1 Short Message Service Cell Broadcast (SMSCB)

The SMSCB data service is discussed in full detail in the Appendix section A.3, here we provide the configuration and OpenBTS implementation details.

The SMSCB service is disabled by default. To enable it, set the "Control.SMSCB.Table" configuration parameter to point to a database file that contains the messages for delivery, e.g. "/var/run/OpenBTS/SMSCB.db". To disable again, execute "unconfig Control.SMSCB.Table".

As of the current version, OpenBTS does not include an application to create entries or otherwise modify the message table. Please refer to Appendix section A.3.2 for the data table schema and additional developer information.

8.2 Radio Resource Location Protocol (RRLP)

RRLP is the protocol used between the network and MS to manage location services (LCS).

As of OpenBTS version 4.0 the implementation of RRLP is in the development stage, and is only mentioned in this manual for future reference.
Chapter 9

General Packet Radio Service (GPRS)

9.1 Configuring GPRS in OpenBTS

9.1.1 Assigning a Range of IP Addresses to GPRS

9.1.2 Configuration Parameters for BSS Functions

9.1.3 Configuration Parameters for SGSN Functions

9.1.4 Configuration Parameters for GGSN Functions

9.2 Configuration of Handsets for OpenBTS GPRS

9.2.1 APN Settings

9.2.2 Data Roaming

9.3 Using CLI to Monitor and Control GPRS

GPRS (General Packet Radio Service) is a 2.5G packet data service that supports speeds of up to 30 KByte/sec on GSM radio channels.

OpenBTS implementation of GPRS differs significantly from conventional networks. See Appendix section A.4 for details on GPRS implementation in OpenBTS, and references to relevant standards.

9.1 Configuring GPRS in OpenBTS

A number of parameters need to be configured/verified before the GPRS service would become available on your BTS. The primary parameter which enables or disables the GPRS function is “GPRS.Enable”. It can be edited using the OpenBTS CLI or OpenRANUI web interface, and requires a restart of OpenBTS.

9.1.1 Assigning a Range of IP Addresses to GPRS

OpenBTS uses the built-in NAT (Network Address Translation) function of the Linux kernel to assign IP addresses to GPRS handsets. The range of available addresses is defined by configuration parameters "GGSN.MS.IP.Base"
and "GGSN.MS.IP.MaxCount". Additionally, the "GGSN.MS.IP.Route" and "GGSN.DNS" parameters can be used to specify the route address and the DNS.

Their default values are as follows:

- **GGSN.MS.IP.Base = "192.168.99.1"**
- **GGSN.MS.IP.MaxCount = 254** – In the default configuration IP addresses will be assigned from the range of 192.168.99.1–192.168.99.255. They will expire after the amount of seconds specified in the "GGSN.IP.ReuseTimeout" developer configuration parameter.
- **GGSN.MS.IP.Route = (disabled)** – When not specified explicitly, OpenBTS manufactures this value from the "GGSN.MS.IP.Base" assuming a 24 bit mask.
- **GGSN.DNS = (disabled)** – When not specified explicitly, DNS servers of the host system are used.

NAT configuration is stored in "/etc/OpenBTS/iptables.rules" file and loaded in "/etc/network/interfaces" at system boot. The default NAT rules work with both static and dynamic IP addresses of the host system, assuming that "eth0" interface is used for internet connectivity. Should you be using a different interface, you need to modify the NAT rules. See Section A.4.2 for details.

See Section A.4.2 for discussion on security implications of using Linux NAT on a BTS unit and various ways of configuring it.

### 9.1.2 Configuration Parameters for BSS Functions

Parameters associated with the BSS functions (layers 1 & 2) of GPRS are prefixed with "GPRS." in the configuration table.

The configuration parameters of particular interest to an operator are:

- **GPRS.Enable** – Enable GPRS service: 0 or 1. If enabled, GPRS service is advertised in the C0T0 beacon, and GPRS service may be started on demand. See also "GPRS.Channels.*".
- **GPRS.ChannelCodingControl.RSSI** – If the signal strength is less than this amount in dB, GPRS uses a lower speed codec CS-1 with less bandwidth but more robust encoding. Note that dB numbers are negative.
- **GPRS.Channels.Min.C0** – Number of timeslots allocated for GPRS service from the C0 (first) ARFCN. See Section A.4.3.
- **GPRS.Channels.Min.CN** – Total number of timeslots allocated for GPRS service from all other ARFCNs except C0. See Section A.4.3.

The following parameters are defined as developer-level parameters (editable via "devconfig" command in the CLI only), but are still worth a note here:

- **GPRS.RAC** – GPRS Routing Area Code, advertised in the C0T0 beacon. The combination of LAC (GSM.Identity.LAC) and RAC must be unique within the network. If all cells have distinct LACs, then this parameter does not matter.
• GPRS.RA.COLOUR – GPRS Routing Area Color as advertised in the C0T0 beacon. This parameter should be distinct from that advertised by any neighboring cells.

• GPRS.MS.Power.* parameters – See Section A.4.3 for details.

• GPRS.Codecs.Downlink – List of allowed GPRS downlink codecs, expressed as a comma-separated list of digits from 1 to 4 for CS-1..CS-4. For example, "1,4" means GPRS may use codecs CS-1 and CS-4. The operator should normally not need to change this.

• GPRS.Codecs.Uplink – List of allowed GPRS uplink codecs, expressed as a comma-separated list of digits from 1 to 4, similar to GPRS.Codecs.Downlink. The operator should normally not need to change this.

9.1.3 Configuration Parameters for SGSN Functions

To get information on the current SGSN status, including IP addresses assigned to specific handsets, use the CLI command "sgsn list" command.

Parameters associated with the SGSN functions of GPRS are prefixed with "SGSN." in the configuration table. These configuration parameters are unlikely to require changes, and are only available using the "devconfig" command in the CLI.

9.1.4 Configuration Parameters for GGSN Functions

Parameters associated with the GGSN functions of GPRS are prefixed with "GGSN." in the configuration table. Some of them have already been described above. The configuration parameters of interest to an operator are:

• GGSN.Firewall.Enable – 0=no firewall; 1=block MS attempted access to OpenBTS or other MS; 2=block all private IP addresses. See Section A.4.2.

• GGSN.IP.ReuseTimeout – How long IP addresses are reserved after a session ends.

• GGSN.IP.TossDuplicatePackets – Any non-zero integer causes duplicate TCP/IP packets to be tossed away to prevent unnecessary traffic on the radio.

• GGSN.Logfile.Name – If specified, internet traffic is logged to this file.

• GGSN.MS.IP.Base – Base IP address assigned to MS.

• GGSN.MS.IP.MaxCount – Number of IP addresses to use for MS.

• GGSN.MS.IP.Route – The route address in the form xxx.xxx.xxx.xxx/yy, must encompass all MS IP addresses. If not specified, OpenBTS manufactures this value from the GGSN.MS.IP.Base assuming a 24 bit mask.

In addition, there is a parameter “GGSN.ShellScript” that enables a shell script that would run each time a new MS device activates GPRS services or creates an IP connection. See Section A.4.3 for details.
9.2 Configuration of Handsets for OpenBTS GPRS

9.2.1 APN Settings

OpenBTS GPRS does not enforce the use of any particular APN (Access Point Name), but at least one APN must be defined in the handset for it to use GPRS.

9.2.2 Data Roaming

If the MCC and MNC of your BTS unit do not match those of a user’s SIM card, it may be necessary to enable data roaming on the handset before it will attempt to use GPRS.

9.3 Using CLI to Monitor and Control GPRS

Two CLI commands used to monitor and control GPRS activity are “gprs” and “sgsn”.

The “gprs” command provides means to selectively start and stop GPRS service, list channels in use by GPRS and set the debug level. The full list of parameters is available in Appendix section C.10.

The “sgsn” command allows to monitor and control SGSN/GGSN sub-system. An overview of its parameters is provided in Appendix section C.25.
Chapter 10

NodeManager

### 10.1 NodeManager API Interface

All commands and parameters in the JSON code are strings with the exception of response codes and the "static" field.

#### 10.1.1 Request Keys

The request to NodeManager includes the following keys:

- **action**  "create", "delete", "read", "update" – this parameter is only used with the "config" command
- **command** "config", "monitor" or "version"
- **key**    a configuration key to which the action is being applied
- **node**   the IP address of a node to which the command applies
- **target** "openbts", "sipauthserve", "smqueue", "rangesimd", "ping" or "rangesysctl"
- **value**  the value to be applied to the configuration key defined in the "key" parameter

NodeManager is a management API allowing third-party applications to manage the following components of OpenBTS Application Suite: OpenBTS, SIPAuthServe and SMQueue. It uses a JSON interface operated over a ZeroMQ REQ-REP socket pair.

10.2 Running NodeManager Queries

10.3 Port Utilization
E.g.: This code requests the version of OpenBTS:

```json
{
    "target": "openbts",
    "command": "version"
}
```

### 10.1.2 Response Codes

The response codes are based on HTTP:

- 200: action ok with response data
- 204: action ok with no response data
- 404: unknown key or action
- 406: request is invalid
- 409: conflicting value
- 500: storing new value failed
- 501: unknown action

### 10.1.3 Response Keys

The NodeManager response includes the following keys:

- **code**: Response code as specified above
- **data**: Response data (optional, depending on the type of request)
- **dirty**: This field is non-zero when a parameter is "static" and could not be applied live, meaning the component needs to be restarted to apply the new value.

E.g.: The "version" command presented above will respond with something like this:

```json
{
    "code": 200,
    "data": "release 4.0.0.8025+GPRS C built Mar 19 2014 rev8026 CommonLibs:rev8026"
}
```

The following is an example of setting an SMQueue parameter:

```json
{
    "target": "smqueue",
    "command": "config",
    "action": "update",
    "key": "SIP.myIP2",
    "value": "192.168.0.22"
}
```

The response is:
Here we read the value of the "GSM.Radio.Band" parameter in OpenBTS:

```
{
  "target": "openbts",
  "command": "config",
  "action": "read",
  "key": "GSM.Radio.Band"
}
```

The response is:

```
{
  "code": 200,
  "data": {
    "defaultValue": "900",
    "description": "The GSM operating band. Valid values are 850 for GSM850, 900 for PGSM900, 1800 for DCS1800 and 1900 for PCS1900. For non-multiband units, this value is dictated by the hardware and should not be changed.",
    "key": "GSM.Radio.Band",
    "scope": 2,
    "static": true,
    "type": "multiple choice",
    "units": "",
    "validValues": "850|GSM850,900|PGSM900,1800|DCS1800,1900|PCS1900",
    "value": "900",
    "visibility": "customer warn - a warning will be presented and confirmation required before changing this sensitive setting"
  }
}
```

**Note:** More examples available in file "JSON_Interface.txt" in the NodeManager sources.

### 10.2 Running NodeManager Queries

OpenRanUI web interface, which is available in the commercial version, provides raw access to the NodeManager.

Access the web UI at "http://<your_bts_ip_address>/ranui/", using username "openbts" and its corresponding password as credentials. Press on ESC in any panel to access the NodeManager modal. Type your command in JSON format in the Input area, and click on "post" to have it executed. The result will appear in the Output area. Pressing ESC again hides the modal.

---

1The source files are available in our GitHub repository at [https://github.com/RangeNetworks/NodeManager](https://github.com/RangeNetworks/NodeManager).
10.3  Port Utilization

Every major component of the OpenBTS application suite has its NodeManager thread. The thread is a ZeroMQ request+response socket which waits for messages on a specific port. Each component uses a different port so they are individually addressable:

<table>
<thead>
<tr>
<th>application</th>
<th>port number</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenBTS</td>
<td>45060</td>
</tr>
<tr>
<td>SMQueue</td>
<td>45063</td>
</tr>
<tr>
<td>SIPAuthServe</td>
<td>45064</td>
</tr>
<tr>
<td>RangeSIMd</td>
<td>45067</td>
</tr>
<tr>
<td>NodeManager</td>
<td>45068</td>
</tr>
</tbody>
</table>
Chapter 11

PowerScanner

The PowerScanner application scans all ARFCNs on the band specified in the OpenBTS configuration parameter “GSM.Radio.Band”, and outputs its findings in the sqlite3 data table located at “/var/run/OpenBTS/PowerScannerResults.db”.

In order to execute the command, you need to disable the transmitter on your system. This can be achieved by stopping the OpenBTS application:

    openbts@ubuntu:~$ sudo stop openbts

Next go to the “/OpenBTS/” directory and run the PowerScanner command as a super-user:

    openbts@ubuntu:~$ cd /OpenBTS
    openbts@ubuntu:/OpenBTS$ sudo ./PowerScanner

Once finished, do not forget to restart the OpenBTS:

    openbts@ubuntu:~$ sudo start openbts

The results of the most recent scan are available in the “/var/run/OpenBTS/PowerScannerResults.db” database. They are also accessible via the OpenRAN web UI in the “Power Scanner” table found in the “Databases” panel.
Appendices
Appendix A

OpenBTS Implementation of GSM & 3GPP Specifications and IETF Standards

A.1 The OpenBTS GSM Air Interface

It is not really necessary to fully understand this chapter to use OpenBTS, but the information is given here for completeness and to provide references to important parts of the GSM specifications to support more detailed study.

A.1 The OpenBTS GSM Air Interface

This section describes the GSM air interface, “Um”, as implemented by OpenBTS. Broadly speaking, Um is organized into channels and layers, as shown in Figure A.1. The rest of this section will explain this diagram.
A.1. Um Layers

The layers of GSM are initially defined in GSM 04.01 Section 7 and roughly follow the OSI model. Um is defined in the lower three layers of the model.

Physical Layer (L1)

The Um physical layer is defined in the GSM 05.xx series of specifications, with the introduction and overview in GSM 05.01. For most channels, Um L1 transmits and receives 184-bit control frames or 260-bit vocoder frames over the radio interface in 148-bit bursts with one burst per timeslot. There are three sublayers:

- Radiomodem. This is the actual radio transceiver, defined in largely in GSM 05.04 and 05.05.
- Multiplexing and Timing. GSM uses TDMA to subdivide each radio channel into as many as 16 traffic channels or as many as 64 control channels. The multiplexing patterns are defined in GSM 05.02.
- FEC Coding. This sublayer provides bit-error concealment and recovery. This sublayer is defined in GSM 05.03.

Radiomodem

OpenBTS supports GMSK modulation with a 13/48 MHz (270.833 kHz) symbol rate and a channel spacing of 200 kHz. Since adjacent channels overlap, the standard does not allow adjacent channels to be used in the same cell. OpenBTS supports the four most common GSM bands:

- GSM850, used in parts of ITU region 2
• PGSM900 and EGSM900, used in most of the world
• DCS1800, used in most of the world
• PCS1900, used in parts of ITU region 2

GSM is frequency duplexed, meaning that the network and MS transmit on different frequencies, allowing the BTS to transmit and receive at the same time. Transmission from the network to the MS is called “downlink”. Transmission from the MS to the network is called “uplink”. GSM uplink and downlink bands are separated by 45 or 50 MHz, depending on the specific band.

Uplink/downlink channel pairs are identified by an index called the ARFCN. Within the BTS, these ARFCNs are given arbitrary carrier indexes C0, C1, etc., with C0 designated as a Beacon Channel and always operated at constant power. The radio channel is time-multiplexed into 8 timeslots, each with a duration of 156.25 symbol periods. These 8 timeslots form a frame of 1,250 symbol periods. The capacity associated with a single timeslot on a single ARFCN is called a physical channel (PhCH) and referred to as “CnTm” where n is a carrier index and m is a timeslot index (0-7).

Each timeslot is occupied by a radio burst with a guard interval, two payload fields, tail bits, and a midamble (or training sequence). The lengths of these fields vary with the burst type but the total burst length is always 156.25 symbol periods. The most commonly used burst is the Normal Burst (NB). There are several other burst formats, though. Bursts that require higher processing gain for signal acquisition have longer midambles. The random access burst (RACH) has an extended guard period to allow it to be transmitted with incomplete timing acquisition. Burst formats are described in GSM 05.02 Section 5.2.

**Multiplexing and Timing**  Each physical channel is time-multiplexed into multiple logical channels according to the rules of GSM 05.02. Traffic channel multiplexing follows a 26-frame (0.12 second) cycle called a ”multiframe”. Control channels follow a 51-frame multiframe cycle. The C0T0 physical channel carries the SCH, which encodes the timing state of the BTS to facilitate synchronization to the TDMA pattern.

GSM timing is driven by the serving BTS through the SCH and FCCH. All clocks in the MS, including the symbol clock and local oscillator, are slaved to signals received from the BTS, as described in GSM 05.10. BTSs in the GSM network can be asynchronous, so that each BTS can run an independent clock.

**FEC Coding**  The coding sublayer provides forward error correction. As a general rule, each GSM channel uses a block parity code (usually a Fire code), a rate-1/2, 4th-order convolutional code and a 4-burst or 8-burst interleaver. Notable exceptions are the synchronization channel (SCH) and random access channel (RACH) that use single-burst transmissions and thus have no interleavers. For speech channels, vocoder bits are sorted into importance classes with different degrees of encoding protection applied to each class (GSM 05.03). Using soft-input Viterbi decoding, the FEC decoders in OpenBTS can recover frames reliably with bit erasure rates in excess of 25%.

Most channels in GSM use 456-bit L1 frames. On channels with 4-burst interleaving (BCCH, CCCH, SDCCH, SACCH), these 456 bits are interleaved in to 4 radio bursts with 114 payload bits per burst. On channels with 8-burst interleaving (TCH, FACCH), these 456 bits are interleaved over 8 radio bursts so that each radio burst carries 57 bits from the current L1 frame and 57 bits from the previous L1 frame. Interleaving algorithms for the most common traffic and control channels are described in GSM 05.03 Sections 3.1.3, 3.2.3 and 4.1.4.
A.1. THE OPENBTS GSM AIR INTERFACE

Data Link Layer (L2)

The Um data link layer, LAPDm, is defined in GSM 04.05 and 04.06. LAPDm is the mobile analog to ISDN’s LAPD and like LAPD, LAPDm is a simplified form of HDLC.

Network Layer (L3)

Um L3 is defined in GSM 04.07 and 04.08 and has three sublayers. A subscriber terminal must establish a connection in each sublayer before accessing the next higher sublayer.

- Radio Resource (RR). This sublayer manages the assignment and release of logical channels on the radio link. It is normally terminated in the BSC, although in OpenBTS, RR is terminated locally in the OpenBTS stack.

- Mobility Management (MM). This sublayer authenticates users and tracks their movements from cell to cell. OpenBTS translates MM transactions into corresponding SIP transactions and uses the Asterisk SIP registry to perform MM functions.

- Call Control (CC). This sublayer connects telephone calls and is taken directly from ITU-T Q.931. GSM 04.08 Annex E provides a table of corresponding paragraphs in GSM 04.08 and ITU-T Q.931 along with a summary of differences between the two. In OpenBTS, CC transactions are translated to corresponding SIP transactions and processed in an external SIP switch or PBX, like Asterisk.

The access order is RR, MM, CC. The release order is the reverse of that.

A.1.2 Um logical channels

Um logical channel types are outlined in GSM 04.03. Broadly speaking, non-GRPS Um logical channels fall into three categories: traffic channels, dedicated control channels and non-dedicated control channels.

Traffic channels (TCH)

These point-to-point channels correspond to the ISDN B channel and are referred to as Bm channels. Traffic channels use 8-burst diagonal interleaving with a new block starting on every fourth burst and any given burst containing bits from two different traffic frames. This interleaving pattern makes the TCH robust against single-burst fades since the loss of a single burst destroys only 1/8 of the frame’s channel bits (a 12.5% bit erasure). The coding of a traffic channel is dependent on the traffic or vocoder type employed, with most coders capable of overcoming single-burst losses. All traffic channels use a 26-multiframe TDMA structure.

Full-rate channels (TCH/F)  A GSM full rate channel uses 24 frames out of a 26-multiframe. The channel bit rate of a full-rate GSM channel is 22.7 kbit/s, although the actual payload data rate is 9.6-14 kbit/s, depending on the channel coding. OpenBTS supports only the GSM full-rate codec (GSM 06.10) as a media type on this channel.
Dedicated Control Channels (DCCHs)

These point-to-point channels correspond to the ISDN D channel and are referred to as Dm channels.

Standalone Dedicated Control Channel (SDCCH)  The SDCCH is used for most short transactions, including initial call setup step, registration and SMS transfer. It has a payload data rate of 0.8 kbit/s. Up to eight SDCCHs can be time-multiplexed onto a single physical channel. The SDCCH uses 4-burst block interleaving in a 51-multiframe. One SDCCH channel can be used to process 10-15 location updates per minute or to transfer 5-10 SMS per minute.

Fast Associated Control Channel (FACCH)  The FACCH is always paired with a traffic channel. The FACCH is a blank-and-burst channel that operates by stealing bursts from its associated traffic channel. Bursts that carry FACCH data are distinguished from traffic bursts by stealing bits at each end of the midamble. The FACCH is used for in-call signaling, including call disconnect, handover and the later stages of call setup. It has a payload data rate of 9.2 kbit/s when paired with a full-rate channel (FACCH/F) and 4.6 kbit/s when paired with a half-rate channel (FACCH/H). The FACCH uses the same interleaving and multiframe structure as its host TCH.

Slow Associated Control Channel (SACCH)  Every SDCCH or FACCH also has an associated SACCH. Its normal function is to carry system information messages 5 and 6 on the downlink, carry receiver measurement reports on the uplink and to perform closed-loop power and timing control. Closed loop timing and power control are performed with a physical header at the start of each L1 frame. This 16-bit physical header carries actual power and timing advance settings in the uplink and ordered power and timing values in the downlink. The SACCH can also be used for in-call delivery of SMS. The SACCH has a payload data rate of 0.2-0.4 kbit/s, depending on the channel with which it is associated. The SACCH uses 4-burst block interleaving and the same multiframe type as its host TCH or SDCCH.

Non-Dedicated Control Channels (NDCCHs)  These are unicast and broadcast channels that do not have analogs in ISDN. These channels are used almost exclusively for radio resource management. The CCCH and RACH together form the medium access mechanism for Um.

Broadcast Control Channel (BCCH)  The BCCH carries a repeating pattern of system information messages that describe the identity, configuration and available features of the BTS. The C0T0 beacon channel must carry an instance of the BCCH.

Synchronization Channel (SCH)  The SCH transmits a Base station identity code and the current value of the TDMA clock. The C0T0 beacon channel must carry an instance of the SCH.

Frequency Correction Channel (FCCH)  The FCCH generates a tone on the radio channel that is used by the MS to discipline its local oscillator.
**A.1. THE OPENBTS GSM AIR INTERFACE**

**Common Control Channel (CCCH)** The CCCH is a downlink unicast channel that carries paging requests and channel assignment messages (specifically, immediate assignment messages). The CCCH is subdivided into the paging channel (PCH) and access grant channel (AGCH). An MS that is camped to a BTS monitors the PCH for service notifications from the network.

**Random Access Channel (RACH)** The RACH is the uplink counterpart to the CCCH. The RACH is a shared channel on which the MSs transmit random access bursts to request channel assignments from the BTS, assignments which are granted on the AGCH part of the CCCH.

**Allowed channel combinations**

The multiplexing rules of GSM 05.02 allow only certain combinations of logical channels to share a physical channel. The combinations currently supported by OpenBTS are:

- **Combination I**: TCH/F + FACCH/F + SACCH. This combination is used for full rate traffic. It can be used anywhere but C0T0.
- **Combination V**: FCCH + SCH + BCCH + CCCH + 4 SDCCH + 4 SACCH. This is the typical C0T0 beacon channel combination for small cells. It can be used only on C0T0. Since this is the only beacon channel combination currently supported by OpenBTS, it must be used on C0T0.
- **Combination VII**: 8 SDCCH + 8 SACCH. This combination is used to provide additional SDCCH capacity in situations where registration loads or SMS usage may be particularly heavy. It can be used anywhere but C0T0.

**A.1.3 Security on Um**

GSM 02.09 defines the following security features on Um:

- authentication of subscribers by the network,
- encryption on the channel,
- anonymization of transactions (at least partially)

Of these, OpenBTS currently supports anonymization and authentication, but not encryption. Authentication relies on a secret key, Ki, that is unique to the subscriber. Copies of Ki are held in the SIM and in the Authentication Center (AuC), a component of the HLR. Ki is never transmitted across Um and should not be visible outside of the AuC.

**Authentication**

The standard GSM authentication procedure is as follows:

1. The MS starts a transaction and presents its IMSI or TMSI to the BSS.
2. If the MS provides a TMSI, BSS resolves the TMSI to an IMSI either through a registry or through an L3 MM Identity Request message.

3. The BSS sends the IMSI to the HLR/AuC.

4. The AuC generates a 128-bit random value, RAND, and sends it to the BSS which then sends it to the MS in the MM Authentication Request message.

5. The MS forms a 32-bit hash value called SRES by encrypting RAND with an algorithm called A3, using Ki as a key. SRES = A3(RAND,Ki). The HLR/AuC performs an identical SRES calculation.

6. The MS sends back its SRES value in the RR Authentication Response message.

7. The BSS relays the SRES, IMSI and RAND back to the HLR/AuC.

8. The HLR/AuC compares its calculated SRES value to the value returned by the MS. If they match, the MS is authenticated.

9. Both the MS and the HLR/AuC also compute a 64-bit ciphering key, Kc, from RAND and Ki using the A8 algorithm. Kc = A8(RAND,Ki).

10. The HLR/AuC reports Kc (or a failure message) to the BSS.

11. The BSS saves Kc for ciphering and reports success or failure to the MS.

The OpenBTS version of this transaction is the same except that OpenBTS replaces the BSS and the subscriber registry replaces the HLR/AuC. The GSM specifications define the characteristics of A3 and A8, but not the algorithms themselves. The specific A3 and A8 algorithms are selected by the SIM manufacturers and usually given to the carriers under NDA. Authentication and Kc generation are closely connected, so Kc is generated even if ciphering will not be used.

**Anonymization**

The TMSI is a 32-bit temporary mobile subscriber identity that can be used to minimized the sending of the IMSI in the clear on Um. The TMSI is assigned by the network with the MM TMSI Reallocation Command during the location updating procedure. Once the TMSI is established, it can be used to anonymize future transactions.

**A.1.4 Service Capacity of Um**

The capacities of OpenBTS products, ARFCN-for-ARFCN, are the same as for any other GSM base stations. The only exception to this is that OpenBTS does not yet support half-rate channels.

OpenBTS offers two types of dedicated channels:

- Full-Rate Traffic Channel (TCH/F). Each Combination-I slot contains a single TCH/F that can carry a single speech call.
• Standalone Dedicated Control Channel (SDCCH). Each Combination-VII slot carries eight SDCCHs. The Combination-V beacon also carries four SDCCHs. Each SDCCH can process about 30 authenticated registration transactions per minute or transfer about 12 text messages per minute, assuming good link margins. *Bear in mind that poor link margins will significantly degrade SDCCH capacity by forcing retransmission of L2 frames and requiring long tear-down times for dropped channels.*

A typical configuration for a single-ARFCN BTS in a speech-oriented application is

• a Combination-IV beacon on C0T0 carrying 4 SDCCHs and
• six Combination-I slots on C0T1-C0T7 carrying total of 7 TCH/Fs.

This combination would typically support about 15 authenticated registrations per minute, about 40 text messages per minute and seven concurrent calls. The number of subscribers that can actually be served with that capacity will be covered in the following sections.

A multi-ARFCN speech-oriented BTS with *N* ARFCNs would typically run

• a Combination-IV or Combination-V beacon on C0T0
• *N* Combination-VII slots, carrying a total of 8*N* SDCCHs and
• 7*N* Combination-I slots, carrying a total of 7*N* TCH/Fs.

This configuration would support roughly 60*N* authentication registrations per minute, 90*N* text messages per minute and 7*N* concurrent calls.

In either of these configuration examples, an operator can trade call capacity for more SMS capacity by using more Combination-VII slots and fewer Combination-I.

**Speech Call Capacity**

Telephone network capacity is measured in Erlangs, where the Erlang number is the average number of subscribers trying to use the network at any given moment (the “offered load”) during the busiest part of the day (the “busy hour”). To get the number of subscribers a BTS can support, take its Erlang capacity and divide by the per-subscriber offered load during the busy hour.\(^1\) Offered loads run from 0.01 to 0.05 Erlang per subscriber, depending on the level of economic development, being higher in more developed areas.\(^2\)

The Erlang capacity of the BTS is dependent on its allowed “blocking probability”, which is the probability that a call will be rejected due to congestion during the busy hour. Blocking probabilities of 2%-3% are typical in public cellular networks. Table A.1 shows Erlang capacities for BTS units in a typical configuration of 7 TCH/F per ARFCN at a blocking probability of 3%. Note that Erlangs/ARFCN increases for larger systems due to increased “trunking efficiency”. All other things being equal, a single 2*N*-ARFCN BTS will have more capacity than 2 *N*-ARFCN units.

---

\(^1\)It’s the busy hour. Do you know your offered load?

\(^2\)For example, take the number of minutes per month you talk on the phone in a month and divide by 14400 to get a conservative estimate of your own offered load.
APPENDIX A. OPENBTS IMPLEMENTATION OF GSM & 3GPP SPECIFICATIONS AND IETF STANDARDS

Table A.1: Speech calling and service capacities of GSM systems.

<table>
<thead>
<tr>
<th>BTS ARFCNs</th>
<th>channels (TCH/F)</th>
<th>Erlangs (3% blocking)</th>
<th>subscribers (0.01 E/sub)</th>
<th>subscribers (0.05 E/sub)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>3.25</td>
<td>325</td>
<td>65</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>8.80</td>
<td>880</td>
<td>176</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>14.9</td>
<td>1490</td>
<td>298</td>
</tr>
<tr>
<td>4</td>
<td>28</td>
<td>21.2</td>
<td>2120</td>
<td>424</td>
</tr>
<tr>
<td>5</td>
<td>35</td>
<td>27.7</td>
<td>2720</td>
<td>554</td>
</tr>
</tbody>
</table>

Registration Load for Camped Phones

The system load from keeping idle phones on the cell is due to their registration activity. A single SDCCH can process about 15 authenticated registrations per minute under good conditions, so a typical single-ARFCN configuration, having 4 SDCCHs, can process about 30 registrations per minute and still have half of its SDCCH capacity available for SMS. Registration can be periodic, driven by the GSM.Timer.T3212 parameter (Section 4.2), or it can be the result of mobility as phones move between cells in a multi-BTS network (Chapter E).

As an example, if a network is configured with a T3212 of one hour and can process 30 authenticated registrations per minute, then it can support 1,800 idle phones. But that is just one configuration example. There are many other possibilities: add a Combination-VII slot to process more registrations (but at the expense of call capacity) or extend or disable the periodic registration timer (at the expense of presence information).

This simple analysis also ignores mobility, assuming that subscribers are not moving much between cells, so the real numbers will be lower, but how much lower is very site-specific. For example, if there is a major road passing through the service area, the registration load might be much higher. So the practical limit on the number of idle phones depends on the capacity and configuration of the cell site, the degree of mobility and the time-granularity of the “presence” information the operator wants, but that limit is on the order of 1,000 phones per ARFCN at the low end in most realistic installations and can be made much higher if TCH/F channels are sacrificed for additional SDCCH capacity.

Finally, note that the numbers given for supportable idle MSs are much larger than the subscriber populations given in Table A.1; registration load should be a non-issue for speech-oriented networks.

A.2 Text Messaging (SMS)

GSM text messaging (“short message service” or SMS) is a service akin to e-mail. Users can send and receive 140-byte messages, allowing up to 160 characters using the SMS 7-bit alphabet. Addresses can be ISDN private network, E.164 or e-mail. SMS is a store-and-forward medium and can be held for minutes, hours or even days if the receiving party is not available. Text messaging also uses reliable channels, like the SDCCH, with frame retransmission and acknowledgement in L2, making it tolerant of frame erasure rates in excess of 50%. These properties make SMS a usable medium over much larger coverage areas than speech, in areas where coverage is spotty or weak and where speech quality would be too poor and calls would disconnect too frequently to be useful.
A.2. TEXT MESSAGING (SMS)

A.2.1 Internet Messaging Protocols

For OpenBTS to handle SMS in a manner consistent with its design goals, the GSM SMS protocol must be translated to and from some open protocol from the internet world. There are many such protocols, but few are well-suited to SMS.

The “Session” Problem

Most messaging protocols in the IETF/IP world, like XMPP, are built around the notion of a “connected session”, a virtual circuit similar to the virtual connection of RTP or TCP/IP. This model assumes an “always-on” network connection where the maintenance of the circuit, with occasional keep-alive messages, is cheap and reasonable.

GSM SMS is different. Maintaining the channel is expensive. There is no keep-alive message mechanism. The circuit switched connection is created and destroyed with every transfer in a process that can take hundreds of milliseconds. Each message transfer is an independent transaction. There is no natural notion of a session.

RFC-3428

RFC-3428 is an IETF standard for the transfer of short messages over the internet. Among IETF/IP protocols for messaging, RFC-3428 is special in that it supports “page mode” messaging without any notion of a session; there is no INVITE to start the transaction or BYE to end it. This makes it a natural fit for SMS. RFC-3428 is straightforward. The sending entity sends a SIP MESSAGE method to the intended receiver. The receiver gives one of the standard SIP responses, preferably 200 OK or 202 Queued to indicate a successful transfer, or a 4xx or 5xx response to indicate failure.

A.2.2 Text Messaging in GSM

GSM 04.11 and 03.40 define conventional SMS in five layers:

1. L1 is taken from the Dm channel type used, either SDCCH or SACCH. This layer terminates in the BSC.
2. L2 is normally LAPDm. This layer terminates in the BTS.
3. L3, the connection layer, is defined in GSM 04.11 Section 5. This layer terminates in the MSC.
4. L4, the relay layer, is defined in GSM 04.11 Section 6. This layer terminates in the MSC.
5. L5, the transfer layer, is defined in GSM 03.40. This layer terminates in the SMSC.

As a general rule, every message transferred in L(n) requires both a transfer and an acknowledgment on L(n-1).

In the OpenBTS implementation of SMS, there is no MSC, so L3 terminates in OpenBTS and L4 is a SIP relay to SMQueue, which takes the place of the SMSC.

Layers of SMS in OpenBTS and SMQueue

We will now consider the handling of each layer of SMS by OpenBTS and SMQueue.
SMS in L3  The Um L3 part of SMS uses three messages:

- CP-DATA to transfer an RPDU across Um and into L4.
- CP-ACK to acknowledge the transfer of an RPDU across Um and into L4.
- CP-ERROR to report the failure to transfer an RPDU to L4.

An RPDU is a “relay (layer) protocol data unit”, which is just an encapsulation of a message from L4. The operation in L3 is simple. The entity that needs to transfer an RPDU sends it in a CP-DATA message. The receiving entity responds with CP-ACK or CP-ERROR. Transactions are non-overlapping.

The action of OpenBTS upon receiving CP-DATA from an MS is to verify the correct encoding of the L3 part of the message and respond with CP-ACK or CP-ERROR. OpenBTS then extracts the RPDU, transfers it to SMQueue as an application/vnd.3gpp.sms MIME payload in a SIP MESSAGE method and waits for a response (200 OK or 202 Queued for success or 4xx or timeout for failure).

After sending CP-DATA to an MS, OpenBTS waits for CP-ACK or CP-ERROR, proceeding after CP-ACK or aborting the transaction after CP-ERROR.

SMS in L4  The Um L4 part of SMS uses four messages:

- RP-DATA to transfer a TPDU across Um and into L5.
- RP-ACK to acknowledge the transfer of a TPDU across Um and into L5.
- RP-ERROR to report the failure to transfer an TPDU to L5.
- RP-SMMA for the MS to report that it has more memory available to receive SMS messages (not yet supported by OpenBTS).

An TPDU is a “transfer (layer) protocol data unit”, which is just an encapsulation of a message from L5. OpenBTS translates between SIP and SMS L4 as follows:

- RP-DATA – MESSAGE method with the RPDU as an application/vnd.3gpp.sms MIME payload,
- RP-ACK – 200 OK or 202 Queued response,
- RP-ERROR – any other response or timeout,
- RP-SMMA – (not yet supported by OpenBTS).

SMS in L5  The Um L5 part of SMS uses these message:

- SMS-SUBMIT to transfer a text message from the MS to the network,
- SMS-DELIVER to transfer a text message from the network to the MS.

OpenBTS transfers L5 PDUs (TPDUs) as opaque payloads. SMQueue manipulates L5 headers as needed to convert SMS-SUMBIT TPDUs into SMS-DELIVER TPDUs during the delivery process.
A.2. TEXT MESSAGING (SMS)

RFC-3428/SMS Transaction Ladders

Now we take a look at all of the GSM layers and the SIP transactions together.³

**Mobile Terminated SMS**  Figure A.2 shows a complete mobile terminated SMS transfer where the network, through SMQueue, transfers a text message to the MS. The message arrives at SMQueue from the outside world addressed either to a SIP user or to a numeric address. SMQueue resolves the destination address to an IMSI-based SIP user name and forwards the message to OpenBTS. OpenBTS pages the MS, establishes a channel, transfers the message as SMS and then responds to SMQueue with 200 OK.

The most common failure in the mobile terminated transfer is that the MS does not respond to paging. In this case, SMQueue never receives any response. The message remains in the SMQueue delivery queue and another delivery attempt will be made in a few minutes.

![Diagram of SMS transfer](image)

**Figure A.2:** Mobile-terminated SMS transfer with no parallel call, normal case.

**Mobile Originated SMS**  Figure A.3 shows a complete mobile originated SMS transfer, where the MS transfers a text message to SMQueue for later delivery to its addressee. The message originates in the MS with a numeric address. The MS establishes a radio channel to OpenBTS and then sends the text message TPDU in an RP-DATA message. OpenBTS translates the TPDU to a SIP MESSAGE method and sends that to SMQueue. SMQueue responds with OK and then OpenBTS responds to the MS with RP-ACK.

³Strictly speaking, page-mode message transfers are not transactions in SIP, since they are not contained within an INVITE-BYE session. However, these transfers are transactions inside of OpenBTS and will be referred to as transactions throughout the OpenBTS documentation.
A.3 Short Message Service Cell Broadcast (SMSCB)

Short Message Service Cell Broadcast (SMSCB) is a low-rate data service defined in GSM Specifications 03.41 and 04.21. It was originally intended for low-rate information services such as traffic reports and sports scores. While the name suggests that this broadcast service is somehow closely related to SMS, the truth is that the two are completely independent of each other. Nearly all GSM MSs are capable of receiving and displaying SMSCB messages, although most MSs are not configured to do so by default.

Note: Currently, OpenBTS supports only ASCII for SMSCB payloads.

A.3.1 Cell Broadcast Channel (CBCH)

The Cell Broadcast Channel (CBCH), defined in GSM Specification 05.02 Section 6.5.4, is essentially an SDCCH that has been set aside to the SMSCB payload. If SMSCB is enabled, one SDCCH must be sacrificed to provide the bandwidth. There are constraints on the placement of the CBCH in terms of carrier index and timeslot. To meet these constraints, either the C0T0 must be Combination-V or GSM.Channels.C1sFirst must be left undefined (an empty string).

The CBCH has a maximum data rate of 97.7 bytes per second. Each message is carried in a fixed-length 88-byte frame having 6 header bytes and an 82-byte payload field. The service is capable of delivering roughly one such message every second.

A.3.2 Scheduling Messages for Delivery

Specification 03.41 describes a hierarchy of servers to distribute content for the SMSCB service and a protocol in L3 for delivery of SMSCB content to the BTS.
A.4. GENERAL PACKET RADIO SERVICE (GPRS)

OpenBTS does not follow this model. Instead, OpenBTS takes messages for delivery from an sqlite3 database table at a path specified in the “Control.SMSSB.Table” parameter. The schema is:

```
CREATE TABLE IF NOT EXISTS SMSCB (  
    GS INTEGER NOT NULL, -- See GSM 03.41 9.3.2.1.  
    MESSAGE_CODE INTEGER NOT NULL, -- See GSM 03.41 9.3.2.1.  
    UPDATE_NUMBER INTEGER NOT NULL, -- See GSM 03.41 9.3.2.1.  
    MSGID INTEGER NOT NULL, -- See GSM 03.41 9.3.2.2.  
    LANGUAGE_CODE INTEGER NOT NULL, -- See GSM 03.41 9.3.2.3 and GSM 03.38.  
    MESSAGE TEXT NOT NULL, -- the actual message text, ASCII  
    SEND_TIME INTEGER DEFAULT 0, -- Unix time when this message was last sent  
    SEND_COUNT INTEGER DEFAULT 0 -- number of times this message has been sent
)
```

Inside OpenBTS, the SMSCB sending loop scans this table at the SMSCB message rate, sending the message with the smallest SEND_TIME on each iteration and then updating the SEND_TIME to the current Unix time. To schedule messages for delivery, an external application, provided by the operator, creates new entries in the SMSCB database table with a SEND_TIME value of zero. To stop delivery of a message permanently, delete its record from the table. To suspend delivery of a message, set its SEND_TIME to the future time at which delivery is to resume.

A.4 General Packet Radio Service (GPRS)

GPRS (General Packet Radio Service) is a 2.5G packet data service that supports speeds of up to 30 KByte/sec on GSM radio channels. GPRS is designed to share physical-layer resources with GSM, but above the physical layer GPRS is essentially a completely different protocol stack that sits alongside the GSM circuit switched service. It is useful, therefore, to think of a 2.5G BTS unit as having two distinct and largely independent subsystems, one for GSM circuit-switched services and one for GPRS packet-switched services.

The implementation of GPRS in OpenBTS is subject to the following limitations:

- No support for handover of IP (Internet Protocol) sessions. If a GPRS handset moves to a new cell, its GPRS IP address will change. This might disrupt in-progress IP transfers. Note, however, that most interactive web-browser sessions use many short individual transactions and can also recover from lost packets, so this typical use scenario is less likely to be disrupted than a longer continuous transaction, for example, a file transfer or listening to music.
- Static IP addresses are not supported, so all IP transactions must be initiated by the handset.
- Alternate named IP portals are not supported, which means all IP sessions simply go to the global internet rather than a specified server. The APN (Access Point Name), if any, is ignored.
- Connection to a customer specified SGSN or GGSN is not supported. The SGSN and GGSN are internal to OpenBTS.
- There is no support for dual-transfer mode, which means no simultaneous voice call and IP session on a single handset.
• The supported channel coding options are codecs CS-1 and CS-4 (the slowest and the fastest).

• Total system performance may be limited by the available bandwidth of the CCCH (Common Control Channel), which may restrict the number of simultaneously active MSs. This restriction has been greatly alleviated in release C3.1, and is reduced even further when the TBF persistence feature (configuration parameter "GPRS.Uplink.Persist") is enabled, however if a multi-ARFCN BTS has a large number of channels dedicated to GPRS it may still be possible to encounter this limitation. See Section A.4.3 for more details.

A.4.1 Components of GPRS

A conventional GPRS network uses these components:

• GPRS-capable BTS (Base Transceiver Station), which supports packet transfer on the air interface.

• GPRS-capable BSC (Base Station Controller), which provides GPRS-specific radio resource management.

• SGSN (Serving GPRS Support Node), which manages GPRS sessions between the network and handset.

• GGSN (Gateway GPRS Support Node), which connects GPRS sessions to the internet. (This is where IP addresses get associated with GPRS sessions.)

In the OpenBTS implementation of GPRS, all of these functions are incorporated into the OpenBTS stack. Still, these functional elements are referred to by their conventional names in documentation and configuration parameters to avoid confusion.

A.4.2 Use of NAT and Security Implications

The OpenBTS GGSN function uses the built-in NAT (Network Address Translation) function of the Linux kernel to assign IP addresses to GPRS handsets. Because of the way NAT is implemented in Linux systems, handsets do not have publicly routable IP addresses, not even dynamic ones. A handset’s IP address as known to the outside world will be the same as that of its serving BTS unit.

The OpenBTS use of NAT for GGSN functions has security implications that must be understood by the operator to avoid insecure configurations. It is very important that operators read and understand this section.

Since the GPRS-enabled device is assigned an IP address inside the BTS and allowed virtually unlimited access to IP protocols, a malicious device could potentially gain undesired access to the BTS itself. To prevent this, the GGSN prevents any GPRS-enabled device from talking to any other GPRS-enabled device in the same BTS, or to the BTS itself, and optionally, to any private IP address by implementing the firewall. The firewall is controlled by the "GGSN.Firewall.Enable" configuration option and is enabled by default. If GGSN internal firewall is disabled, a GPRS-enabled device can access the BTS itself, which poses a serious security risk. It is critical to add routing rules on the Linux system running the BTS to prevent IP addresses in the range allocated to the GPRS-enabled devices from accessing the BTS unit itself.

Note: It is also important that GPRS systems use strong passwords on the BTS units and to avoid the creation of extraneous user accounts.
Additionally, depending on the internet topology where the BTS is connected, it is also possible that these GPRS-enabled devices are entering the private IP network to which the BTS unit is connected. For a simple network topology where you have other computers or base stations using private IP addresses (for example, 192.168.xxx.yyy) behind a single firewall/router, this access can be prevented by setting the "GPRS.Firewall.Enable" parameter to "2" which attempts to prevent the GPRS-enabled devices from accessing ANY private IP address.

For more complex network topologies, firewalls on the BTS units themselves must be configured to prevent such access. The current GGSN internal firewall settings are logged at the NOTICE level when GPRS starts.

**NAT Programming**

Whether a BTS is using a dynamic or static IP address, the following commands can be used to configure the NAT. Note that the last command implies the use of "eth0" interface. Should your system use a different one, the parameter needs to be set accordingly:

```
sudo modprobe iptable_nat
sudo iptables -t nat -F
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

To view the current NAT settings the command is:

```
sudo iptables -t nat --list
```

Once you confirm that the NAT settings are correct, they need to be saved so that they would still be in effect after a reboot:

```
sudo iptables-save > ~/iptables.rules
sudo mv ~/iptables.rules /etc/OpenBTS/
```

The settings are then being loaded in "/etc/network/interfaces" at system boot.

**A.4.3 Radio Resource Management and Performance in GPRS**

**GPRS Registration**

This is the process whereby the MS (Mobile Station or cell phone) informs the SGSN component of GPRS of its IMSI identity. Each MS must register before it can use IP services. The GPRS registration process is separate from the GSM registration process, and an MS will usually hold two registrations, one for CS (Circuit Switched, i.e. voice) services, and another for PS (Packet Switched, or Internet Protocol) services. These registrations last for a programmable period, typically about an hour, after which they must be renewed. The GSM registration is periodically renewed using a "Location Area Update" command and the GPRS registration is renewed using a "Routing Area Update" command.

If GPRS is enabled, then almost any MS in range of the BTS will attempt to register whether they intend to use GPRS packet services or not. For this reason there will be continual GPRS activity even if there is no packet flow. You can view a list of registered devices via the OpenBTS CLI command:

```
sgsn list
```
APPENDIX A. OPENBTS IMPLEMENTATION OF GSM & 3GPP SPECIFICATIONS AND IETF STANDARDS

Temporary Block Flow

A GPRS channel assignment is called a TBF (Temporary Block Flow). In a TBF, an MS is granted the use of one or more PhCH (physical channels) to transfer IP data. Each PhCH occupies one timeslot on one ARFCN. Adjacent PhCHs can be ganged to provide "multi-slot" transfer capability.

OpenBTS creates a TBF when IP packets need to be transferred between the network and the MS. The TBF remains open as long as there are IP packets to be transferred. When there are no more IP packets being transmitted and the persistence delay has expired, the TBF is released. Another TBF will be established for the next transfer when the next IP packet arrives.

Performance

Over each PhCH (slot) in a TBF, IP packets are segmented in layer 2 into RLC frames\(^4\) carrying 20-50 bytes of user data each, depending on the encoding codec type used,\(^5\) and transferred at a rate of 50 frames per second per PhCH "slot" assigned to the MS. The codec that OpenBTS uses in turn depends on the instantaneous link margin (signal strength) for each individual MS. This means that the slowest GPRS connection (single slot, codec CS-1) is about 1000 bytes per second, while the fastest downlink using the default multi-slot configuration (2-up, 3-down) is about 50x50x3 = 7500 bytes per second. Actual available payload rates are usually 10%-30% lower due to TBF establishment overhead and TCP/IP handshaking and header overhead.

Performance will be markedly lower (as low as 1/2 of the maximum) for applications that perform many small IP transfers, since the ratio of overhead to payload is larger for smaller transfers.

CCCH Congestion

If the MS and BTS have not communicated recently (around 5 seconds), OpenBTS must send TBF assignment messages to the MS using the same format as GSM immediate assignments, which are carried on the AGCH (Access Grant Channel) and PCH (Paging Channel) subchannels of the CCCH. The CCCH resource is shared with GSM phone call initiation, sending of text messages, periodic location and routing area updating messages, and other services. OpenBTS is currently capable of issuing up to a total of 12 CCCH messages/sec, of which 4/sec can be used to initiate uplink TBFs, 4/sec can be used to initiate downlink TBFs, and 4/sec are reserved for GSM use. TBF initiation using CCCH involves a significant delay that increases with the number of pending CCCH requests, and therefore, with the number of simultaneous users. When the delay in TBF initiation gets too long, the MS device gives up and may retry or simply report a loss of internet service to higher layers. Therefore when CCCH congestion exceeds a threshold (controlled by "GSM.CCCH.AGCH.QMax") there is no longer any point in OpenBTS attempting to send the message, and instead OpenBTS discards the message and logs a message at the CRITICAL or ALERT level. (The "GSM.CCCH.AGCH.QMax" config parameter should not normally be changed by the operator as its value is determined by fixed delays in the GSM specification interacting with the intrinsic bandwidth of the OpenBTS CCCH channel.)

Channel Congestion

To maximize GPRS bandwidth the MS must be granted adjacent PhCH, meaning the operator must set the "GSM.Channels.*" parameters so that adjacent channels would be allocated for GPRS, which simply means they must have values greater than one. If all channels are in use, OpenBTS starts issuing

---

\(^4\)RLC is Radio Link Control, the layer 2 part of GPRS. Similar to LAPDm, it provides reliable segmentation and reassembly across the physical channel.

\(^5\)The slowest encoding type, CS-1 uses a rate-1/2 convolutional encoder. The fastest type, CS-4, uses no convolutional encoding. CS-2 and CS-3 are intermediate rates based on a punctured rate-1/2 code, but these intermediate rates are not currently supported by OpenBTS.
TBF assignments that share the available channels, which may impact the performance of individual MSs. In the current implementation, all MSs share channel bandwidth equally.

**TCP Delays** The underlying internet does not guarantee that data packets will be delivered. Web browser sessions require guaranteed packet delivery so they usually use the TCP protocol, which is a guaranteed-delivery protocol that works by re-transmitting packets that are lost. The TCP/IP algorithm assumes that packet loss is due to congestion on the internet, and so it assumes that any packet that has not been delivered in a few seconds has been lost, and retransmits it after a variable delay that is designed to reduce congestion globally on the internet. Unfortunately, this algorithm interacts badly with slow connections like GPRS. There are multiple TCP algorithms in common use, and TCP also deliberately introduces random numbers into the congestion avoidance algorithm, so the exact delays incurred by TCP are difficult to characterize. But typically when the amount of information to be transferred simultaneously (for example, when loading a single web page) exceeds a threshold that is approximately 25 KB per 10 KB/sec of bandwidth, then the information cannot be transferred before TCP starts retransmitting the packets. This introduces two additional sources of delay: first, the extra back-off delay can result in the channel going idle, which typically happens only for transfers near the threshold; second, bandwidth may be wasted by unnecessary packet retransmission, which for large transfers can grow exponentially. This means that for a typical GPRS connection with a download bandwidth 30 KB/sec, the maximum size web-page that can be viewed is about 75KB before additional extraneous delay is added by the TCP algorithm. That's why there are websites custom tailored for cell phones. It is also why measuring GPRS performance by downloading a 100KB file is inaccurate.

**GPRS Channel Allocation**

The operator must decide how many channels (timeslots) to dedicate for GPRS services and how many to dedicate for CS (Circuit Switched, i.e. voice) services. On startup, the GRPS subsystem will take over the specified number of Combination-I physical channels (timeslots) and dedicate them to GPRS service, converting them to Combination-XIII. These physical channels are taken as contiguous groups on the same ARFCN when possible. The number of channels is specified by configuration parameters “GPRS.Channels.Min.C0” and “GPRS.Channels.Min.CO”. Currently, the number of GPRS channels is fixed at startup and cannot be changed without restarting OpenBTS. The reason the operator can specify the number of GPRS channels for ARFCN C0 separately is to balance the system resources between GSM and GPRS services to reduce the total number of ARFCNs being powered at any given moment. Since there will always be GPRS registration activity when GPRS is enabled, the operator should always allocate some channels on ARFCN C0. For best GPRS performance, the number of channels allocated per-ARFCN should be a multiple of the maximum multi-slot allocation. For example, if “GPRS.Multislot.Max.Downlink” is set at the default value of 3, then the optimal values for “GPRS.Channels.Min.CO” are 3 or 6. Alternatively, for a multi-ARFCN BTS that is used primarily for voice calls and rarely for GPRS, the operator could allocate just 1 GPRS channel on ARFCN C0 to handle GPRS registration activity, and still get multi-slot GPRS capability by allocating more GPRS channels (e.g. 3) on the other ARFCNs.

**Multislot Allocation** GPRS can group timeslots together to provide greater transfer bandwidth, a feature called “multi-sloting”. The “GPRS.Multislot.Max.Uplink” and “GPRS.Multislot.Max.Downlink” parameters control the maximum number of slots that OpenBTS will provide to an MS device in the uplink and downlink directions, respectively. The multi-slot configuration assigned to each MS is also limited by the

---

6 This feature is available in release 3.1 and higher.
number of adjacent GPRS channels (timeslots) available for GPRS service when the TBF is created, and also by
the individual MS capabilities. Most modern MS devices support up to five total slots in some combination of
1-4 slots each on downlink and/or uplink. Some common multislots classes for consumer devices that are also
supported by OpenBTS are given in Table A.2.

The default multi-slot assignment is specified in the config parameters and is 2-slots up, 3-slots down. This is
the fastest configuration for most purposes, especially web-browsing. Another common configuration is 2-slots
up, 2-slots down, which is almost as fast but packs MS into ARFCNs better. While it is possible to change
the default to 1-slot up, 4-slots down, it is not recommended. Counter-intuitively, browsers send a large amount
of IP traffic in the uplink direction, sometimes more than in the downlink direction, so using a 1-up, 4-down is
usually much slower than the default.

Table A.2: Common GPRS multislot classes, giving maximum uplink and downlink multislot capabilities.

<table>
<thead>
<tr>
<th>Multislots Class</th>
<th>Max. Rx</th>
<th>Max. Tx</th>
<th>Max. Sum Rx+Tx</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

**GPRS Uplink Power Control**

GPRS, as implemented by OpenBTS, uses open loop uplink power control, where the handset sets its transmitted
power based on the RSSI of the network’s downlink signal, assuming roughly equal uplink and downlink path
losses. The algorithm used to set the transmitted power is described in GSM 05.08 Section 10.2.1. The formula
is:

\[ P_t = \min(\Gamma_0 - \Gamma_C - \alpha(C + 48), P_m) \]

where:

- \( P_t \) is the transmitted power in dBm.
- \( \Gamma_0 \) is 39 dBm in the low bands and 36 dBm in the high bands.
- \( \Gamma_C \) is a configurable parameter with units of dB, encoded into the parameter GPRS.MS.Power.Gamma in
  2 dB steps. (For example, a configured value of 16 would give \( \Gamma_C \) of 32 dB.)
- \( \alpha \) is a unitless configurable parameter, encoded into the configuration parameter GPRS.MS.Power.Alpha
  in steps of 0.1. (For example, a configured value of 5 would give \( \alpha \) of 0.5.)
A.4. **GENERAL PACKET RADIO SERVICE (GPRS)**

- $C$ is the RSSI of the network downlink signal as measured by the handset.
- $P_m$ is the maximum transmission power of the handset, usually 33 dBm.

For a low-band (850 or 900 MHz) BTS unit with an output of 33 dBm per ARFCN, values of $\alpha = 1.0$ and $\Gamma_C = 58$ dB will cause the handset to hit maximum output power when the downlink RSSI hits -100 dBm. Values of $\alpha = 1.0$ and $\Gamma_C = 60$ dB will have the same effect in the high band. For every dB of power that the BTS output is increased, $\Gamma_C$ should be decreased by a corresponding dB.\(^7\) The corresponding configuration parameters are:

- **GPRS.MS.Power.Alpha** – The $\alpha$ parameter, $\times 10$ to give a step size of 0.1, so $\alpha = 1.0$ is encoded as a value of 10.
- **GPRS.MS.Power.Gamma** – The $\Gamma_C$ parameter, $\div 2$ to give a step size of 2 dB, so $\Gamma_C = 58$ dB is encoded as a value of 29 and $\Gamma_C = 60$ dB is encoded as a value of 30.

**GGSN Shell Script**

This is intended primarily for telemetry applications. OpenBTS can invoke a shell script each time a new MS device activates GPRS services or creates an IP connection. The “GGSN.ShellScript” parameter is set to the name of a UNIX shell script. The arguments passed to the shell script for each action and the significance of the action are as follows:

- “Start”  
  The system has started.
- “GprsAttach <IMSI>”  
  MS specified by <IMSI> has GPRS attached.
- “GprsDetach <IMSI>”  
  MS specified by <IMSI> has GPRS detached, or attach has expired.
- “PdpActivate <IMSI IPaddress NSAPI>”  
  MS specified by <IMSI> has activated an <IPaddress>. Each MS may activate up to 11 IP addresses numbered by <NSAPI> which is numbered from 5 to 15.
- “PdpDeactivate <IMSI IPaddress NSAPI>”  
  MS specified by <IMSI> has deactivated an <IPaddress> numbered by <NSAPI>.
- “PdpDeactivateAll <IMSI>”  
  Deactivate all IP addresses for the specified MS; there may be none.

Notes:

1. The shell script invocation is serialized by OpenBTS, so only one shell script runs at a time.

\(^7\)In future versions of OpenBTS, $\alpha$ and $\Gamma_C$ will adapt automatically based on uplink RSSI measurements.
2. The shell script does not provide any way to prevent an MS from attaching in the first place; that authentication is accomplished using the subscriber registry.

3. The MS devices may activate/deactivate PdpContexts on short timeframes, meaning that they could potentially change IP addresses often. This problem is mitigated by OpenBTS by making the IP addresses issued to the MS devices semi-static, meaning that the same MS will get the same IP address every time until the BTS is power cycled or the IP address pool is exhausted requiring re-issue of previously issued IP addresses.
# Appendix B

## Configuration Parameters

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1</td>
<td>OpenBTS Parameters</td>
</tr>
<tr>
<td>B.1.1</td>
<td>Customer Site Parameters</td>
</tr>
<tr>
<td>B.1.2</td>
<td>Customer Tuneable Parameters</td>
</tr>
<tr>
<td>B.1.3</td>
<td>Developer/Factory Parameters</td>
</tr>
<tr>
<td>B.2</td>
<td>Smqueue Parameters</td>
</tr>
<tr>
<td>B.2.1</td>
<td>Customer Site Parameters</td>
</tr>
<tr>
<td>B.2.2</td>
<td>Customer Tuneable Parameters</td>
</tr>
<tr>
<td>B.2.3</td>
<td>Developer/Factory Parameters</td>
</tr>
<tr>
<td>B.3</td>
<td>SIPAuthServe Parameters</td>
</tr>
<tr>
<td>B.3.1</td>
<td>Customer Site Parameters</td>
</tr>
<tr>
<td>B.3.2</td>
<td>Customer Tuneable Parameters</td>
</tr>
<tr>
<td>B.3.3</td>
<td>Developer/Factory Parameters</td>
</tr>
</tbody>
</table>

## B.1 OpenBTS Parameters

OpenBTS configuration parameters are stored in OpenBTS Configuration Table located at "/etc/OpenBTS/OpenBTS.db". See section 4.1 for details on the Configuration Table schema and access.

### B.1.1 Customer Site Parameters

These parameters must be changed to fit your site.

- `Control.Emergency.GatewaySwitch` – Gateway SIP switch for inbound calls from other networks. This host is used to form the return path for emergency calls, so it should be a host address that will route from your serving PSAP.
• **Control.Emergency.Geolocation** – If defined, send this location as an RFC-4119 XML GEOPRIV object during SIP emergency call establishment. Format is dd:mm:ss[NS] ddd:mm:ss[EW].

• **GSM.Identity.BSIC.BCC** – GSM basestation color code; lower 3 bits of the BSIC. BCC values in a multi-BTS network should be assigned so that BTS units with overlapping coverage do not share a BCC. This value will also select the training sequence used for all slots on this unit.

• **GSM.Identity.BSIC.NCC** – GSM network color code; upper 3 bits of the BSIC. Assigned by your national regulator. Must be distinct from NCCs of other GSM operators in your area.

• **GSM.Identity.CI** – Cell ID, 16 bits. In some cases, the last digit of the cell id represents the sector id. A last digit of 0 is used for an omnidirectional antenna. A last digit of 1, 2, 3, etc indicates a sector of the multi-sector antenna. Should be unique.

• **GSM.Identity.LAC** – Location area code, 16 bits, values 0xFFxx are reserved. For multi-BTS networks, assign a unique LAC to each BTS unit. (This is not the normal procedure in conventional GSM networks, but is the correct procedure in OpenBTS networks.)

• **GSM.Identity.MCC** – Mobile country code; must be three digits. Defined in ITU-T E.212. Value of 001 for test networks.

• **GSM.Identity.MNC** – Mobile network code, two or three digits. Assigned by your national regulator. 01 for test networks.

• **GSM.Identity.ShortName** – Network short name, displayed on some phones. Optional but must be defined if you also want the network to send time-of-day.

• **GSM.Radio.C0** – The C0 ARFCN. Also the base ARFCN for a multi-ARFCN configuration.

### B.1.2 Customer Tuneable Parameters

These parameters can be changed to optimize your site.

• **Control.Emergency.Destination.Host** – SIP destination host to be used for the “To:” header of emergency calls. This host may be different from the address in SIP.Proxy.Emergency.

• **Control.Emergency.Destination.User** – SIP destination user or extension to be used for the “To:” header of emergency calls. IMS specifies “sos”, but correct value must be matched to your switch configuration and PSAP interface.

• **Control.Emergency.QueueTime** – Maximum time to wait for a channel to open up for an emergency call in a congested system, in milliseconds.

• **Control.Emergency.RFC5031** – Use the RFC-5031 URN sip:sos@SIP.Proxy.Emergency as the request URN for outbound emergency calls over SIP, regardless of the value of Emergency.Destination.User. The “To:” header will still be Emergency.Destination.User@Emergency.Destination.Host.

• **Control.Emergency.Source.User** – SIP identity to use if no IMSI is available. IMS specifies “anonymous” but other values might be more useful depending on your configuration.

• **Control.GSMTAP.GPRS** – Capture GPRS signaling and traffic at L1/L2 interface via GSMTAP.
B.1. OPENBTS PARAMETERS

- Control.GSMTAP.GSM – Capture GSM signaling at L1/L2 interface via GSMTAP.
- Control.GSMTAP.TargetIP – Target IP address for GSMTAP packets; the IP address of Wireshark, if you use it for real time traces.
- Control.LUR.404RejectCause – Reject cause for location updating failures for phones that fail authentication. The SIP result code from the Registrar in this case is 404. Reject causes come from GSM 04.08 10.5.3.6. Reject cause 0x02 or 0x04 is usually the right one.
- Control.LUR.AttachDetach – Use attach/detach procedure. This will make initial LUR more prompt. It will also cause an un-registration if the handset powers off and really heavy LUR loads in areas with spotty coverage. Range Networks strongly recommends setting this to 1.
- Control.LUR.FailMode – Action to take after registration failure due to network failure, error in Registrar, or other unexpected error. This does not apply to regular authorization failure handled by other config options. If ACCEPT the handset is authorized for service. If FAIL the handset is denied service. If OPEN the open registration procedure is applied.
- Control.LUR.FailedRegistration.Message – Send this text message, followed by the IMSI, to un-provisioned handsets that are denied registration.
- Control.LUR.FailedRegistration.ShortCode – The return address for the failed registration message. If unset, the message will not be sent.
- Control.LUR.NormalRegistration.Message – The text message, followed by the IMSI, to be sent to provisioned handsets when they attach on Um. By default, no message is sent. To have a message sent, specify one. To stop sending messages again, execute “unconfig Control.LUR.NormalRegistration.Message”.
- Control.LUR.NormalRegistration.ShortCode – The return address for the normal registration message. If unset, the message will not be sent.
- Control.LUR.OpenRegistration – This value is a regular expression. Any handset with an IMSI matching the regular expression is allowed to register, even if it is not provisioned. By default, this feature is disabled. To enable open registration, specify a regular expression to match. E.g. 001, which matches any IMSI starting with 001, the MCC for test networks. To disable open registration again, execute “unconfig Control.LUR.OpenRegistration”.
- Control.LUR.OpenRegistration.Message – Send this text message, followed by the IMSI, to un-provisioned handsets when they attach on Um due to open registration.
- Control.LUR.OpenRegistration.Reject – This value is a regular expression. Any unprovisioned handset with an IMSI matching the regular expression is rejected for registration, even if it matches Control.LUR.OpenRegistration. By default, this filter is disabled. To enable the filter, specify a regular expression. E.g. 666 matches any IMSI starting with 666, which currently does not correspond to any known MCC. Stay on the light side of the Force! To disable the filter again, execute “unconfig Control.LUR.OpenRegistration.Reject”. If Control.LUR.OpenRegistration is disabled, this parameter has no effect.
- Control.LUR.OpenRegistration.ShortCode – The return address for the open registration message.
- Control.LUR.QueryClassmark – Query every MS for classmark during LUR.
- Control.LUR.QueryIMEI – Query every MS for IMEI during initial LUR.
APPENDIX B. CONFIGURATION PARAMETERS

• **Control.LUR.RegistrationMessageFrequency** — This option helps determine when a registration message is sent by the BTS to a handset. If 'PLMN' the message is sent only when the handset first registers in the PLMN, as reported by the handset. If 'NORMAL' the message is sent whenever the handset enters the cell, as reported by the handset. If 'FIRST' the message is sent the first time this BTS sees this MS as determined by the WELCOME_SENT field of the TMSI_TABLE. This option is not completely reliable because the functioning of this option depends on information provided by the handset during their initial attach procedure, and some handsets set this information improperly.

• **Control.LUR.SendTMSIs** — Send new TMSI assignments to handsets that are allowed to attach.

• **Control.LUR.UnprovisionedRejectCause** — Reject cause for location updating failures for unprovisioned phones, that is, the IMSI was not found in the Registrar database. The SIP result code from the Registrar in this case is 401. Reject causes come from GSM 04.08 10.5.3.6. Reject cause 0x02 or 0x04 is usually the right one.

• **Control.Reporting.PhysStatusTable** — File path for channel status reporting database.

• **Control.Reporting.StatsTable** — File path for statistics reporting database.

• **Control.Reporting.TMSITable** — File path for TMSI table database.

• **Control.Reporting.TransactionTable** — File path for transaction table database.

• **Control.SMSCB.Table** — File path for SMSCB scheduling database. By default, this feature is disabled. To enable, specify a file path for the database e.g. /var/run/OpenBTS/SMSCB.db. To disable again, execute “unconfig Control.SMSCB.Table”.

• **Control.TMSITable.MaxAge** — Maximum allowed age in hours for a TMSI entry in the TMSITable. This is not the authorization/registration expiry period, this is how long the BTS remembers assigned TMSIs. Currently old entries are only discarded at startup.

• **Control.VEA** — Use very early assignment for speech call establishment. See GSM 04.08 Section 7.3.2 for a detailed explanation of assignment types. If VEA is selected, GSM.CellSelection.NECI should be set to 1. See GSM 04.08 Sections 9.1.8 and 10.5.2.4 for an explanation of the NECI bit. Note that some handset models exhibit bugs when VEA is used and these bugs may affect performance.

• **GGSN.DNS** — The list of DNS servers to be used by downstream clients. By default, DNS servers of the host system are used. To override, specify a space-separated list of DNS servers, in IP dotted notation, eg: 1.2.3.4 5.6.7.8. To use the host system DNS servers again, execute “unconfig GGSN.DNS”.

• **GGSN.Firewall.Enable** — 0=no firewall; 1=block MS attempted access to OpenBTS or other MS; 2=block all private IP addresses.

• **GGSN.IP.TossDuplicatePackets** — Toss duplicate TCP/IP packets to prevent unnecessary traffic on the radio.

• **GGSN.MS.IP.Base** — Base IP address assigned to MS.

• **GGSN.MS.IP.MaxCount** — Number of IP addresses to use for MS.

• **GGSN.MS.IP.Route** — A route address to be used for downstream clients. By default, OpenBTS manufactures this value from the GGSN.MS.IP.Base assuming a 24 bit mask. To override, specify a route address in the form xxx.xxx.xxx.xxx/yy. The address must encompass all MS IP addresses. To use the auto-generated value again, execute “unconfig GGSN.MS.IP.Route”.
B.1. OPENBTS PARAMETERS

- **GGSN.ShellScript** – A shell script to be invoked when MS devices attach or create IP connections. By default, this feature is disabled. To enable, specify an absolute path to the script you wish to execute e.g. /usr/bin/ms-attach.sh. To disable again, execute “unconfig GGSN.ShellScript”.

- **GPRS.CellOptions.T3168Code** – Timer 3168 in the MS controls the wait time after sending a Packet Resource Request to initiate a TBF before giving up or reattempting a Packet Access Procedure, which may imply sending a new RACH. This code is broadcast to the MS in the C0T0 beacon in the GPRS Cell Options IE. See GSM 04.60 12.24. Range 0..7, representing values from 0.5sec to 4sec in 0.5sec steps.

- **GPRS.CellOptions.T3192Code** – Timer 3192 in the MS specifies the time MS continues to listen on PDCH after all downlink TBFs are finished, and is used to reduce unnecessary RACH traffic. This code is broadcast to the MS in the C0T0 beacon in the GPRS Cell Options IE. The value must be one of the codes described in GSM 04.60 12.24. Value 0 implies 500msec; 2 implies 1500msec; 3 impiles 0msec.

- **GPRS.ChannelCodingControl.RSSI** – If the initial unlink signal strength is less than this amount in dB, GPRS uses a lower bandwidth but more robust encoding CS-1. This value should normally be GSM.Radio.RSSIOrigin + 10 dB.

- **GPRS.Channels.Min.C0** – Minimum number of channels allocated for GPRS service on ARFCN C0.

- **GPRS.Channels.Min.CN** – Minimum number of channels allocated for GPRS service on ARFCNs other than C0.

- **GPRS.Enable** – If enabled, GPRS service is advertised in the C0T0 beacon, and GPRS service may be started on demand. See also GPRS.Channels.*.

- **GPRS.LocalTLLI.Enable** – Enable recognition of local TLLI.

- **GPRS.Multislot.Max.Downlink** – Maximum number of channels used for a single MS in downlink.

- **GPRS.Multislot.Max.Uplink** – Maximum number of channels used for a single MS in uplink.

- **GPRS.NMO** – Network Mode of Operation. See GSM 03.60 Section 6.3.3.1 and 24.008 4.7.1.6. Allowed values are 1, 2, 3 for modes I, II, III. Mode II (2) is recommended. Mode I implies combined routing updating procedures.

- **GPRS.Reassign.Enable** – Enable TBF Reassignment.

- **GPRS.TBF.EST** – Allow MS to request another uplink assignment at end up of uplink TBF. See GSM 4.60 9.2.3.4.

- **GPRS.TBF.Retry** – If 0, no tbf retry, otherwise if a tbf fails it will be retried with this codec, numbered 1..4.

- **GSM.CCCH.AGCH.QMax** – Maximum number of access grants to be queued for transmission on AGCH before declaring congestion.

- **GSM.CellOptions.RADIO-LINK-TIMEOUT** – Seconds before declaring a physical link dead.

- **GSM.CellOptions.RESELECT-HYSTERESIS** – Cell Reselection Hysteresis. See GSM 04.08 10.5.2.4, Table 10.5.23 for encoding. Encoding is $2^N$ dB, values of $N$ are 0..7 for 0..14 dB.

- **GSM.CellSelection.NCCsPermitted** – NCCs Permitted. An 8-bit mask of allowed NCCs. The NCC of your own network is automatically included. Unless you are coordinating with another carrier, this should be left at zero.
• GSM.CellSelection.NECI – NECI, New Establishment Causes. This must be set to 1 if you want to support very early assignment (VEA). It can be set to 1 even if you do not use VEA, so you might as well leave it as 1. See GSM 04.08 10.5.2.4, Table 10.5.23 and 04.08 9.1.8, Table 9.9 and the Control.VEA parameter.

• GSM.Channels.C1sFirst – Allocate C-I slots first, starting at C0T1. Otherwise, allocate C-VII slots first.

• GSM.Channels.NumC1s – Number of Combination-I timeslots to configure. The C-I slot carries a single full-rate TCH, used for speech calling.

• GSM.Channels.NumC7s – Number of Combination-VII timeslots to configure. The C-VII slot carries 8 SDCCHs, useful to handle high registration loads or SMS. If C0T0 is C-IV, you must also have at least one C-VII.

• GSM.Channels.SDCCHReserve – Number of SDCCHs to reserve for non-LUR operations. This can be used to force LUR transactions into a lower priority.

• GSM.Cipher.CCHBER – Probability of a bit getting toggled in a control channel burst for cracking protection.

• GSM.Cipher.Encrypt – Encrypt traffic between MS and OpenBTS.

• GSM.Cipher.RandomNeighbor – Probability of a random neighbor being added to SI5 for cracking protection.

• GSM.Cipher.ScrambleFiller – Scramble filler in layer 2 for cracking protection.

• GSM.Handover.FailureHoldoff – The number of seconds to wait before attempting another handover with a given neighbor BTS.

• GSM.Handover.LocalRSSIMin – Do not handover if downlink RXLEV (reported by the MS) is above this level (in dBm), regardless of power difference.

• GSM.Handover.ThresholdDelta – A neighbor downlink signal must be this much stronger (in dB) than this downlink signal for handover to occur.

• GSM.MS.Power.Damping – Damping value for MS power control loop in percent. The ordered MS power is based on RSSI [Received Signal Strength Indication]. A value of 100 here ignores RSSI entirely; a value of 0 causes the MS power to change instantaneously based on RSSI, which is inadvisable because it sets up power oscillations. The ordered MS power is then clamped between GSM.MS.Power.Max and GSM.MS.Power.Min.

• GSM.MS.Power.Max – Maximum commanded MS power level in dBm.

• GSM.MS.Power.Min – Minimum commanded MS power level in dBm.

• GSM.MS.TA.Damping – Damping value for timing advance control loop.

• GSM.MS.TA.Max – Maximum allowed timing advance in symbol periods. One symbol period of round-trip delay is about 0.55 km of distance. Ignore RACH bursts with delays greater than this. Can be used to limit service range. Valid range is 1..62.
B.1. OPENBTS PARAMETERS

- **GSM.MaxSpeechLatency** – Maximum allowed speech buffering latency, in 20 millisecond frames. If the jitter is larger than this delay, frames will be lost.

- **GSM.Neighbors** – A list of IP addresses of neighbor BTSs available for handover. By default handover is disabled. To enable, specify a space-separated list of a maximum of 31 OpenBTS IP addresses in IP dotted notation, optionally followed by a colon and the port number. E.g.: 1.2.3.4 5.6.7.8:16001. To disable again, execute “unconfig GSM.Neighbors”.

- **GSM.Neighbors.NumToSend** – Maximum number of neighbors to send to handset in the neighbor list broadcast in the beacon.

- **GSM.Ny1** – Maximum number of repeats of the Physical Information Message during handover procedure, GSM 04.08 11.1.3.

- **GSM.RACH.AC** – Access class flags. This is the raw parameter sent on the BCCH. See GSM 04.08 10.5.2.29 for encoding. Set to 0 to allow full access. Set to 0x0400 to indicate no support for emergency calls.

- **GSM.RACH.MaxRetrans** – Maximum RACH retransmission attempts. This is the raw parameter sent on the BCCH. See GSM 04.08 10.5.2.29 for encoding.

- **GSM.RACH.TxInteger** – Parameter to spread RACH bursts over time. This is the raw parameter sent on the BCCH. See GSM 04.08 10.5.2.29 for encoding.

- **GSM.Radio.ARFCNs** – The number of ARFCNs to use. The ARFCN set will be C0, C0+2, C0+4, etc.

- **GSM.Radio.Band** – The GSM operating band. Valid values are 850 for GSM850, 900 for PGSM900, 1800 for DCS1800 and 1900 for PCS1900. For non-multiband units, this value is dictated by the hardware and should not be changed.

- **GSM.Radio.MaxExpectedDelaySpread** – Expected worst-case delay spread in symbol periods, roughly 3.7 us or 1.1 km per unit. This parameter is dependent on the terrain type in the installation area. Typical values are: 1 for open terrain and small coverage areas, a value of 4 is strongly recommended for large coverage areas. This parameter has a large effect on computational requirements of the software radio; values greater than 4 should be avoided.

- **GSM.Radio.PowerManager.MaxAttenDB** – Maximum transmitter attenuation level, in dB wrt full scale on the D/A output. This sets the minimum power output level in the output power control loop.

- **GSM.Radio.PowerManager.MinAttenDB** – Minimum transmitter attenuation level, in dB wrt full scale on the D/A output. This sets the maximum power output level in the output power control loop.

- **GSM.Radio.RSSITarget** – Target uplink RSSI for MS power control loop, in dB wrt to A/D full scale. Should be 6-10 dB above the noise floor.

- **GSM.ShowCountry** – Tell the MS to show the country name based on the MCC.

- **GSM.SpeechBuffer** – Size of speech buffer in milliseconds. If set to 0, no RTP speech buffer is used. If set to 1, the RTP speech buffer size is determined adaptively. Any other value sets the speech buffer size. The speech buffer is needed to overcome jitter caused by natural variation in the internet traffic delay. Note that speech is noticeably delayed by this amount, so we want to keep it as low as possible and still have reasonably reliable delivery. The specified delay is in addition to the intrinsic buffering inside OpenBTS. This value is used only at the start of a call; changing it does not affect on-going calls.
APPENDIX B. CONFIGURATION PARAMETERS

- **GSM.Timer.Handover.Holdoff** – Handover will not be permitted until this time has elapsed after an initial channel seizure or handover.

- **GSM.Timer.T3212** – Registration timer T3212 period in minutes. Should be a factor of 6. Set to 0 to disable periodic registration. Should be smaller than SIP registration period.

- **Log.Alarms.Max** – Maximum number of alarms to remember inside the application.

- **Log.Level** – Default logging level when no other level is defined for a file.

- **Peering.Neighbor.RefreshAge** – Seconds before refreshing parameters from a neighbor.

- **Peering.NeighborTable.Path** – File path for neighbor information database.

- **Peering.Port** – The UDP port used by the peer interface for handover.

- **RTP.Range** – Range of RTP port pool. Pool is RTP.Start to RTP.Range - 1.

- **RTP.Start** – Base of RTP port pool. Pool is RTP.Start to RTP.Range - 1.

- **SIP.DTMF.RFC2833** – Use RFC-2833 (RTP event signalling) for in-call DTMF.

- **SIP.DTMF.RFC2833.PayloadType** – Payload type to use for RFC-2833 telephone event packets.

- **SIP.DTMF.RFC2976** – Use RFC-2976 (SIP INFO method) for in-call DTMF.

- **SIP.Local.IP** – IP address of the OpenBTS machine as seen by its proxies. If these are all local, this can be localhost.

- **SIP.Local.Port** – IP port that OpenBTS uses for its SIP interface.

- **SIP.Proxy.Emergency** – The hostname or IP address and port of the proxy to be used for emergency calls.

- **SIP.Proxy.Registration** – The hostname or IP address and port of the proxy to be used for registration and authentication. This should normally be the subscriber registry SIP interface, not Asterisk.

- **SIP.Proxy.SMS** – The hostname or IP address and port of the proxy to be used for text messaging. This is smqueue, for example.

- **SIP.Proxy.Speech** – The hostname or IP address and port of the proxy to be used for normal speech calls. This is Asterisk, for example.

- **SIP.Proxy.USSD** – The hostname or IP address and port of the proxy to be used for USSD, or “testmode” to test by reflecting USSD messages back to the handset. To disable USSD, execute “unconfig SIP.Proxy.USSD”.

- **SIP.RFC3428.NoTrying** – Send “100 Trying” response to SIP MESSAGE, even though that violates RFC-3428.

- **SIP.SMSC** – The SMSC handler in smqueue. This is the entity that handles full 3GPP MIME-encapsulated TPDUs. If not defined, use direct numeric addressing. The value should be disabled with “unconfig SIP.SMSC” if SMS.MIMETYPE is “text/plain” or set to “smsc” if SMS.MIMETYPE is “application/vnd.3gpp”.

- **SMS.FakeSrcSMSC** – Use this to fill in L4 SMSC address in SMS delivery.
B.1. OPENBTS PARAMETERS

- **SMS.MIMEType** – This is the MIME Type that OpenBTS will use for RFC-3428 SIP MESSAGE payloads. Valid values are “application/vnd.3gpp.sms” and “text/plain”.

- **TRX.Args** – Extra arguments for the Transceiver.

- **TRX.IP** – IP address of the transceiver application.

B.1.3 Developer/Factory Parameters

These parameters should only be changed by when developing new code.

- **CLI.SocketPath** – Path for Unix domain datagram socket used for the OpenBTS console interface.

- **Control.Call.QueryRRLP.Early** – Query every MS for its location via RRLP during the setup of a call.

- **Control.Call.QueryRRLP.Late** – Query every MS for its location via RRLP during the teardown of a call.

- **Control.Emergency.RRLP** – Query every MS for its location via RRLP during an Emergency Call.

- **Control.LUR.QueryRRLP** – Query every MS for its location via RRLP during LUR.

- **Control.LUR.TestMode** – Used for testing the LUR procedure.

- **Control.NumSQLTries** – Number of times to retry SQL queries before declaring a database access failure.

- **Control.Reporting.TransactionMaxCompletedRecords** – Maximum completed records to be stored for gathering by an external stats tool.

- **Control.SACCHTimeout.BumpDown** – Decrease the RSSI by this amount to induce more power in the MS each time we fail to receive a response from it on SACCH.

- **Control.SMS.QueryRRLP** – Query every MS for its location via RRLP during an SMS.

- **Control.WatchdogMinutes** – Number of minutes before the radio watchdog expires and OpenBTS is restarted, set to 0 to disable.

- **GGSN.IP.MaxPacketSize** – Maximum size of an IP packet. Should normally be 1520.

- **GGSN.IP.ReuseTimeout** – How long IP addresses are reserved after a session ends.

- **GGSN.Logfile.Name** – If specified, internet traffic is logged to this file. E.g. ggsn.log.

- **GGSN.TunName** – Tunnel device name for GGSN.

- **GPRS.Channels.Congestion.Threshold** – The GPRS channel is considered congested if the desired bandwidth exceeds available bandwidth by this amount, specified in percent.

- **GPRS.Channels.Congestion.Timer** – How long in seconds GPRS congestion exceeds the Congestion.Threshold before we attempt to allocate another channel for GPRS.

- **GPRS.Codecs.Downlink** – An empty value specifies GPRS may use all available codecs. Otherwise list of allowed GPRS downlink codecs 1..4 for CS-1..CS-4. Currently, only 1 and 4 are supported e.g. 1,4.
• GPRS.Codecs.Uplink – An empty value specifies GPRS may use all available codecs. Otherwise list of allowed GPRS uplink codecs 1..4 for CS-1..CS-4. Currently, only 1 and 4 are supported e.g. 1,4.

• GPRS.Counters.Assign – Maximum number of assign messages sent.

• GPRS.Counters.N3101 – Counts unused USF responses to detect nonresponsive MS. Should be \( \leq 8 \). See GSM04.60 Sec 13.

• GPRS.Counters.N3103 – Counts ACK/NACK attempts to detect nonresponsive MS. See GSM04.60 sec 13.

• GPRS.Counters.N3105 – Counts unused RRBP responses to detect nonresponsive MS. See GSM04.60 Sec 13.

• GPRS.Counters.Reassign – Maximum number of reassign messages sent.

• GPRS.Counters.TbfRelease – Maximum number of TBF release messages sent.

• GPRS.Debug – Toggle GPRS debugging.

• GPRS.Downlink.KeepAlive – How often to send keep-alive messages for persistent TBFs in milliseconds; must be long enough to avoid simultaneous in-flight duplicates, and short enough that MS gets one every 5 seconds. GSM 5.08 10.2.2 indicates MS must get a block every 360ms

• GPRS.Downlink.Persist – After completion, downlink TBFs are held open for this time in milliseconds. If non-zero, must be greater than GPRS.Downlink.KeepAlive.

• GPRS.MS.KeepExpiredCount – How many expired MS structs to retain; they can be viewed with gprs list ms -x

• GPRS.MS.Power.Alpha – MS power control parameter, unitless, in steps of 0.1, so a parameter of 5 is an alpha value of 0.5. Determines sensitivity of handset to variations in downlink RXLEV. Valid range is 0...10 for alpha values of 0...1.0. See GSM 05.08 10.2.1.

• GPRS.MS.Power.Gamma – MS power control parameter, in 2 dB steps. Determines baseline of handset uplink power relative to downlink RXLEV. The optimum value will tend to be lower for BTS units with higher power output. This default assumes a balanced link with a BTS output of 2-4 W/ARFCN. Valid range is 0...31 for gamma values of 0...62 db. See GSM 05.08 10.2.1.

• GPRS.MS.Power.T_AVG_T – MS power control parameter; see GSM 05.08 10.2.1.

• GPRS.MS.Power.T_AVG_W – MS power control parameter; see GSM 05.08 10.2.1.

• GPRS.NC.NetworkControlOrder – Controls measurement reports and cell reselection mode (MS autonomous or under network control); should not be changed. See GSM 5.08 10.1.4.

• GPRS.PRIORITY-ACCESS-THR – Code contols GPRS packet access priorities allowed. See GSM04.08 table 10.5.76.

• GPRS.RAC – GPRS Routing Area Code, advertised in the C0T0 beacon.

• GPRS.RA_COLOUR – GPRS Routing Area Color as advertised in the C0T0 beacon.

• GPRS.RRBP.Min – Minimum value for Relative Reserved Block Period (RRBP) reservations, range 0..3. Should normally be 0. A non-zero value gives the MS more time to respond to the RRBP request.
B.1. OPENBTS PARAMETERS

- **GPRS.SendIdleFrames** – Should be 0 for current transceiver or 1 for deprecated version of transceiver.
- **GPRS.TBF.Downlink.Poll1** – When the first poll is sent for a downlink tbf, measured in blocks sent.
- **GPRS.TBF.Expire** – How long in milliseconds to try before giving up on a TBF.
- **GPRS.TBF.KeepExpiredCount** – How many expired TBF structs to retain; they can be viewed with gprs list tbf -x.
- **GPRS.Timers.Channels.Idle** – How long in milliseconds a GPRS channel is idle before being returned to the pool of channels. Also depends on Channels.Min. Currently the channel cannot be returned to the pool while there is any GPRS activity on any channel.
- **GPRS.Timers.MS.Idle** – How long in seconds an MS is idle before the BTS forgets about it.
- **GPRS.Timers.MS.NonResponsive** – How long in milliseconds a TBF is non-responsive before the BTS kills it.
- **GPRS.Timers.T3169** – Nonresponsive uplink TBF resource release timer, in milliseconds. See GSM04.60 Sec 13.
- **GPRS.Timers.T3191** – Nonresponsive downlink TBF resource release timer, in milliseconds. See GSM04.60 Sec 13.
- **GPRS.Timers.T3193** – Timer T3193 (in milliseconds) in the base station corresponds to T3192 in the MS, which is set by GPRS.CellOptions.T3192Code. The T3193 value should be slightly longer than that specified by the T3192Code. If 0, the BTS will fill in a default value based on T3192Code.
- **GPRS.Timers.T3195** – Nonresponsive downlink TBF resource release timer, in milliseconds. See GSM04.60 Sec 13.
- **GPRS.Uplink.KeepAlive** – How often to send keep-alive messages for persistent TBFs in milliseconds; must be long enough to avoid simultaneous in-flight duplicates, and short enough that MS gets one every 5 seconds.
- **GPRS.Uplink.Persist** – After completion uplink TBFs are held open for this time in milliseconds. If non-zero, must be greater than GPRS.Uplink.KeepAlive. This is broadcast in the beacon and cannot be changed once BTS is started.
- **GPRS.advanceblocks** – Number of advance blocks to use in the CCCH reservation.
- **GSM.CCCH.CCCH-CONF** – CCCH configuration type. DO NOT CHANGE THIS. Value is fixed by the implementation. See GSM 10.5.2.11 for encoding. Value of 1 means we are using a C-V beacon. Any other value selects a C-IV beacon.
- **GSM.CellSelection.MS-TXPWR-MAX-CCH** – Cell selection parameters. See GSM 04.08 10.5.2.4.
- **GSM.CellSelection.RXLEV-ACCESS-MIN** – Cell selection parameters. See GSM 04.08 10.5.2.4.
- **GSM.Control.GPRSMaxIgnore** – Ignore GPRS messages on GSM control channels. Value is number of consecutive messages to ignore.
- **GSM.Neighbors.Averaging** – If non-zero, neighbor measurement reports are averaged. To be considered for handover a neighbor must appear in 2 of the last GSM.Neighbors.Averaging measurement reports sent by the MS.
• **GSM.RRLP.ACCURACY** – Requested accuracy of location request. K in \( r = 10 \cdot (1.1^K - 1) \), where \( r \) is the accuracy in meters. See 3GPP 03.32 Sec 6.2.

• **GSM.RRLP.ALMANAC.ASSIST.PRESENT** – Send almanac info to mobile.

• **GSM.RRLP.ALMANAC.REFRESH.TIME** – How often the almanac is refreshed, in hours.

• **GSM.RRLP.ALMANAC.URL** – URL of the almanac source.

• **GSM.RRLP.EPHEMERIS.ASSIST.COUNT** – Number of satellites to include in navigation model.

• **GSM.RRLP.EPHEMERIS.REFRESH.TIME** – How often the ephemeris is refreshed, in hours.

• **GSM.RRLP.EPHEMERIS.URL** – URL of ephemeris source.

• **GSM.RRLP.RESPONSETIME** – Mobile timeout. (OpenBTS timeout is 130 sec = max response time + 2.) N in \( 2^{**N} \). See 3GPP 04.31 Sec A.2.2.1.

• **GSM.RRLP.SEED._ALTITUDE** – Seed altitude in meters wrt geoidal surface.

• **GSM.RRLP.SEED.LATITUDE** – Seed latitude in degrees: -90 (south pole) .. +90 (north pole).

• **GSM.RRLP.SEED.LONGITUDE** – Seed longitude in degrees: -180 (west of greenwich) .. +180 (east).

• **GSM.RRLP.SERVER.URL** – URL of RRLP server. By default, this feature is disabled. To enable, specify a server URL eg: http://localhost/cgi/rrlpserver.cgi. To disable again, execute "unconfig GSM.RRLP.SERVER.URL".

• **GSM.Radio.NeedBSIC** – Whether the Radio type requires the full BSIC.

• **GSM.Radio.PowerManager.NumSamples** – Number of samples averaged by the output power control loop.

• **GSM.Radio.PowerManager.Period** – Power manager control loop master period, in milliseconds.

• **GSM.Radio.PowerManager.SamplePeriod** – Sample period for the output power control loop in milliseconds.

• **GSM.Radio.PowerManager.TargetT3122** – Target value for T3122, the random access hold-off timer, for the power control loop.

• **GSM.Radio.RxGain** – Receiver gain setting in dB. Ideal value is dictated by the hardware; 47 dB for RAD1. This database parameter is static but the receiver gain can be modified in real time with the CLI “rxgain” command.

• **GSM.Timer.T3103** – Handover timeout in milliseconds, GSM 04.08 11.1.2. This is the timeout for a handset to seize a channel during handover.

• **GSM.Timer.T3105** – Milliseconds for handset to respond to physical information. GSM 04.08 11.1.2.

• **GSM.Timer.T3113** – Paging timer T3113 in milliseconds. This is the timeout for a handset to respond to a paging request. This should usually be the same as SIP.Timer.B in your VoIP network.

• **GSM.Timer.T3122Max** – Maximum allowed value for T3122, the RACH holdoff timer, in milliseconds. This timer is sent to the MS with a granularity of seconds in the range 1-255. GSM 4.08 10.5.2.43.
B.1. OPENBTS PARAMETERS

- **GSM.Timer.T3122Min** – Minimum allowed value for T3122, the RACH holdoff timer, in milliseconds. GSM 4.08 10.5.2.43. This timer is sent to the MS with a granularity of seconds in the range 1-255. The purpose is to postpone the MS RACH procedure until an SDCCH available, so there is no point making it any smaller than the expected availability of the SDCCH, which will take several seconds.

- **Log.File** – Path to use for textfile based logging. By default, this feature is disabled. To enable, specify an absolute path to the file you wish to use, eg: /tmp/my-debug.log. To disable again, execute “unconfig Log.File”.

- **Peering.ResendCount** – Number of tries to send message over the peer interface before giving up.

- **Peering.ResendTimeout** – Milliseconds before resending a message on the peer interface.

- **SGSN.Debug** – Add layer 3 messages to the GGSN.Logfile, if any.

- **SGSN.Timer.ImplicitDetach** – 3GPP 24.008 11.2.2. GPRS attached MS is implicitly detached in seconds. Should be at least 240 seconds greater than SGSN.Timer.RAUpdate.

- **SGSN.Timer.MS.Idle** – How long an MS is idle before the SGSN forgets TLLI specific information.

- **SGSN.Timer.RAUpdate** – Also known as T3312, 3GPP 24.008 4.7.2.2. How often MS reports into the SGSN when it is idle, in seconds. Setting to 0 or 12000 deactivates entirely, i.e., sets the timer to effective infinity. Note: to prevent GPRS Routing Area Updates you must set both this and GSM.Timer.T3212 to 0.

- **SGSN.Timer.Ready** – Also known as T3314, 3GPP 24.008 4.7.2.1. Inactivity period required before MS may perform another routing area or cell update, in seconds.

- **SIP.DTMF.RFC2967** – Obsolete; incorrect RFC number. Use SIP.DTMF.RFC2976.

- **SIP.MaxForwards** – Maximum allowed number of referrals.

- **SIP.Proxy.Mode** – If set to direct, then direct BTS to BTS calls are permitted without an intervening SIP switch, for example, no asterisk needed.

- **SIP.RegistrationPeriod** – Registration period in minutes for MS SIP users. Should be longer than GSM T3212.

- **SIP.Timer.A** – SIP timer A, the INVITE retry period, RFC-3261 Section 17.1.1.2, in milliseconds.

- **SIP.Timer.B** – INVITE transaction timeout in milliseconds. This value should usually match GSM.Timer.T3113.

- **SIP.Timer.E** – Non-INVITE initial request retransmit period in milliseconds.

- **SIP.Timer.F** – Non-INVITE initial request timeout in milliseconds.

- **SIP.Timer.H** – ACK timeout period in milliseconds.

- **TRX.MinimumRxRSSI** – Bursts received at the physical layer below this threshold are automatically ignored. Values in dB. Set at the factory. Do not adjust without proper calibration.

- **TRX.Port** – IP port of the transceiver application.

- **TRX.RadioFrequencyOffset** – Fine-tuning adjustment for the Transceiver master clock. Roughly 170 Hz/step. Set at the factory. Do not adjust without proper calibration.
• TRX.Timeout.Clock – How long to wait during a read operation from the Transceiver before giving up.
• TRX.Timeout.Start – How long to wait during system startup before checking to see if the Transceiver can be reached.
• TRX.TxAttenOffset – Hardware-specific gain adjustment for transmitter, matched to the power amplifier, expressed as an attenuation in dB. Set at the factory. Do not adjust without proper calibration.
• Test.GSM.SimulatedFER.Downlink – Probability (0-100) of dropping any downlink frame to test robustness.
• Test.GSM.SimulatedFER.Uplink – Probability (0-100) of dropping any uplink frame to test robustness.
• Test.GSM.UplinkFuzzingRate – Probability (0-100) of flipping a bit in any uplink frame to test robustness.
• Test.SIP.SimulatedPacketLoss – Probability (0-100) of dropping any inbound or outbound SIP packet to test robustness.

B.2 Smqueue Parameters

Smqueue configuration parameters are stored in Smqueue Configuration Table located at "/etc/OpenBTS/smqueue.db". The schema is identical to the OpenBTS Configuration Table discussed in section 4.1.

B.2.1 Customer Site Parameters

None defined at this time.

B.2.2 Customer Tuneable Parameters

These parameters can be changed to optimize your site.

• Asterisk.address – The Asterisk/SIP PBX IP address and port.
• Bounce.Code – The short code that bounced messages originate from.
• Bounce.Message.IMSILookuupFailed – The bounce message that is sent when the originating IMSI cannot be verified.
• Bounce.Message.NotRegistered – Bounce message indicating that the destination phone is not registered.
• CDRFile – Log CDRs here. To enable, specify an absolute path to where the CDRs should be logged. To disable, execute "unconfig CDRFile".
• Log.Alarms.Max – Maximum number of alarms to remember inside the application.
• Log.Level – Default logging level when no other level is defined for a file.
B.2. SMQUEUE PARAMETERS

- **SC.DebugDump.Code** – Short code to the application which dumps debug information to the log. Intended for administrator use.
- **SC.Info.Code** – Short code to the application which tells the sender their own number and registration status.
- **SC.QuickChk.Code** – Short code to the application which tells the sender the how many messages are currently queued. Intended for administrator use.
- **SC.Register.Code** – Short code to the application which registers the sender to the system.
- **SC.Register.Digits.Max** – The maximum number of digits a phone number can have.
- **SC.Register.Digits.Min** – The minimum number of digits a phone number must have.
- **SC.Register.Digits.Override** – Ignore phone number digit length checks.
- **SC.Register.Msg.AlreadyA** – First part of message sent during registration if the handset is already registered, followed by the current handset number.
- **SC.Register(Msg.AlreadyB** – Second part of message sent during registration if the handset is already registered.
- **SC.Register.Msg.ErrorA** – First part of message sent during registration if the handset fails to register, followed by the attempted handset number.
- **SC.Register.Msg.ErrorB** – Second part of message sent during registration if the handset fails to register, followed by the handset IMSI.
- **SC.Register.Msg.TakenA** – First part of message sent during registration if the handset fails to register because the desired number is already taken, followed by the attempted handset number.
- **SC.Register(Msg.TakenB** – Second part of message sent during registration if the handset fails to register because the desired number is already taken.
- **SC.Register.Msg.WelcomeA** – First part of message sent during registration if the handset registers successfully, followed by the assigned handset number.
- **SC.Register(Msg.WelcomeB** – Second part of message sent during registration if the handset registers successfully.
- **SC.SMSC.Code** – The SMSC entry point. There is where OpenBTS sends SIP MESSAGES to.
- **SIP.Default.BTSPort** – The default BTS port to try when none is available.
- **SIP.GlobalRelay.ContentType** – The content type that the global relay expects.
- **SIP.GlobalRelay.IP** – IP address of global relay to send unresolvable messages to. By default, this is disabled. To override, specify an IP address. To disable again use “unconfig SIP.GlobalRelay.IP”.
- **SIP.GlobalRelay.Port** – Port of global relay to send unresolvable messages to.
- **SIP.GlobalRelay.RelaxedVerify** – Relax relay verification by only using SIP Header.
- **SIP.Timeout.ACKedMessageResend** – Number of seconds to delay resending ACK messages.
• **SIP.Timeout.MessageBounce** – Timeout, in seconds, between bounced message sending tries.
• **SIP.Timeout.MessageResend** – Timeout, in seconds, between message sending tries.
• **SIP.myIP** – The internal IP address. Usually 127.0.0.1.
• **SIP.myIP2** – The external IP address that is communicated to the SIP endpoints.
• **SIP.myPort** – The port that smqueue should bind to.
• **SMS.FakeSrcSMSC** – Use this to fill in L4 SMSC address in SMS delivery.
• **SMS.HTTPGateway.Retries** – Maximum retries for HTTP gateway attempt.
• **SMS.HTTPGateway.Timeout** – Timeout for HTTP gateway attempt in seconds.
• **SMS.HTTPGateway.URL** – URL for HTTP API. Used directly as a C format string with two “
• **SMS.MaxRetries** – Messages will only be attempted to be sent this many times before giving up and being dropped. Set to 0 to allow infinite retries.
• **SMS.RateLimit** – Limit delivery rate to one message every X seconds. Set to 0 to disable rate limiting.
• **SubscriberRegistry.A3A8** – Path to the program that implements the A3/A8 algorithm.
• **SubscriberRegistry.Port** – Port used by the SIP Authentication Server. NOTE: In some older releases (pre-2.8.1) this is called SIP.myPort.
• **SubscriberRegistry.UpstreamServer** – URL of the subscriber registry HTTP interface on the upstream server. By default, this feature is disabled. To enable, specify a server URL eg: http://localhost/cgi/subreg.cgi.
• **SubscriberRegistry.db** – The location of the sqlite3 database holding the subscriber registry.
• **savefile** – The file to save SMS messages to when exiting.

**B.2.3 Developer/Factory Parameters**

These parameters should only be changed by when developing new code.

• **Control.NumSQLTries** – Number of times to retry SQL queries before declaring a database access failure.
• **Debug.print_as_we_validate** – Generate lots of output during validation.
• **Log.File** – Path to use for textfile based logging. By default, this feature is disabled. To enable, specify an absolute path to the file you wish to use, eg: /tmp/my-debug.log.
• **SC.WhiplashQuit.Code** – Short code to the application which will make the server quit for valgrind leak checking. Intended for developer use only.
• **SC.WhiplashQuit.Password** – Password which must be sent in the message to the application at SC.WhiplashQuit.Code.
• **SC.WhiplashQuit.SaveFile** – Contents of the queue will be dumped to this file when SC.WhiplashQuit.Code is activated.
• **SC.ZapQueued.Code** – Short code to the application which will remove a message from the queue, by its tag. If first char is “-”, do not reply, just do it. If argument is **SC.ZapQueued.Password**, then delete any queued message with timeout greater than 5000 seconds.

• **SC.ZapQueued.Password** – Password which must be sent in the message to the application at **SC.ZapQueued.Code**.

### B.3 SIPAuthServe Parameters

SIPAuthServe configuration parameters are stored in SIPAuthServe Configuration Table located at “/etc/OpenBTS/sipauthserve.db”. The schema is identical to the OpenBTS Configuration Table discussed in section 4.1.

#### B.3.1 Customer Site Parameters

None defined at this time.

#### B.3.2 Customer Tuneable Parameters

These parameters can be changed to optimize your site.

• **Log.Alarms.Max** – Maximum number of alarms to remember inside the application.

• **Log.Level** – Default logging level when no other level is defined for a file.

• **SubscriberRegistry.A3A8** – Path to the program that implements the A3/A8 algorithm.

• **SubscriberRegistry.Port** – Port used by the SIP Authentication Server. NOTE: In some older releases (pre-2.8.1) this is called **SIP.myPort**.

• **SubscriberRegistry.db** – The location of the sqlite3 database holding the subscriber registry.

#### B.3.3 Developer/Factory Parameters

These parameters should only be changed by when developing new code.

• **Control.NumSQLTries** – Number of times to retry SQL queries before declaring a database access failure.

• **Log.File** – Path to use for textfile based logging. By default, this feature is disabled. To enable, specify an absolute path to the file you wish to use, eg: /tmp/my-debug.log.
# Appendix C

## The Command Line Interface (CLI) Reference

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.1</td>
<td>“alarms” Command</td>
<td>103</td>
</tr>
<tr>
<td>C.2</td>
<td>“audit” Command</td>
<td>103</td>
</tr>
<tr>
<td>C.3</td>
<td>“calls” Command</td>
<td>103</td>
</tr>
<tr>
<td>C.4</td>
<td>“cellid” Command</td>
<td>104</td>
</tr>
<tr>
<td>C.5</td>
<td>“chans” Command</td>
<td>104</td>
</tr>
<tr>
<td>C.6</td>
<td>“config” Commands</td>
<td>105</td>
</tr>
<tr>
<td>C.7</td>
<td>“devconfig” Command</td>
<td>106</td>
</tr>
<tr>
<td>C.8</td>
<td>“endcall” Command</td>
<td>106</td>
</tr>
<tr>
<td>C.9</td>
<td>“freqcorr” Command</td>
<td>107</td>
</tr>
<tr>
<td>C.10</td>
<td>“gprs” Command</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>C.10.1 “gprs list” Subcommand</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>C.10.2 Other “gprs” Subcommands</td>
<td>108</td>
</tr>
<tr>
<td>C.11</td>
<td>“handover” Command</td>
<td>108</td>
</tr>
<tr>
<td>C.12</td>
<td>“help” Command</td>
<td>108</td>
</tr>
<tr>
<td>C.13</td>
<td>“load” Command</td>
<td>108</td>
</tr>
<tr>
<td>C.14</td>
<td>“memstat” Command</td>
<td>109</td>
</tr>
<tr>
<td>C.15</td>
<td>“neighbors” Command</td>
<td>109</td>
</tr>
<tr>
<td>C.16</td>
<td>“noise” Command</td>
<td>109</td>
</tr>
<tr>
<td>C.17</td>
<td>“notices” Command</td>
<td>109</td>
</tr>
<tr>
<td>C.18</td>
<td>“page” Command</td>
<td>109</td>
</tr>
<tr>
<td>C.19</td>
<td>“power” Command</td>
<td>110</td>
</tr>
<tr>
<td>C.20</td>
<td>“rawconfig” Command</td>
<td>110</td>
</tr>
<tr>
<td>C.21</td>
<td>“regperiod” Command</td>
<td>111</td>
</tr>
<tr>
<td>C.22</td>
<td>“rmconfig” Command</td>
<td>111</td>
</tr>
<tr>
<td>C.23</td>
<td>“rgain” Command</td>
<td>111</td>
</tr>
<tr>
<td>C.24</td>
<td>“sendsimple” and “ sendsms” Commands</td>
<td>112</td>
</tr>
<tr>
<td>C.25</td>
<td>“sgsn” Command</td>
<td>112</td>
</tr>
<tr>
<td>C.26</td>
<td>“shutdown” Command</td>
<td>112</td>
</tr>
</tbody>
</table>
C.1  “alarms” Command

List recent alarms. The number of alarms shown in the list is set by the “Log.Alarms.Max” configuration value.

C.2  “audit” Command

Examines the current configuration and reports possible issues. Troubleshooting information includes the following severity levels and sections:

- ERROR – keys with invalid values,
- WARNING – keys which differ from factory radio calibration values (only available with Range Networks radio hardware),
- WARNING – interaction among values which could cause problems,
- WARNING – site specific keys which are still set to their default values,
- INFO – keys with non-default values,
- INFO – deprecated key/value pairs.

C.3  “calls” Command

List in-progress Q.931 and SMS transactions from the transaction table (Section 4.4). Displayed information includes:
• transaction id – The key for the corresponding entry in the transaction table that is currently making use of this channel.
• SIP call state
• Q.931/GSM call state
• time since last state change
• subscriber IMSI
• called or calling party number

C.4 “cellid” Command

Display or change cell identity parameters. These parameters are:

• MCC – Mobile Country Code (3 digits)
• MNC – Mobile Network Code (2 or 3 digits)
• LAC – Location Area Code (16 bits, 1-65520 are valid values)
• CI – Cell Identity (16 bits, 0-65535 are valid values)

With no arguments, this command displays the current MCC, MNC, LAC and CI values. With arguments

cellid <MCC> <MNC> <LAC> <CI>

this command sets the parameters to the given values and updates the corresponding GSM Identity configuration table parameters, as described in Section 4.2. Using the command with arguments will also cause the TMSI Table to be cleared.

C.5 “chans” Command

This command displays physical channel status from the channel table (Section 4.5) for active dedicated channels.

The command accepts the following options:

• -a – report for all channels
• -l – longer listing
• -tab – tab-separated output format

The reported values are:
C.6. “CONFIG” COMMANDS

- CN – Channel number;
- TN – Timeslot number;
- chan type – The dedicated channel type, or GPRS if reserved for Packet Services;
- transaction id – The key for the corresponding entry in the transaction table that is currently making use of this channel;
- LAPDm state – The current acknowledged message state, if any, otherwise ‘active’ or ‘inactive’;
- recyc – ‘true’ if channel is recyclable, i.e., can be reused now;
- RSSI – Uplink signal level dB above noise floor measured by BTS, should be near config parameter “GSM.Radio.RSSITarget”;
- SNR – Signal to Noise Ratio measured by the BTS; higher is better, less than 10 is likely unusable;
- BER – Bit Error Rate before decoding measured by the BTS, as a percentage;
- FER – voice frame loss rate as a percentage measured by the BTS;
- TA – Timing advance in symbol periods measured by the BTS;
- TXPWR – Uplink transmit power dB reported by MS;
- TXTA – Timing advance in symbol periods reported by MS;
- DNLEV – Downlink signal level dB reported by MS;
- DNBER – Downlink bit error rate percentage reported by MS;
- IMSI – International Mobile Subscriber Id of the MS on this channel, reported only if known;
- Frames – number of bad, stolen, and total frames sent; only for traffic channels;
- Neighbor ARFCN and dBm – One of the neighbors channel and downlink RSSI reported by the MS, may also be:
  - ”no-MMContext” to indicate the layer2 channel is open but has not yet sent any layer3 messages, or
  - ”no-MMUser” to indicate that layer3 is connected but the IMSI is not yet known.

Note: to view GPRS channel information, use the “gprs stat” command.

C.6 “config” Commands

This command displays and modifies parameters in the configuration table (Section 4.2). It is possibly the most useful and powerful command in the interface.

When issued without arguments, the “config” command lists all user-level configuration parameters.

To list the configuration parameters containing a given pattern, type:
config <pattern>

The “config” command followed by a valid configuration parameter (key) shows the key’s complete description, current value, default value and valid values:

```bash
config <key>
```

Example:

```bash
OpenBTS> config GSM.Identity.MCC
GSM.Identity.MCC 001 [default]
- type: string
- default value: 001
- visibility level: customer site - these values are different for each BTS and should not be left default
- static: 0
- valid val regex: ^[0-9]{3}$
- scope: value must be the same across all nodes
```

The “config” command treats two arguments as a key-value pair and sets the key to a new value:

```bash
config <key> <value>
```

Example:

```bash
OpenBTS> config GSM.Identity.MCC 901
GSM.Identity.MCC changed from "001" to "901"
```

Not all configuration parameters can be accessed via the “config” command. See “devconfig” and “rawconfig” commands for further details. Also see the “rmconfig” command which resets parameters to their default values.

C.7 “devconfig” Command

Certain parameters are not meant to be edited in the normal workflow, but might require tweaking in the development environment. By default, “config” hides these parameters from the user. To review and manipulate them, use the less restrictive “devconfig” command. See Appendix section ?? for information on specific configuration values and their effect of the system.

C.8 “endcall” Command

Force the termination of a call or other transaction.

```bash
endcall <transactionID>
```
C.9  “freqcorr” Command

This command reads the radio frequency offset and allows to set a different offset. The format is as follows:

OpenBTS> freqcorr
  current freq. offset is 128

C.10  “gprs” Command

This command is used to invoke a number of subcommands the GPRS subsystem. The syntax is: “gprs subcommand <options...>”

C.10.1  “gprs list” Subcommand

The “list” subcommand prints active objects of specified type.

gprs list [ms|tbf|ch] [-v] [-x] [-c] [<id>]

If “ms”, “tbf” or “ch” is specified, the listing is limited to that type of entity. If the optional <id> is specified, it is the numeric ordinal identifier for an MS or TBF. This is the same id that is part of the name of the MS or TBF. For example, to list just "MS#1" the id is "1".

The options for “gprs list” are:
  -v  Verbose listing.
  -c  include MS capabilities in the listing.
  -x  list expired rather than active entities

When invoked without options, “gprs list” prints a short listing of the MS, TBF and Channels in use by GPRS:

OpenBTS> gprs list
  PDCH ARFCN=51 TN=1 FER=0%
  PDCH ARFCN=51 TN=2 FER=0%
C.10.2 Other “gprs” Subcommands

- **gprs console [0|1]**: Send messages to console as well as the Log.File (default=1 for debugging).
- **gprs debug [level]**: Set debug level; 0 turns debugging off.
- **gprs free ms|tbf|ch <id>**: Free the corresponding entity.
- **gprs freeex**: Free expired ms and tbf structs.
- **gprs help**: Print the complete list of commands available from the GPRS subsystem.
- **gprs mem**: Memory leak detector - print numbers of structs in use.
- **gprs set name [val]**: Print and optionally set a variable, see source for names.
- **gprs start [step]**: Start GPRS, optionally in single-step-mode.
- **gprs stat**: Print GPRS statistics, including total number of channels, MS, and TBF allocated.
- **gprs step**: Single step the MAC service loop (requires 'start step').
- **gprs stop -c**: Stop GPRS thread (and release channels with optional '-c').
- **gprs rach**: Simulate a RACH, which starts gprs service.
- **gprs testbsn**: Test bsn–¿frame number functions.
- **gprs testmsg**: Test message functions.

C.11 “handover” Command

The “handover” command attempts a handover of a particular IMSI to a neighbor specified by its ip address. The parameters are as follows:

```
handover imsi neighbor
```

C.12 “help” Command

The “help” command provides a list of all commands available in the CLI. “help” followed by a command name provides help on that command. For example:

```
OpenBTS> help cellid
cellid [MCC MNC LAC CI] -- get/set location area identity (MCC, MNC, LAC) and cell ID (CI)
```

C.13 “load” Command

Give the current BTS load, in terms of active channels and queue lengths.

```
load
```

The results mean:

- **SDCCH load** – The number of active SDCCHs out of the total available.
- **TCH/F load** – The number of active TCH/Fs out of the total available.
C.14. “MEMSTAT” COMMAND

- AGCH/PCH load – The number of queued messages awaiting transmission on the AGCH and PCH.
- Paging table size – The number of MSs currently being paged.
- Transactions/TMSIs – The number of active transactions in the BTS and the size of the TMSI table.
- T3122 – The current value of the T3122 hold-off timer, in seconds. See Section 5.1 for details.
- current PDCHs – GRPS active channels
- utilization % – GPRS channel utilization

C.14 “memstat” Command

The development command used for internal testing. It prints memory use statistics.

C.15 “neighbors” Command

This command displays the current neighbor table, as generated through automatic neighbor information exchange. Each line contains:

- the neighbor’s IP address
- the neighbor’s C0 ARFCN
- the neighbor’s BSIC

C.16 “noise” Command

The “noise” command reports receive noise level in RSSI dB as follows:

```
OpenBTS> noise
noise RSSI is -67 dB wrt full scale
MS RSSI target is -50 dB wrt full scale
INFO: the current noise level is acceptable.
```

C.17 “notices” Command

Print the copyright and legal notices associated with your installation of OpenBTS.

C.18 “page” Command

Print the paging table.
C.19 "power" Command

Inspect or change the downlink power parameters described in Section 5.1. With no arguments, this command displays the current power setting and bounds. With arguments this command changes the power control bounds.

power <minAtten> <maxAtten>

For example:

OpenBTS> power 10 20
current downlink power 0 dB wrt full scale
current attenuation bounds 0 to 10 dB
new attenuation bounds 10 to 20 dB

C.20 "rawconfig" Command

The "rawconfig" command behaves much like "config" but with two additional features. It can be used to define and manipulate custom key-value pairs in the configuration table. It also ignores any input validation allowing developers to enter experimental values for existing configuration keys.

The command is most commonly used to set custom log levels for individual system components for troubleshooting purposes as shown in the following example.

The user wishes to log more Layer 3 Call Control information:

OpenBTS> rawconfig Log.Level.CallControl.cpp INFO
defined new config Log.Level.CallControl.cpp as "INFO"

OpenBTS> rawconfig Log.Level.CallControl.cpp
Log.Level.CallControl.cpp INFO

Note: The "unconfig" and "rmconfig" commands can also be used on these custom key-values.

OpenBTS> unconfig Log.Level.CallControl.cpp
Log.Level.CallControl.cpp disabled

OpenBTS> rawconfig Log.Level.CallControl.cpp
Log.Level.CallControl.cpp (disabled)

OpenBTS> rmconfig Log.Level.CallControl.cpp
Log.Level.CallControl.cpp removed from the configuration table

OpenBTS> rawconfig Log.Level.CallControl.cpp

OpenBTS>
C.21 “regperiod” Command

The “regperiod” command prints or sets the registration timer GSM T3212 and SIP registration period.

```
OpenBTS> regperiod
T3212 is 0 minutes
SIP registration period is 90 minutes
```

In order to modify both parameters, type: “regperiod [GSM] [SIP]”, where GSM is the value of the “GSM.Timer.T3212” parameter in minutes, and SIP is the value of “SIP.RegistrationPeriod”, also in minutes.

```
OpenBTS> regperiod 60 72
OpenBTS> config GSM.Timer.T3212
GSM.Timer.T3212 60
OpenBTS> config SIP.RegistrationPeriod
SIP.RegistrationPeriod 72
```

Note that the “GSM.Timer.T3212” parameter should have a value which is a factor of 6. It should be smaller than the SIP registration period. Setting it to “0” disables periodic registration.

C.22 “rmconfig” Command

The “rmconfig” command followed by a key reverts the corresponding configuration parameter back to its default value. If the key is a custom (user-defined) key, it gets completely removed from the configuration table.

```
rmconfig <key>
```

Example:

```
OpenBTS> rmconfig GSM.Identity.MCC
GSM.Identity.MCC set back to its default value
```

C.23 “rxgain” Command

The “rxgain” command prints or sets the receiver gain.

When called without parameters, it prints the current RX gain in dB, stored in the “GSM.Radio.RxGain” parameter:

```
OpenBTS> rxgain
current RX gain is 47 dB
```
The ideal value for this parameter is dictated by the hardware; 47 dB for Range SDR1. This database parameter is static but the "rxgain" command can modify the receiver gain in real time:

```
OpenBTS> rxgain 52
  current RX gain is 47 dB
  new RX gain is 52 dB
```

### C.24 "sendsimple" and "sendsms" Commands

Both of these commands send a text message via SMS to a given MS, addressed by IMSI appearing to originate from a specified source address.

```
sendsimple <IMSI> <sourceAddress> <message test>
sendsms <IMSI> <sourceAddress> <message text>
```

The difference between the two is that "sendsimple" operates by sending an RFC-3428 SIP MESSAGE packet to the OpenBTS SIP port, while "sendsms" operates directly in the SMS control layers of OpenBTS.

### C.25 "sgsn" Command

The "sgsn" command provides sub-commands to control SGSN/GGSN sub-system. The syntax is:

```
sgsn subcommand <options...>
```

The following sub-commands are currently available:

- `free (imsi|tlli) id`
- `help` – print help
- `list [(imsi|tlli) id]` – print the list of current GPRS sessions tracked by the SGSN, normally one per GPRS-attached handset

### C.26 "shutdown" Command

The "shutdown" command shuts down the OpenBTS and transceiver processes. If an argument is provided, the command will wait up to the given number of seconds for in-progress calls and transactions to clear before exiting. During this wait time, no new calls or transactions will be allowed to start. The "shutdown" command with no arguments exits the OpenBTS process immediately.

In embedded configurations from Range, OpenBTS runs as an upstart service, and therefore the OpenBTS GSM stack and its transceiver will restart automatically after a shutdown. However, the CLI would not be able to reconnect to the new instance of OpenBTS, and will require a restart. This is an expected behavior.
C.27 “STATS” Command

Prints performance counters. The format is:

```
stats [patt] | clear
```

By default prints all counters. When a pattern is provided, prints only the counters matching the pattern. When issued with the “clear” option, clears all counters.

C.28 “sysinfo” Command

Prints current system information messages.

C.29 “tmsis” Command

This command manipulates the TMSI table (Section 4.3). The syntax is:

```
tmsis [-a | -l | -ll | -r | clear |
       dump [-l] <filename> |
       delete -tmsi <tmsi> |
       delete -imsi <imsi> |
       query <query> set name=value]
```

The default action is to print the most recent 100 of TMSI table records. The following options apply:

- `-a` – list all records rather than the last 100
- `-l`, `-ll` – longer listing
- `-r` – raw TMSI table listing

Other possible actions are:

- `clear` – clear the TMSI table
- `delete -imsi <imsi>` or `delete -tmsi <tmsi>` – delete an entry for the specified IMSI or TMSI
- `dump [-l] <filename>` – dump the TMSI table to a specified filename
- `set name=value` – set TMSI database field name to value. If value is a string use apostrophes, eg: `set IMSI='12345678901234'`
C.30 “trans” Command

Prints completed transaction table in the tabular format. The data in the table includes:

- “Active” – whether the transaction is active or not (always “no”)
- “TranId” – Transaction ID
- “L3TI”
- “Service”
- “To” and “From”
- “AgeSec” – transaction age in seconds
- “Start Time” and “End Time”
- “Message”

Issued with the optional purge parameter the command purges the transaction records table.

C.31 “trxfactory” Command

This command displays information stored in the radio at the factory (only compatible with Range Networks hardware):

- SDR Serial Number
- RF Serial Number
- GSM.Radio.Band
- TRX.RadioFrequencyOffset
- GSM.Radio.RxGain
- TRX.TxAttenOffset

**Note:** This command does not work in version 4.0.0. It is a known issue.

C.32 “txatten” Command

When issued without parameters, this command prints the transmitter attenuation in dB. When issued with a parameter, it sets the attenuation to the corresponding value. For example:
C.33  “unconfig” Command

The “unconfig” command sets the value associated with the given key to an empty string in the configuration table, effectively disabling it. This command does not have effect on configuration parameters, for which empty string is not a valid value.

unconfig <key>

Example:

OpenBTS> config Control.LUR.OpenRegistration ^001
Control.LUR.OpenRegistration is already set to "^001", nothing changed

OpenBTS> unconfig Control.LUR.OpenRegistration
Control.LUR.OpenRegistration disabled

C.34  “uptime” Command

Shows BTS uptime and BTS frame number.

C.35  “version” Command

Display version and revision information on the installed version of OpenBTS.

C.36  Executing OS shell commands from the CLI

Commands prepended with an exclamation mark are passed to OS shell for execution:

    ! grep Register /var/log/OpenBTS.log | grep IMSI | tail -n 10
Appendix D

Logging in OpenBTS

In embedded Range systems OpenBTS logs to rsyslogd or syslogd as facility “local7” as well as a log-file located at “/var/log/OpenBTS.log”.

OpenBTS defines the syslogd logging levels to mean the following:

- **EMERGENCY**: serious fault associated with service failure
- **WARNING**: likely service disruption caused by misconfiguration, poor connectivity or some other problem not internal to the software
- **CRITICAL**: anomalous event that is likely to degrade service
- **ERROR**: an internal error of the software that may result in degradation of service in unusual circumstances
- **WARNING**: an anomalous event that may indicate a degradation of service
- **NOTICE**: an anomalous event that probably does not affect service but may be of interest to network operators
- **INFO**: a normal event
- **DEBUG**: detailed information about internal data structures

The overall logging level for OpenBTS is set in the configuration variable “Log.Level”. Logging levels can be set for individual source files by defining a custom configuration variable of the form “Log.Level.filename” with a value equal to the desired logging level. These log levels are dynamic and can be set and changed in real time with the “rawconfig” command (Section C.20).

For example, “rawconfig Log.Level.CallControl.cpp INFO” sets the logging level to INFO for all functions in the file CallControl.cpp.

Some useful logging settings are:

- `rawconfig Log.Level.GSML2LAPDm.cpp INFO` – for an L2 trace
- `rawconfig Log.Level.MobilityManagement.cpp INFO` – for an L3 MM trace
- `rawconfig Log.Level.CallControl.cpp INFO` – for an L3 CC trace
- `rawconfig Log.Level.SIPInterface.cpp INFO` – for a trace of all SIP messages
- `rawconfig Log.Level.SIPEngine.cpp INFO` – for a trace of SIP state machine activity
- `rawconfig Log.Level.SMSControl.cpp INFO` – for a trace of L3 SMS activity

Log events at the “CRIT”, “ALERT” and “EMERGENCY” levels are treated as special cases inside OpenBTS:
• High level log events are echoed to the OpenBTS stdout, regardless of the Log.LogFile and Log.Level settings or the configuration of syslogd.

• High level log events are stored in an internal table accessible from the CLI (Section C.1). The maximum size of this table is set with the Log.Alarms.Max configuration value.

Certain components of OpenBTS can be configured to log their activity to their own log files. In order to enable a certain type of log, set the corresponding configuration parameter to a desired file path using the “devconfig” command.

• GGSN.Logfile.Name – setting this developer-level parameter enables logging of internet traffic from GPRS.

• Log.File – setting this parameter enables textfile logging of OpenBTS activity.

Use “rmconfig <logging-related-parameter>” to disable the particular type of log again.

SIPAuthServe and SMQueue have their own log level settings, which can be configured using the OpenRAN UI or by manually editing the corresponding sqlite3 configuration database.

Note: Excessive logging can affect the operation of the BTS.
Appendix E

Multi-BTS Networks

E.1 How Mobility Works

E.1.1 How Mobility Works In GSM

There are two mobility mechanisms in GSM that are implemented in OpenBTS: the location area and the neighbor list.

“Mobility” is the ability to transfer an MS’s service as it moves from one BTS unit to another in a multi-BTS network so that inbound calls for that MS get routed to it correctly. Mobility is one of the defining features of a cellular network. “Handover” is the transfer of an active call from one BTS unit to the other. OpenBTS supports both mobility and handover.

E.1 How Mobility Works

E.1.1 How Mobility Works In GSM

There are two mobility mechanisms in GSM that are implemented in OpenBTS: the location area and the neighbor list.
E.1. HOW MOBILITY WORKS

Location Areas

A GSM network is divided into geographical regions called “location areas”, each one typically served by a single BSC. Every BTS broadcasts its location area code (LAC) on the BCCH. In OpenBTS, the LAC is controlled with the GSM.Identity.LAC configuration parameter, which is dynamic and can be altered in real time. An MS performs a location updating request whenever it enters a new location area. When an MS is paged for an inbound call, the paging request is sent to all of the cells in the location area in which an MS is registered. The implication here is that the GSM core network does not need to know the specific cell that is serving an MS, only its location area. However, if every cell in the network has a different LAC, the MS will perform a location updating request every time it moves to a new cell.

Neighbor Lists

Every BTS broadcasts a list of ARFCNs of neighboring cells, called the “neighbor list”, on the BCCH. This list is also sent on the SACCH during transactions and calls. The MS monitors these ARFCNs, measuring their power levels and decoding their SCH bursts. In idle mode, the MS compares these power levels to the power level of its serving cell. When the power level of the strongest neighbor exceeds that of the serving cell by a given threshold, the MS will recamp from the current serving cell to the strong neighbor. The power difference threshold at which this happens is called the “cell reselection hysteresis”, a parameter broadcast on the BCCH. If the LAC of the new cell is different from the previous cell, the MS will also make a location updating request to ensure proper routing of inbound transactions.

BTS Configuration

In the OpenBTS configuration, the following parameters are relevant for mobility:

- GSM.Radio.C0 & GSM.Radio.ARFCNs,
- all GSM.Identity.* parameters,
- all GSM.CellSelection.* parameters.

See Section 4.2 for information about these parameters. Many of these parameters have been assigned the default values that would work for most multi-BTS applications, however, the following parameters must be provided by the operator:

- GSM.Radio.C0 & GSM.Radio.ARFCNs, although GSM.Radio.ARFCNs is usually dictated by the hardware,
- all GSM.Identity.* parameters,
- GSM.CellSelection.NCCsPermitted, which must include the NCC of the network.

E.1.2 How Mobility Works in SIP

From the point of view of a SIP switch, a mobile SIP user is a user whose IP address changes. In Asterisk and the subscriber registry, Asterisk supports this with the “host=dynamic” qualifier in the SIP user profile. The
subscriber registry keeps track of the IP address from which a user last registered. (These IP addresses can be observed from the Asterisk console with the “sip show peers” command.) Once a user is registered to a given IP address, all inbound calls for that user are routed to that address.

E.1.3 Combined GSM-SIP Mobility in OpenBTS

OpenBTS translates every GSM location updating operation into a SIP registration transaction, as explained in Section 6.5.1. The MS performs a location update every time it moves into a new location area. If we give every BTS a different LAC the MS will perform a location update every time it camps to a new cell, resulting in a SIP registration that updates the MS’s associated IP address in the Subscriber Registry. Because SMQueue also uses the Subscriber Registry for address resolution and message routing, SMS routing will be updated as well.

Note that this mobility approach does not require the Asterisk, the subscriber registry or SMQueue servers to have any prior knowledge of the BTS units. New BTS units can be added and removed without any modifications to the core network.

E.2 Example of Mobility Configuration, Simple Case

In this section, we will consider the simplest OpenBTS mobility implementation, where multiple BTS units share a common collection of central servers all running on the same physical machine. An example of such a network is shown in Figure E.1. In this configuration, there is a central server in a private IP network at 192.168.1.20 running central instances of Asterisk, the subscriber registry and SMQueue. There are three BTS units in the same subnet and the allowed ARFCNs are 40, 42, and 44. For simplicity, we show only the configuration parameters related to mobility.

![Diagram](image)

Figure E.1: A three-BTS network with a common server.

Common configuration, the same for all OpenBTS units:
E.2. EXAMPLE OF MOBILITY CONFIGURATION, SIMPLE CASE

- SIP.Proxy.Registration 192.168.1.20:5064 (the Subscriber Registry)
- SIP.Proxy.SMS 192.168.1.20:5063 (SMQueue)
- SIP.Local.Port 5062
- GSM.Radio.ARFCNs 1
- GSM.Identity.MCC 001
- GSM.Identity.MNC 01
- GSM.Identity.NCC 0
- GSM.CellSelection.NCCsPermitted 1 (matches GSM.Identity.NCC in bit 0)
- GSM.CellSelection.CELL-RESELECT-HYSTERESIS 3 (reselection threshold of 6 dB)

For unit A, on ARFCN 40 at IP 192.168.1.30:

- GSM.Radio.C0 40
- GSM.Neighbors 192.168.1.31 192.168.1.32
- GSM.Identity.BCC 0
- GSM.Identity.LAC 1000
- SIP.Local.IP 192.168.1.30

For unit B, on ARFCN 42 at IP 192.168.1.31:

- GSM.Radio.C0 42
- GSM.Neighbors 192.168.1.30 192.168.1.32
- GSM.Identity.BCC 1
- GSM.Identity.LAC 1001
- SIP.Local.IP 192.168.1.31

For unit C, on ARFCN 44 at IP 192.168.1.32:

- GSM.Radio.C0 44
- GSM.Neighbors 192.168.1.30 192.168.1.31
- GSM.Identity.BCC 2
- GSM.Identity.LAC 1002
• SIP.Local.IP 192.168.1.32

Note the following about the configurations:

• SIP.Proxy.* are the same on all units, pointing all of the BTS units to a common server.
• SIP.Local.IP is the IP address of each BTS as seen by the Asterisk server.
• GSM.Radio.C0 is different on each BTS.
• GSM.Neighbors lists all of the other BTS IP addresses on each BTS.
• GSM.Identity.NCC is the same on all units.
• GSM.Identity.BCC is different on all units.
• GSM.Identity.LAC is different on all units.
• GSM.CellSelection.NCCsPermitted is 1 because GSM.Identity.NCC is 0, so the NCC mask selects just the NCC for this network.
• GSM.CellSelection.CELL-RESELECT-HYSTERESIS is 3, giving a hysteresis of 6 dB, so whenever a neighboring cell is measured to be more than 6 dB stronger than the serving cell, the MS will recamp to the neighbor, which will trigger a registration in the Asterisk server at the new cell’s IP address.

E.3 Example of Mobility Configuration, More Advanced Case

The simple example in the previous section is easy to understand and easy to configure, but has some weaknesses:

• If a BTS unit loses connectivity to the core network due to a backhaul failure, that unit will cease to provide any services.
• All database, call routing and message routing loads are concentrated in the central servers, limiting scalability.

We can improve the performance and reliability of the network by using additional server sets inside the BTS units themselves or at each cell site, as shown in Figure E.2. The available ARFCNs are 40-45 and the IP addresses are in the 192.168.0.x range. The short summary of this configuration is that all of the OpenBTS units in each site point to the local server at that site (“S1” and “S2” in the figure). Servers S1 and S2 may be separate server machines in the cell sites with the BTS units or may use computational resources of one of the OpenBTS units, as shown in Figure 2.2.

For brevity, we will not show the entire configuration for all of the devices in this network, but the general rules and patterns for BTS configuration are:

• All BTS units have the same MCC, MNC and NCC. All units have the same NCCs-permitted mask, set to select the NCC that is used.
• All BTS units have different LAC, CI and BCC.
All BTS units have different ARFCNs and each BTS unit has a neighbor list that lists the ARFCNs of any nearby units.

BTS units in each site designate the site-local servers as their SIP proxies for all services:
- Units 1A, 1B and 1C list the servers in S1 as their SIP.Proxy.* parameters.
- Units 2A, 2B and 2C list the servers in S2 as their SIP.Proxy.* parameters.

The configuration rules for the servers are:
- The subscriber registry servers in S1 and S2 point to the subscriber registry server in CS as their upstream server.
- The Asterisk servers in S1 and S2 are configured to refer any call whose destination number cannot be resolved locally to the central Asterisk server in CS.
- The SMQueue servers in S1 and S2 are configured to refer any message whose destination number cannot be resolved locally to the central SMQueue server in CS.
- Any transaction for which the central servers in CS cannot resolve an address is referred to an external network or service.

E.4 Handover

In conventional GSM networks, handover is coordinated by a BSC or MSC that the two BTS units have in common. In OpenBTS networks, handover is coordinated by the BTS units themselves using the Range Peering Protocol (RPP). On the air interface, Um, the procedure looks like any other GSM handover. On the SIP
network interface, the handover operation looks as if a SIP endpoint changed its IP address in mid-call. The full transaction ladder is shown in Figure E.3, including RPP messaging.

OpenBTS currently supports handovers of telephone calls, but does not yet attempt handover during other circuit-switched operations (SMS, USSD, etc.). OpenBTS does support handover of emergency/SOS calls if the serving switch or PBX is compatible with the procedure, but OpenBTS does not update location information after the call is initiated.

![Diagram of OpenBTS handover process]

Figure E.3: GSM, RPP and SIP signaling for OpenBTS handover. In this example, a call that originated on BTS1 is handed over to BTS2. The call is between the MS and the “remote party”. Signaling between BTS units and MS is GSM. Signaling between BTS units and the remote party is SIP. Signaling between BTS1 and BTS2 is RPP.

### E.4.1 Handover Process

Handover is a complex process that affects every aspect and element of the OpenBTS software.
Neighbor Information Exchange

BTS units exchange configuration information directly using RPP. On each BTS unit, the GSM.Neighbors configuration parameter gives a list of IP addresses from which the BTS will request information. These requests are repeated at a configurable period, usually every 10 seconds, using the RPP REQ NEIGHBOR_PARAMS and RPP RSP NEIGHBOR_PARAMS messages.

Measurement Reports

The serving BTS unit is responsible for making the decision to initiate a handover. The decision is made based on “measurement reports” from the handset of other BTS units in the area.

During a call, there is a constant exchange of control information between the BTS unit and the handset. Most of this information concerns the receiver signal strength indications (“RSSI”) of neighboring BTS units so that the network can decide which BTS unit is best suited to serve the handset. The handset makes these measurements during the idle periods between active timeslots. (Since the call is carried on a single slot, the handset is idle at least 87.5% of the time during the call.)

- In the downlink direction, the BTS unit sends to the handset a list of neighboring BTS units to measure.
- In the uplink direction, the handset sends signal level information for those neighbors that it can measure.

This exchange occurs on the SACCH, roughly every 0.6 second during a call. Every time this exchange occurs, the BTS unit makes a decision as to whether or not to initiate a handover. The decision algorithm is:

1. If the RSSI of the serving BTS is greater than GSM.Handover.LocalRSSIMin (given in dBm) do not hand over; stop.
2. If the RSSI of the strongest neighbor exceeds the RSSI of the serving BTS by more than the value given in GSM.Handover.ThresholdDelta (in dB), initiate a handover to the strongest neighbor.

Initiating Handover

The actual handover transaction starts when the serving BTS (“BTS1”) sends the RPP REQ HANOVER message to the selected neighbor (“BTS2”). This message carries the entire call state, in the GSM, SIP and RTP domains. BTS2 responds with RPP RSP HANOVER. This response normally indicates acceptance of the inbound handover by BTS2 and includes a description of the radio channel to which the handset will be transferred. If BTS2 cannot accept the handover (usually due to congestion), this message will contain an error cause code and a hold-off time, during which additional handovers should not be requested from BTS2.

Moving to the New Radio Channel

At this point in the protocol, BTS2 has a copy of the complete call state and BTS1 has a description of the radio channel on BTS2 to which the call is to be transferred. BTS1 now sends the handset a GSM Handover

\[1\] Notice that in the idle mode it is the handset’s decision to move to a new cell, while during an active call the decision to move the handset to a new cell is made by the BTS unit.
APPENDIX E. MULTI-BTS NETWORKS

Command, containing a description of the radio channel on BTS2. Meanwhile, BTS2 creates a clone of the
call’s transaction record in its own transaction table.

Upon receiving the GSM Handover Command, the handset retunes to the new channel on BTS2 and begins
sending the Handover Access repeatedly (about 216 times per second) at full power and with zero timing
advance. BTS2 responds as quickly as possible with the GSM Physical Information message, which sets the
new timing advance and transmit power level for the handover. The handset responds with the GSM Handover
Complete message to verify that all is well.

Moving the Call to a New IP Address

Now that the handset is established on the radio channel, BTS2 moves the handset’s end of the call to the new
IP address with a SIP re-INVITE exchange.

Cleaning Up

Once BTS2 receives the GSM Handover Complete message from the handset, BTS2 sends the RPP IND
HANDOVER message to BTS1 to indicate that the handover was successful and that the BTS resources
originally used for the call can now be released. BTS1 responds with ACK HANDOVER to close the transaction.

E.4.2 PBX & Switch Support

The SIP signaling used by OpenBTS for handovers is known to work with Asterisk 10, FreeSWITCH 1.1 and
recent versions of yate. Handover will not work with Asterisk versions prior to 10. Handover has not been tested
with other SIP switch/PBX products.

E.4.3 Configuring Handover

The OpenBTS parameters related to handover and requiring configuration by the user are:

- GSM.Neighbors – This is the list of IP addresses of neighboring cells.
- GSM.Handover.ThresholdDelta – This is the power level difference (in dB) between the serving cell and
  the neighbor cell at which handover is initiated. In other words, when a neighbor cell’s RSSI level exceeds
  the serving cell’s RSSI level (as the handset) by this much, the serving BTS will initiate a handover to
  the neighbor.
- GSM.Handover.LocalRSSIMin – This is the power level above which handover will not be attempted.
  Regardless of the neighbor cell RSSI levels, if the serving cells’ RSSI is above this level, the BTS will not
  initiate the handover.

E.5 Remote Logging

In multi-BTS networks it is extremely useful to be able to selectively forward log events to a central monitoring
station. The rsyslog software used in most OpenBTS installations supports this function already; it just needs
E.5. REMOTE LOGGING

to be configured.

E.5.1 Configuring the Remote BTS Unit

To send log events to a remote host via UDP add this line to /etc/rsyslog.d/OpenBTS.conf:

    local7.<level> @<host>:<port>

To send to a remote host via TCP use

    local7.<level> @@<host>:<port>

For either protocol, the port parameter is optional and defaults to the standard port of 514. In either case, all log events at or numerically below the given logging level will be echoed to the given remote host over the specified protocol. The original configuration line writing to /var/log/OpenBTS.log will still be in effect, producing an additional local copy of the log on the BTS unit itself, possibly at a different logging level.

E.5.2 Configuring the Monitoring Station

At the remote host where the log events are to be received, there is a corresponding rsyslogd instance configured to listen on the given port. The network-reported will be further processed according to the remote system’s rsyslogd configuration. The configuration lines are:

    # For TCP on port 514:
    $InputServerRun 514

    # For UDP on port 514:
    $UDPServerRun 514

This monitoring station should also have a /etc/rsyslog.d/OpenBTS.conf file containing the line

    local7.* /var/log/OpenBTS.log

to route OpenBTS-related message to /var/log/OpenBTS.log, giving a unified log of all of the activity in the OpenBTS network.

E.5.3 Example

Using multi-BTS example network from Section E.2, we can configure the central server at 192.168.1.20 to also act as a central logging monitor over TCP/IP. In this example, the central facility will log events at or above the “CRIT” level.

On every BTS unit, add this line to /etc/rsyslog.d/OpenBTS.conf:

    local7.crit @@192.168.1.20:514
Then execute this line (on each unit) to restart rsysld:

```bash
sudo service rsyslog restart
```

On the central server at 192.168.1.20, add this line to `/etc/rsyslogd.conf`:

```bash
# For TCP on port 514:
$InputServerRun 514
```

Add this line to `/etc/rsyslog.d/OpenBTS.conf`:

```bash
local7.* /var/log/OpenBTS.log
```

Then execute this line to restart rsysld:

```bash
sudo service rsyslog restart
```

Once this configuration is in place, all OpenBTS-related log events at or above the “CRIT” level will also be logged in `/var/log/OpenBTS.log` on the central server at 192.168.1.20.

### E.5.4 Email Notifications

Rsysld can be configured to send notifications via email. For maximum reliability, this should be configured on each BTS unit.

To use this feature, the Unix sendmail utility should be installed on each unit. Sendmail can be installed (or updated) with the line

```bash
sudo apt-get install sendmail
```

Once sendmail is installed, rsyslog can be configured to send email notifications by adding lines like these to `/etc/rsyslog.conf`:

```bash
# Email notification
#$ModLoad ommail
$ActionMailSMTPServer 127.0.0.1
$ActionMailFrom openbts
$ActionMailTo support@mynoc.com
$template mailSubject, "alarm on \%hostname\%"
$template mailBody, "\%msg\%"
$ActionMailSubject mailSubject
# make sure we receive a mail only once every 15 minutes
$ActionExecOnlyOnceEveryInterval 900
# the if ... then ... mailBody must be on one line!
if ($syslogfacility-text == 'local7') and ($syslogseverity <= 1) then :ommail:;mailBody
```
In the example above,

- Sendmail is running locally, at 127.0.0.1.
- The source address of the mail is “openbts”.
- The destination address of the mail is “support@mynoc.com”.
- The if ... then line at the end of the example will filter messages from facility local7 (OpenBTS and its allies) with numeric severity values of 1 or lower (ALERT and EMERGENCY).
- Regardless of log activity, the BTS unit will send no more than one email message every 15 minutes.

The typical email delivery in this example is:

From: openbts
Subject: alarm on unit_24
Date: October 5, 2011 7:53:46 PM PDT
To: support@mynoc.com

ALERT 3079411408 OpenBTS.cpp:141::main:
OpenBTS starting, ver P2.8TRUNK build date Oct 5 2011
Appendix F

Test Procedures

The default configuration of a Range Networks factory-installed OpenBTS Release 4.0 system includes an Asterisk configuration to support testing of OpenBTS, Asterisk, the SIP authorization server (SIPAuthServe), Subscriber Registry and SMQueue as a system. In order to test the system, you need two SIM cards with known IMSI numbers and unlocked handsets that operate in the GSM band specified by the “GSM.Radio.Band” parameter of your OpenBTS system.

F.1 Test SIM Procedures

In order to perform these test procedures your SIM cards must be provisioned in your OpenBTS system. See section 6.2 on how to provision pre-existing SIMs. If you purchased the Range Networks 3-Phone Starter Kit, part #8810-000-0, and the Range Networks hardware at the same time, your system already has the SIM cards provisioned. Similarly, if you generated the SIMs using the Range Networks SIM Reader/Programmer connected to your system, they are already provisioned. Additional blank SIM cards are available from Range Networks, part #8430-999-0.

Extension/Phone Numbers Used for Testing

Each SIM card that will be used in the tests needs to be assigned an extension/phone number. In order to assign and/or update a phone number associated with a SIM, access the Subscriber Registry HTTP interface (described in section 6.1.2) using your web browser. Click the “[Link number]” action corresponding to an
IMSI of the SIM card, specify the number, and click on “Link”. Once entered, it should appear in the “Phone number” column of the table.

The loopback number defined in the default Asterisk configuration is 2600.

F.1.1 Tests with the Test SIM

The supported tests are:

- Location updating;
- MOC to a loopback, where the handset dials 2600;
- MOC to a handset, where the originating handset dials the number assigned to the terminating handset;
- Loopback SMS (MO-SMS and MT-SMS), where a handset sends SMS to its extension number and waits for the result;
- SMS with a parallel call, where a handset starts a loopback call to 2600, then sends SMS to another phone while still on a call.

Location Updating

The location updating step must be performed before any other test can proceed. Location updating verifies the operation of:

- the OpenBTS GSM stack in the BTS,
- the air interface between the handset and BTS,
- SIPAuthServe, and
- the SIP interface between OpenBTS and SIPAuthServe.

The procedure is:

- Power up the BTS unit. All of the required software will start automatically within two minutes.
- Turn on the handsets with the test SIMs.
- Within a few seconds, the handsets should show service at network MCC=001 MNC=01. If the handsets are set for manual network selection, choose network "001 01".
- In the OpenBTS CLI, the “tmsis” command (Section C.29) should show the IMSI numbers of the corresponding SIM cards with a “used” value of just a few seconds.
MOC to Loopback

MOC loopback verifies the signaling and media paths through OpenBTS and Asterisk.

- Be sure the handset’s SIM card is registered and the handset shows service as in the location updating test.
- From the handset, place a call to extension 2600.
- The call should connect quickly to an echo test. The echo latency will start large, around 1 second, but decrease quickly as the jitter buffers adapt.

MOC to Another Handset

The MOC-to-handset test exercises MOC signaling more completely than the MOC-to-loopback test.

- Be sure the phones are registered and show service as in the location updating test.
- From one of the handsets, place a call to the number assigned to another handset.
- The call should ring at the second handset and connect normally when answered.
- When the call is terminated, it should be a normal termination with no warnings.
- Try the test with each originating and terminating handset disconnect.

F.2 Testing with Open Registration

The open registration feature (Section 6.2.3) provides a mechanism for testing components of an OpenBTS installation in a standalone manner.

F.2.1 Interference Prevention

Because of the potential for interference to existing networks, this testing should be conducted only at low power levels and without mast-mounted antennas. The recommended procedure is to use a small omni antenna (Range part #8101-000 or similar) connected directly to the antenna connector (for multiband units, connect the antenna to the “RX” antenna connector). For BTS units with output power greater than 1 Watt (Range 5150 series -xx5 and -xx7 types), a high-power attenuator should be used (Range part #8435-207 or equivalent) between the BTS and the antenna.

F.2.2 Test Procedures

Unlike the test SIM procedures of the previous section, the open registration tests exercise the subscriber registry and the connections among the Subscriber Registry, Asterisk, SMQueue and OpenBTS.

All of these procedures are performed with open registration enabled, using a handset with a SIM card that is novel to the basestation being tested.
F.2. TESTING WITH OPEN REGISTRATION

Location Updating

This test exercises the Um link between the BTS and handset. This test must be conducted first to establish service on the handset for the other procedures.

1. In OpenBTS, set Control.LUR.OpenRegistration to ".*" to enable open registration. (Do not leave OpenBTS in this configuration for long periods, since it could disrupt local cellular service. This parameter is a regular expression used to match SIMs selectively, and ".*" means “accept any SIM”.)

2. In OpenBTS, define the Control.LUR.OpenRegistration.Message to some non-null string and set Control.LUR.OpenRegistration.ShortCode to 101.

3. Power up the handset within about 5 meters (15 feet) of the BTS unit test antenna.

4. The handset will “attach” to the BTS within a few seconds, perform a successful location update and show service.

5. The handset will receive a text message from 101 containing the open registration welcome message.

Autoprovisioning

The autoprovisioning test verifies correct interaction between OpenBTS, SMQueue and the Subscriber Registry.

1. Perform the location updating procedure described in the previous section.

2. Send a text message to 101 containing a 10 digit telephone number.

3. Within a minute, the handset should receive a text message verifying successful provisioning.

MTC and MOC with Newly Provisioned Handsets

The MTC and MOC tests verify correct interaction between OpenBTS, Asterisk and the subscriber registry.

1. Provision two handsets at two different telephone numbers using the procedure of the previous section.

2. From one handset, dial the telephone number of the other handset.

3. The call should connect normally.
Appendix G

FOSS Licenses

G.1 Creative Commons License

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

"Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work,
or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image (“synching”) will be considered an Adaptation for the purpose of this License.

“Collection” means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined below) for the purposes of this License.

“Creative Commons Compatible License” means a license that is listed at http://creativecommons.org/compatiblelicenses that has been approved by Creative Commons as being essentially equivalent to this License, including, at a minimum, because that license: (i) contains terms that have the same purpose, meaning and effect as the License Elements of this License; and, (ii) explicitly permits the relicensing of adaptations of works made available under that license under this License or a Creative Commons jurisdiction license with the same License Elements as this License.

“Distribute” means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

“License Elements” means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.

“Licensor” means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

“Original Author” means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

“Work” means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

“You” means an individual or entity exercising rights under this License who has not previously violated the
terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

"Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

"Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights.

Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant.

Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";

to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

to Distribute and Publicly Perform Adaptations.

For the avoidance of doubt:

Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.
4. Restrictions.

The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(c), as requested.

You may Distribute or Publicly Perform an Adaptation only under the terms of: (i) this License; (ii) a later version of this License with the same License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-ShareAlike 3.0 US)); (iv) a Creative Commons Compatible License. If you license the Adaptation under one of the licenses mentioned in (iv), you must comply with the terms of that license. If you license the Adaptation under the terms of any of the licenses mentioned in (i), (ii) or (iii) (the ”Applicable License”), you must comply with the terms of the Applicable License generally and the following provisions: (I) You must include a copy of, or the URI for, the Applicable License with every copy of each Adaptation You Distribute or Publicly Perform; (II) You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License; (III) You must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform; (IV) when You Distribute or Publicly Perform the Adaptation, You may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License.

If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor’s copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(c) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for
all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author’s honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author’s honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.


EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a
license to the Work on the same terms and conditions as the license granted to You under this License.

Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

G.2 GNU General Public License, v.1

GNU GENERAL PUBLIC LICENSE
Version 1, February 1989

Copyright (C) 1989 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA  02110-1301 USA

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

Preamble

The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. The General Public License applies to the Free Software Foundation's software and to any other program whose authors commit to using it. You can use it for your programs, too.
When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of a such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must tell them their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any
other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

3. You may copy and distribute the Program (or a portion or derivative of it, under Paragraph 2) in object code or executable form under the terms of Paragraphs 1 and 2 above provided that you also do one of the following:

a) accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Paragraphs 1 and 2 above; or,

b) accompany it with a written offer, valid for at least three years, to give any third party free (except for a nominal charge for the cost of distribution) a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Paragraphs 1 and 2 above; or,
c) accompany it with the information you received as to where the corresponding source code may be obtained. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making modifications to it. For an executable file, complete source code means all the source code for all modules it contains; but, as a special exception, it need not include source code for modules which are standard libraries that accompany the operating system on which the executable file runs, or for standard header files or definitions files that accompany that operating system.

4. You may not copy, modify, sublicense, distribute or transfer the Program except as expressly provided under this General Public License. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the Program is void, and will automatically terminate your rights to use the Program under this License. However, parties who have received copies, or rights to use copies, from you under this General Public License will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing or modifying the Program (or any work based on the Program) you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein.

7. The Free Software Foundation may publish revised and/or new versions of the General Public 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.

Each version is given a distinguishing version number. If the Program specifies a version number of the license which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the license, you may choose any version ever published by the Free Software Foundation.

8. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author
to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to humanity, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option)
any later version.

This program 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 GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (a program to direct compilers to make passes at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

That's all there is to it!

G.3 GNU General Public License, v.2

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any
part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public 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.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any
later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

This program 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
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate
parts of the General Public License. Of course, the commands you use may
be called something other than 'show w' and 'show c'; they could even be
mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into
proprietary programs. If your program is a subroutine library, you may
consider it more useful to permit linking proprietary applications with the
library. If this is what you want to do, use the GNU Lesser General
Public License instead of this License.
G.4 GNU General Public License, v.3

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 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.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and
authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for
infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.


The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for
the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.


No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work’s users, your or third parties’ legal rights to forbid circumvention of technological measures.


You may convey verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

   a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

   b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

   c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

   d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is
available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

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, you do not qualify to receive new licenses for the same material under section 10.


You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.


If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public 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.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS
How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program 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 GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program
into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

G.5 GNU Lesser General Public License, v.2.1

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.
For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain
special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)
"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

   c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

   d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.
(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer’s own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user’s computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.
It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public 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.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a
14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the
APPENDIX G. FOSS LICENSES

"copyright" line and a pointer to where the full notice is found.

<one line to give the library’s name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library 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 GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

G.6 GNU Lesser General Public License, v.3

GNU LESSER GENERAL PUBLIC LICENSE Version 3, 29 June 2007
Copyright (c) 2007 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.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions. As used herein, this License refers to version 3 of the GNU Lesser General Public License, and the GNU GPL refers to version 3 of the GNU General Public License.
The Library refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An Application is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A Combined Work is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the Linked Version.

The Minimal Corresponding Source for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The Corresponding Application Code for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL. You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions. If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files. The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License. b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works. You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License. b) Accompany the Combined Work with a copy of the GNU GPL and this license document. c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document. d) Do one of the following: 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source. 1) Use a suitable shared library mechanism for linking with the Library. A
suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version. e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries. You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License. b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License. The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public 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.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License or any later version applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy’s public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

G.7 BSD Licenses

G.7.1 BSD License on the A5/1 Implementation

Copyright (C) 1998-1999: Marc Briceno, Ian Goldberg, and David Wagner

The source code below is optimized for instructional value and clarity. Performance will be terrible, but that's not the point. The algorithm is written in the C programming language to avoid ambiguities inherent to the English language. Complain to the 9th Circuit of Appeals if you have a problem with that.

This software may be export-controlled by US law.
The software is free for commercial and non-commercial use as long as the following conditions are adhered to. Copyright remains the authors’ and as such any Copyright notices in the code are not to be removed. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

This software is provided ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

G.7.2 BSD License on Sparkline plug-in

Copyright (c) 2012, Splunk Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Splunk Inc nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
List of Figures

2.1 Components of the OpenBTS application suite and their communication channels as installed in each access point. Sharp-cornered boxes are hardware components. Round-cornered boxes are software components. .................................................. 17
2.2 Two access points with unit #1 providing servers for both. ................................. 18

6.1 GSM location update mapped to a SIP REGISTER (non-authenticating case)........ 45
6.2 GSM location update mapped to a SIP REGISTER (with authentication)............. 46
6.3 A GSM-SIP mobile-terminated call, VEA, normal case. ................................. 46
6.4 A GSM-SIP mobile-originated call, VEA, normal case. ................................. 47
6.5 Paired Asterisk servers for IAX trunking in satellite-based applications............. 49

A.1 Layers and channels of the Um interface. This figure shows the basic logical channel types in a subset of a typical configuration. ................................................................. 65
A.2 Mobile-terminated SMS transfer with no parallel call, normal case................. 75
A.3 Mobile-originated SMS transfer with no parallel call, normal case................. 76

E.1 A three-BTS network with a common server. ................................................. 120
E.2 Improved mobility architecture. For simplicity, only two cell sites are shown, each with three units. 123
E.3 GSM, RPP and SIP signaling for OpenBTS handover. In this example, a call that originated on BTS1 is handed over to BTS2. The call is between the MS and the “remote party”. Signaling between BTS units and MS is GSM. Signaling between BTS units and the remote party is SIP. Signaling between BTS1 and BTS2 is RPP. ................................................. 124
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Maximum output power levels for GSM MSs. From GSM 05.05 Section 4.1.1.</td>
<td>34</td>
</tr>
<tr>
<td>6.1</td>
<td>Backhaul bandwidth for various codec/trunking configurations. All rates in kbit/sec and assuming 20 ms framing.</td>
<td>48</td>
</tr>
<tr>
<td>A.1</td>
<td>Speech calling and service capacities of GSM systems.</td>
<td>72</td>
</tr>
<tr>
<td>A.2</td>
<td>Common GPRS multislot classes, giving maximum uplink and downlink multislot capabilities.</td>
<td>82</td>
</tr>
</tbody>
</table>