osmocom.org - FOSS for mobile comms

community based Free / Open Source Software for communications

Harald Welte <laforge@gnumonks.org>

gnumonks.org
hmw-consulting.de
sysmocom GmbH

June 16, 2014, DORS/CLUC, Zagreb

Outline

1. Researching communications systems
2. The Osmocom project
3. Non-osmocom projects

About the speaker

- Using + toying with Linux since 1994
- Kernel / bootloader / driver / firmware development since 1999
- IT security expert, focus on network protocol security
- Former core developer of Linux packet filter netfilter/iptables
- Board-level Electrical Engineering
- Always looking for interesting protocols (RFID, DECT, GSM)
- OpenEXZ, OpenPCD, Openmoko, OpenBSC, OsmocomBB, OsmoSGSN

What this talk is about

- Implementing GSM/GPRS/3G network elements as FOSS
- Applied Protocol Archaeology
- Doing all of that on top of Linux (in userspace)
- From two nerds with a BTS off e-bay to a community project, several companies and real-world deployments around the globe

Research in TCP/IP/Ethernet

Assume you want to do some research in the TCP/IP/Ethernet communications area,

- you use off-the-shelf hardware (x86, Ethernet card)
- you start with the Linux / *BSD stack
- you add the instrumentation you need
- you make your proposed modifications
- you do some testing
- you write your paper and publish the results

Research in (mobile) communications

Assume it is 2008 (before Osmocom) and you want to do some research in mobile comms

- there is no FOSS implementation of any of the protocols or functional entities
- almost no university has a test lab with the required equipment. And if they do, it is black boxes that you cannot modify according to your research requirements
- you turn away at that point, or you cannot work on really exciting stuff
- only chance is to partner with commercial company, who puts you under NDAs and who wants to profit from your research

Running small (mobile) networks

Assume it is 2008 (before Osmocom) and you want to run a small cellular network for research, education, testing. You

- go to Ericsson/Huawei/ZTE/Nokia/Alcatel/...
- spend lots of time convincing them that youâĂŹre an eligible customer
- spend a six-digit figure for even the most basic full network
- end up with black boxes that you can neither study or improve
    - WTF?
    - I used FOSS protocol stacks for the Internet since 1994 and hacked on them since 1999. I knew a better world.

GSM/3G vs. Internet

- Observation
  - Both GSM/3G and TCP/IP protocol specs are publicly available
  - The Internet protocol stack (Ethernet/Wifi/TCP/IP) receives lots of scrutiny
  - GSM networks are as widely deployed as the Internet
  - Yet, GSM/3G protocols receive no such scrutiny!
- There are reasons for that:
  - GSM industry is extremely closed (and closed-minded)
  - Only about 4 closed-source protocol stack implementations
  - GSM chipset makers never release any hardware documentation

GSM is more than phone calls

Listening to phone calls is boring...

- Machine-to-Machine (M2M) communication
  - BMW can unlock/open your car via GSM
  - Alarm systems often report via GSM
  - Smart Metering (Utility companies)
  - GSM-R / European Train Control System
  - Vending machines report that their cash box is full
  - Control if wind-mills supply power into the grid
  - Transaction numbers for electronic banking

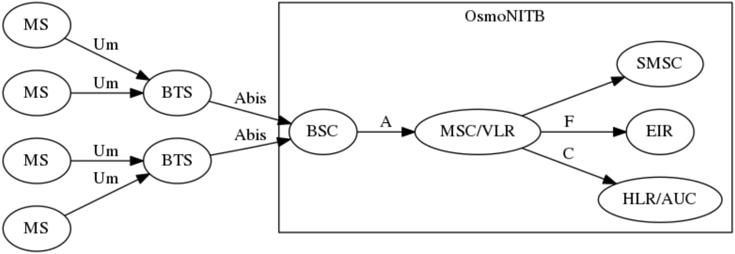
Enter Osmocom

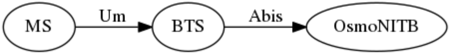
In 2008, two crazy Germans (Dieter Spaar + yours truly) started to write FOSS for GSM.

- to boldly go where no FOSS hacker has gone before
- where protocol stacks are deep
- and acronyms are plentiful
- we went from `bs11-abis` to `bsc_hack` to OpenBSC to OsmoNITB + OsmoBSC
- many other projects were created
- finally leading to the *Osmocom* umbrella project

Siemens BS-11 via ebay

Simplifying the GSM Network





Osmocom / osmocom.org

- Osmocom == Open Soruce Mobile Communications
- Classic collaborative, community-driven FOSS project
- Gathers creative people who want to explore this industry-dominated closed mobile communications world
- communication via mailing lists, IRC
- soure code in git, information in trac/wiki
- http://osmocom.org/

OpenBSC

- first Osmocom project
- Implements GSM A-bis interface towards BTS
- Primarily supports sysmoBTS and ip.access nanoBTS
- Limited support for some Siemens, Ericsson and Nokia BTS models
- can implement only BSC function (osmo-bsc) or a fully autonomous self-contained GSM network (osmo-nitb) that requires no external MSC/VLR/AUC/HLR/EIR
- deployed in (at least) > 300 installations world-wide, commercial and research

First OpenBSC test installation (HAR 2009)

Osmocom Cellular Network use cases

- can be used either as pure BSC (A-over-IP)
  - suitable for operators with existing core (MSC/VLR/HLR/AUC)
  - easy integration into existing infrastructure
- or together with OsmoMSC, OsmoHLR to form a Network In The Box
  - suitable for private / autonomous small networks (PBX style)
  - no dependency on any other external component
  - connect to the outside via ISDN or VoIP (using linux call router, osmo-sip-connector)
  - off-shore drilling rigs, underground mining, alternative to PMR

OsmoPCU / OsmoSGSN / OsmoGGSN

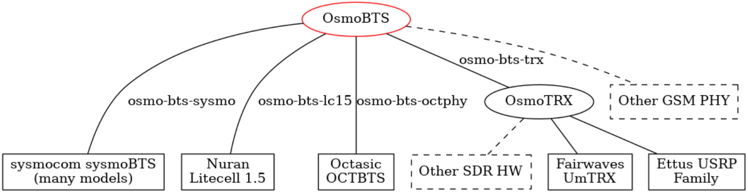
- extends the Osmocom based network from GSM to GPRS/EDGE by implementing the classic PCU, SGSN and GGSN functional entities
- OsmoGGSN based on pre-existing OpenGGSN code that was abandoned by original author
- Works only with BTSs that provides Gb interface, like sysmoBTS or nanoBTS
- Suitable for research only, not production ready

OsmoSGSN / OsmoGGSN use cases

- Testing of M2M devices using your own BTS+SGSN+GGSN
- Mobile malware research (analyze cellular data traffic of apps)
- Any type of GPRS related research
- Teaching, training on mobile data protocols/interfaces (RLC, MAC, LLC, SNDCP, BSSGP, NS, GTP, etc.)
- 3G / 3.5G support since 2016 by means of IuPS interface

OsmoBTS

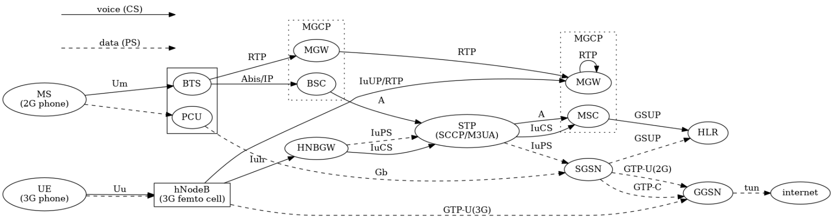
- OpenBSC/OsmoNITB takes care of BTS and higher elements
- OsmoBTS implements a BTS with A-bis/IP back-haul to OpenBSC
- Developed primarily for sysmoBTS hardware
- Ported to various other hardware, even by some BTS vendors!



Osmocom 3G

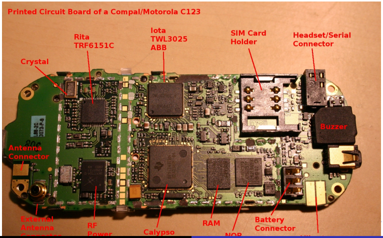
- OsmoBTS,PCU,BSC,MSC,HLR,SGSN,GGSN developed for 2G/2.5G/2.75G
- in 2015/2016, we added 3G/3.5G support
- OsmoMSC got IuCS interface
- OsmoSGSN got IuPS interface
- OsmoHLR got support for 3G mutual authentication
- OsmoHNBGW for talking Iuh to femtocells

Osmocom Cellular Network in 2017



OsmocomBB

- Full baseband processor firmware implementation of a mobile phone (MS)
- We re-use existing phone hardware and re-wrote the L1, L2, L3 and higher level logic
- Higher layers reuse code from OpenBSC wherever possible
- Used in a number of universities and other research contexts



OsmocomBB use cases

- Applied security research on Infrastructure
  - Fuzzing / exploiting of protocol parsers on network side
  - RACH denial of service
  - Check if networks use random padding
  - Detect IMSI catchers or other fals base stations
  - Assess GSM network (operator) security level
- Study + learn how a GSM stack / phone work
- Protocol tracing of your own transactions with the network

OsmocomTETRA

- SDR implementation of a TETRA radio-modem (PHY/MAC)
- Rx is fully implemented, Tx only partial
- Can be used for air interface interception
- Accompanied by wireshark dissectors for the TETRA protocol stack

OsmocomTETRA use cases

- Analysis/assessment of TETRA network security
- Learn how TETRA works on teh lowest levels (L1, MAC, L3)
- Protocol analysis / sniffing / intercepting unencrypted networks

OsmocomGMR

- ETSI GMR (Geo Mobile Radio) is "GSM for satellites"
- GMR-1 used by Thuraya satellite network
- OsmocomGMR implements SDR based radiomodem + PHY/MAC (Rx)
- Partial wireshark dissectors for the protocol stack
- Reverse engineered implementation of GMR-A5 crypto
- Speech codec is proprietary, still needs reverse engineering

OsmocomGMR use cases

- Analysis/assessment of GMR/Thuraya security (there is none)
- Learn and understnad how satellite telephony L1 and protocol work
- Actual interception of SMS + data
- Voice still difficult due to proprietary undocumented codec

OsmocomDECT

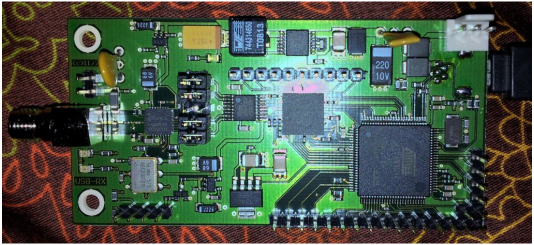
- ETSI DECT (Digital European Cordless Telephony) is used in millions of cordless phones
- deDECTed.org project started with open source protocol analyzers and demonstrated many vulnerabilities
- OsmocomDECT is an implementation of the DECT hardware drivers and protocols for the Linux kernel
- Integrates with Asterisk

OsmocomOP25

- APCO25 is Professional PMR system used in the US
- Can be compared to TETRA in Europe
- OsmocomOP25 is again SDR receiver + protocol analyzer
- Use cases like OsmocomTETRA

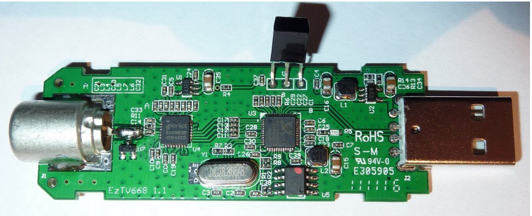
OsmoSDR

- small, low-power / low-cost USB SDR hardware
- higher bandwidth than FunCubeDonglePro
- much lower cost than USRP
- Open Hardware
- Developer units available



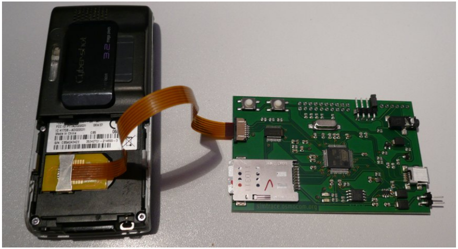
rtl-sdr

- re-purpose a USD 20 DVB-T USB dongle based on Realtek chipset
- deactivate/bypass DVB-T demodulator / MPEG decoder
- pass baseband samples via high-speed USB into PC
- no open hardware, but Free Software



OsmocomSIMTRACE

- Hardware protocol tracer for SIM - phone interface
- Wireshark protocol dissector for SIM-ME protocol (TS 11.11)
- Can be used for SIM Application development / analysis
- Also capable of SIM card emulation and man-in-the-middle attacks



osmo_ss7, osmo_map, signerl

- Erlang-language SS7 implementation (MTP3, SCCP, TCAP, MAP)
- SIGTRAN variants (M2PA, M2UA, M3UA and SUA)
- Enables us to interface with GSM/UMTS inter-operator core network
- Already used in production in some really nasty special-purpose protocol translators (think of NAT for SS7)

osmo_ss7, osmo_map, signerl use cases

- Implement GSM/3G core network elements (HLR, SCF, etc.)
- Applications that interact with GSM/3G core network elements
- Mostly useful for small MVNOs or other operators who have requirements that cannot be fulfilled with off-the-shelf proprietary equipment.

More Osmocom projects

- Have a look at http://git.osmcoom.org/
- 100 public git repositories / projects at this point
- way too many to cover here in this talk
- Often RTFS, no manual/docs

The OpenBTS Um - SIP bridge

- OpenBTS is a SDR implementation of GSM Um radio interface
- directly bridges to SIP/RTP, no A-bis/BSC/A/MSC
- suitable for research on air interface, but very different from traditional GSM networks
- work is being done to make it interoperable with OpenBSC

airprobe.org

- SDR implementation of Um sniffer
- suitable for receiving GSM Um downlink and uplink
- predates all of the other projects
- more or less abandoned at this point

xgoldmon

- extract all GSM/GPRS and even 3G protocol messages from your Samsung Galaxy 2, Galaxy 3, Note 2, Nexus phone via USB
- feed them into your PC running xgoldmon
- forward them from xgoldmon via GSMTAP into wireshark
- https://github.com/2b-as/xgoldmon

sysmocom GmbH

systems for mobile communications

- small company, started by two Osmocom developers in Berlin
- provides commercial R&d and support for professional users of Osmocom software
- develops + sells products like sysmoBTS (inexpensive, small-form-factor, OpenBSC compatible BTS)
- runs a small webshop for Osmocom related hardware items like SIMtrace

Where do we go from here?

- Now that we have GSM, GPRS, EGPRS, UMTS: LTE, of course
- Re-using femtocells in creative ways
- Proprietary PMR systems

Call for contributions

- Don't you agree that classic Internet/TCP/IP is boring and has been researched to death?
- There are many more communications systems out there
- Never trust the industry, they only care about selling their stuff
- Lets democratize access to those communication systems
- Become a contributor or developer today!
- Join our mailing lists, use/improve our code
- for OsmocomBB you only need a EUR 20 phone to start

Thanks

I'd like to thank the many Osmocom developers and contributors, especially

- Dieter Spaar
- Holger Freyther
- Andreas Eversberg
- Sylvain Munaut
- Neels Hofmeyr

Also, thanks to CEPT for permitting the GSM specs to be written in English (not French, the official Language of the international postal system)

Thanks

Thanks for your attention. I hope we have time for Q&A.

EOF.