Running
Your own
GSM Network

by

Harald Welte <laforge@gnumonks.org>
Dieter Spaar <spaar@mirider.augusta.de>
Why would you run your own GSM network?

- For the same reason you might run other networks
  - To learn and experiment with technology
  - To boldly go where no [free] man has gone before ;)
- Practical demonstration of known GSM security problems
- Raise public awareness about GSM [in]security
  - thus increase the incentive for the market to improve
Legal Disclaimer

- Don't try this at home!
- GSM operates on LICENSED spectrum
  - Thus, you need approval from the regulatory authority
  - Only use BTS with dummy load!
  - Don't interfere with the operators!
- Our software is strictly for research purpose only
GSM Network Architecture

The Hitchhiker’s Guide to the GSM Network
- unfortunately does not exist

The GSM related literature
- is typically too high-level

The GSM protocol specifications
- are publicly available but _very_ comprehensive
  (1,108 PDFs, 414MByte)
GSM Network Architecture

GSM is a bit-synchronous network
- it draws many analogies from ISDN and SDN
- layer 2 modelled after Q.921 / LAPD
- call signalling modelled Q.931
- but: many more protocols for mobility management, radio resources, ...
- like all traditional Telco protocols: Intelligence in the network, not in the end nodes.

GSM is a TDMA "nightmare"
- e.g. you never know from/for whom data is without the timing context
GSM Network Architecture

**MS**
- Mobile Station (your Phone)

**BTS**
- Base Transceiver Station

**BSC**
- Base Station Controller

**MSC**
- Mobile Switching Center

**HLR/VLR**
- Home/Visitor Location Register
GSM Base Transceiver Station

BTS

- As the name indicates "transceiver"
- Handles
  - Layer 1 and some parts of RF layer 2
  - Modulation/Demodulation
  - Time Multiplex, scheduling of frames
- Is not a "Base Station", i.e. not self-contained
  - True 'slave' to the BSC
BSC

- Base Station Controller
- Handles
  - most of the actual decision making
  - really controls most aspects of BTSs
  - handles intra-BSC cell handover
MSC
- Mobile Switching Center
- Handles
  - Actual switching of the calls
  - Interworking with ISDN or POTS
  - Inter-BSC cell handover

HLR/VLR
- Home/Visitor Location Register
- Handles
  - database of local / roaming subscribers
GSM A-bis interface

BSC <-> BTS Interface

- is called A-bis
- has the following control layers on E1 TS1
  - L2ML (Layer 2 Management)
    - TEI management similar to ISDN
  - OML (Organization & Maintenance)
    - System parameters, events
  - RSL (Radio Subsystem Layer)
- has encoded voice data (TRAU frames) on other E1 TS
ISDN PRI (Primary Rate Interface), European variant (E1)

Frame: 125 μs (8 kHz), 256 bits

8 bits per timeslot

32 bidirectional channels transferred using TDM (time-division multiplexing)

two wire pairs, one for each direction

ternary signal (three levels) on the wire, HDB3 (high density bipolar of order 3) coding

64 kbit/s on each channel (32 * 64 kbit/s = 2048 kbit/s total)

timeslot 0 is used for synchronization and error handling (CRC)

timeslot 16 usually used for signalling (D-Channel), the others for traffic (B-Channels)
Speech/Data traffic

A timeslot is subdivided into four sub-channels with 16 kbit/s each.

Division is done by using 2 bits of a byte for each of the four sub-channels.

"Transparent" mode, no HDLC.

In GSM speech is compressed, several options (e.g. Full Rate or Half Rate).

Full Rate (GSM 06.10) compresses speech into blocks of 260 bits every 20 ms (13 kbit/s).

Source code for a Full-Rate codec at http://kbs.cs.tu-berlin.de/~jutta/toast.html

Each block of 260 bits is packed into a TRAU (Transcoder and Rate Adaptation Unit) frame of 320 bits by the BTS (Base Transceiver Station), total 50 * 320 bits/s = 16 kbit/s.

TRAU frames are specified by GSM 08.20.
GSM A-bis interface

- Abis RSL contains messages for
  - Radio Link Layer (RLL)
  - Dedicated Channel (DCHAN)
  - Common Channel (CCHAN)
  - Transceiver (TRX)
GSM Mobile Switching Center

Abis RSL Radio Link Layer

- contains messages for
  - Call Control (CC)
  - Mobility Management (MM)
  - Radio Resource (RR)
  - Short Message Service (SMS)

- mostly specified in GSM TS 04.08
Siemens BS-11 microBTS

- plain old 2G (GSM voice calls, CSD)
- one or two TRX, 30mW to 2W each, GSM900
- two E1 interfaces (for daisy-chaining)
- documentation under NDA, but
  - 99.9% of the A-bis protocol available from GSM specs
    - See TS 04.08 (RLL), 12.21 (OML), 08.58 (RSI)
- RS232 serial port for Local Maintenance Terminal
  - LMT software proprietary under NDA
    - not needed for operation of the BTS
The Siemens BS-11 microBTS
Running Your Own GSM Network

The Siemens BS-11 microBTS
The Siemens BS-11 microBTS
The Siemens BS-11 microBTS
First steps with the Siemens BS-11

- Harald bought a BS-11 on e-Bay in 2006
  - Started to read some specs (08.5x) about A-bis
  - Started to build cables for E1 and power
  - Bought HFC-E1 PCI card
  - Bought Elmi EGM35 Abis analyzer (e-Bay once again)
  - Contacted with other people who also bought BS-11
  - Found somebody who could provide Abis traces
  - Never really had time due to Openmoko and other projects
Further steps with the Siemens BS-11

- Dieter bought a BS-11 09/2008
  - Bought HFC-E1 PCI card
  - Started development based on HFC-E1 reference driver code
  - Found somebody who could provide Abis traces
  - Made very quick progress
BS11-Init (09/2008)
- Chip cologne HFC-E1 reference code for DOS
  - polling, no interrupts
- ported to Windows and Linux (mmap of HFC registers to userspace)
- proof-of-concept code based on challenge-response
- handles TEI assignment, brings OML and RSL up
- allows for location update and paging of single phone
BS11-Init
From BS11-Init to OpenBSC (12/2008)

- get L2ML to work with mISDN
  - mainline mISDN doesn't deal with multiple SAPIs and fixed TEI
- learn how new sockets-based mISDN API works
- come up with event-driven architecture, single select loop, no threads, ...

At 25C3:
- add libdbi/sqlite database for "HLR"
- get paging to work, support for configurable network ID
- debugging + stabilization with > 1000 test users ;)
- IMSI + IMEI skimming
Work at 25C3

IMSI+IMEI skimming

- very simple:
  - phones with automatic network selection pick strongest network
  - they send LOCATION UPDATE REQUEST
  - we send IDENTITY REQUEST IMSI + IMEISV
  - they send IMSI + IMEISV
  - we store this in the database
  - and then send LOCATION UPDATE REJECT
Mobile Originated Call

- once a MS is registered, we can
  - dial a number from the MS
  - allocate and establish a TCH/F
  - deal with the Signalling and get into Connect

- unfortunately, code for handling voice streams not finished
Work at 25C3

Mobile Originated SMS
- once a MS is registered, we can
  - send a SMS
  - parse + acknowledge SMS PDU data
Work at 25C3

The Egypt simulation

- apparently GPS is illegal in mobile phones in Egypt
  - "Egypt detection" implemented by checking if any surrounding cells are with Egypt country code
  - phones don't even have to register to our BTS!
  - so if we claim to be e.g. MobiNil, phones will shut off their GPS
Other GSM related FOSS

- OpenBTS
  - 100% Software Defined Radio based on USRP + gnuradio
  - implements entire RF+layer1/2/3 and interfacing to SIP/Asterisk
  - much more than just a BTS!!
  - some code overlap with OpenBSC
Links

- OpenBSC
  - http://openbsc.gnumonks.org/
- 3GPP / ETSI GSM Specs
  - http://www.3gpp.org/
- Priv-Doz. Dr.-Ing Joachim Goeller
  - http://www2.informatik.hu-berlin.de/~goeller
- THC GSM Wiki
  - http://wiki.thc.org/gsm
- OpenBTS
  - http://gnuradio.org/trac/wiki/OpenBTS
- Harald's branch of gsm-tvoid, etc
  - git://git.gnumonks.org/gsm.git
Thanks to

- zecke, alaphone, Stefan for their work on OpenBSC
- W. for his extensive A-bis protocol traces and MA-10
- all the voluntary testers at 25C3
- Karsten Keil for mISDN
Thanks

LIVE DEMO