GSM Workout

Improving GSM protocol analysis

Harald Welte

gnumonks.org
gpl-violations.org
OpenBSC
airprobe.org
hmw-consulting.de

FOSS.in conference, December 2009, Bangalore/India

Outline

The FOSS.in/2009 GSM workout

What do we want to achieve?

- improve airprobe.org GSM protocol analyzer
- improve wireshark protocol dissectors for GSM

The FOSS.in/2009 GSM workout

What skills do you need?

- general underestanding about communications protocols
- wireshark usage and preferrably wireshark dissector architecture
- GSM protocol knowledge not really required

airprobe architecture

- Software to receive GSM off the air
  - implements GSM layer 0 and 1, sometimes 2
  - many implementations available in airprobe.org
  - gsm-receiver and gsm-tvoid most popular
- Intermediate data formate to pass information to protocol analyzer
- Actual protocol analyzers like
  - gsmdecode, part of airprobe
  - wireshark.org project

Intermediate data formats

- Intermediate data formate to pass information between GSM receiver and actual protocol analyzer
  - hex bytes for every layer 2 or layer 3 message, or
  - PCAP file with GSM encapsulation type, or
  - some non-standard frames through tun/tap device, or
  - GSMTAP header (like wiretap) inside UDP packets over loopback device

Understanding Um

Overview

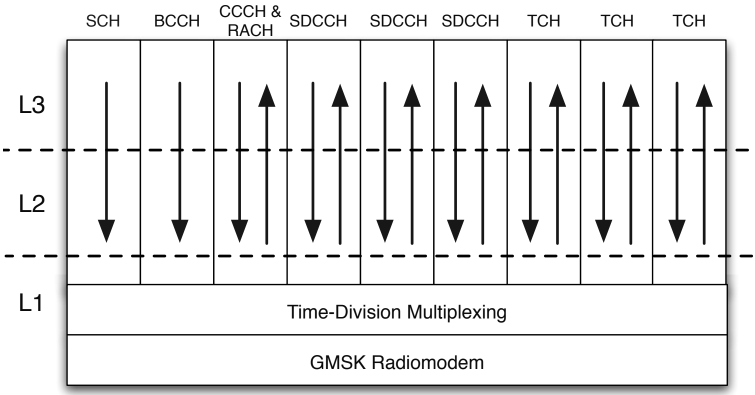
- Modeled after the U interface of ISDN
- Broadcast channels: SCH, BCCH, FCCH
- Common channels: CCCH (PCH & AGCH), RACH
- Dedicated Channels:

  Dm SDCCH, FACCH, SACCH

  Bm TCH/H, TCH/F

Understanding Um

Channels & Layers



Understanding Um

TDM Structure

- ARFCN (Absolute Radio Freq. Chan. Num.)– A 270,833 Hz radio channel. ARFCNs within a BTS numbered C0, C1, etc.
- 8 timeslots per frame on each ARFCN, numbered T0..T7.
- "physical channel" – one slot on one ARFCN, designated C0T0, C0T1, C1T5, etc.
- Physical channel TDM follows a 26- or 52-frame multiframe, carrying multiple logical channels.

Understanding Um –TDM Example

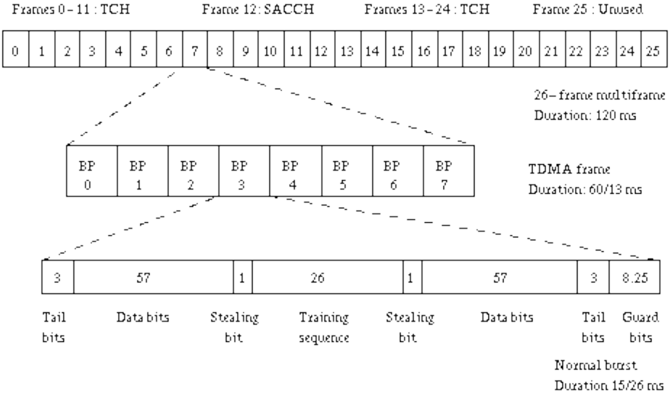


Figure: Example of traffic channel TDM

Understanding Um

The Beacon

The beacon is always on C0T0 and always constant full power

- **SCH** (Sync.) – TDM timing and reduced BTS identity
- **FCCH** (Freq. Corr.) – Fine frequency synchronization
- **BCCH** (Broadcast Control) – Cell configuration and neighbor list
- **CCCH** (Common Control) – a set of unicast channels
  - **PCH** paging channel for network-originated transactions
  - **AGCH** access grant channel
  - **RACH** uplink access request

Understanding Um

SCH – Synchronization CHannel

- First channel acquired by a handset
- T1, T2, T3' – TDM clocks for GSM frame number
- BCC – 3 bits, identifies BTS in the local group
- NCC – 3 bits, identifies network within a region
- BSIC is NCC:BCC

Understanding Um

BCCH – Broadcast Control CHannel

- Second channel acquired by the handset.
- A repeating cycle of system information messages.

Type 1	ARFCN set
Type 2	Neighbor list
Type 3	Cell/Network identity, CCCH configuration
Type 4	Network identity, cell selection parameters
GPRS	adds a few more (7, 9, 13, 16, 17)

Understanding Um

CCCH – Common Control CHannel

PCH Paging

- Unicast. Handsets addressed by IMSI or TMSI, never IMEI.
- Handset sees paging request and then requests service on RACH.

RACH Random Access

- Handset requests channel with RACH burst, 8-bit tag.

AGCH Access Grant

- BTS answers on AGCH, echoing tag and timestamp.

Understanding Um

Dm Channels

SDCCH Most heavily used control channel: registration, SMS transfers, call setup in many networks. Payload rate of 0.8 kb/s.

FACCH Blank and burst channel steals bandwidth from traffic. Used for in-call signaling, call setup in some networks. Payload rate up to 9.2 kb/s on TCH/F.

SACCH Low rate channel muxed onto every other logical channel type. Used for timing/power control, measurement reports and in-call SMS transfers.

Frequency Hopping

- Intended to improve radio performance through diversity in fading and interference
- Two ways to implement hopping
  - Baseband hopping: *N* fixed-frequency transceivers are connected to *N* baseband processors through a switch or commutator. Allows CA of *N* ARFCNs. C0 can be in the CA.
  - Synthesizer hopping: Each of *N* baseband processors connects to a dedicated transceiver. This requires transceivers that can be retuned and settled in less than 30 μs. Allows CA to have $\gg$ *N* ARFNCs. C0 is not in the CA.
- Some networks implement synchronous hopping to prevent collisions of hopping bursts from neighboring cells.

Frequency Hopping Parameters

A *hopping sequence* is an ordered list of ARFCNs used by a given physical channel (PCH), synced to the GSM frame clock. Each PCH can have an independent hopping sequence.

CA Cell Allocation, set of ARFCNs used for hopping in BTS

HSN Hopping Sequence Number, parameter used in pseudorandom algorithm generating hopping sequence

MA Mobile Allocation, subset of CA used by a particular PCH

MAIO MA Index Offset, offset added to hopping sequence when indexing MA.

- CA is the same for every PCH in the BTS
- HSN, MA and MAIO can be different for every PCH, usually only MAIO is unique

Understanding Um

The Layers

The Layers are not exactly the ISO model, but a similar theme.

- **L1** The radiomodem, TDM and FEC functions
- **L2** Frame segmentation and retransmission
- **L3** Connection & mobility management
- **L4** Relay functions between BSC and other entities

Understanding Um
The Layers

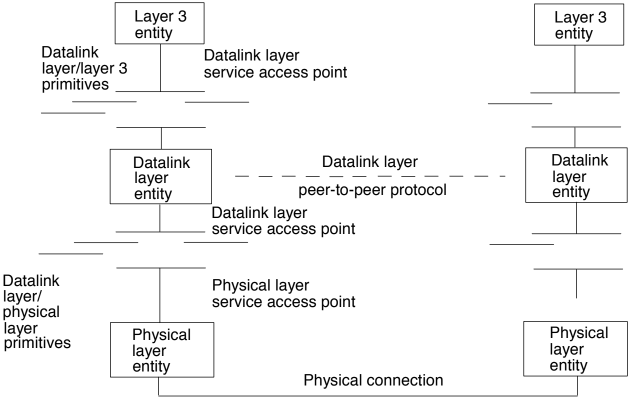


Figure: Layers of a Dm channel

Understanding Um

L1

- Analog radio path (transceiver, amplifiers, duplexer, antenna)
- GMSK or GMSK/EDGE radiomodem (“L0”)
- TDM to define logical channels
- FEC (Forward Error Correction)
  - Rate-1/2 convolutional code is typical.
  - 40-bit Fire code parity word on most control channels.
  - 4-burst or 8-burst interleaving is typical.

L1 Overview (see handout)

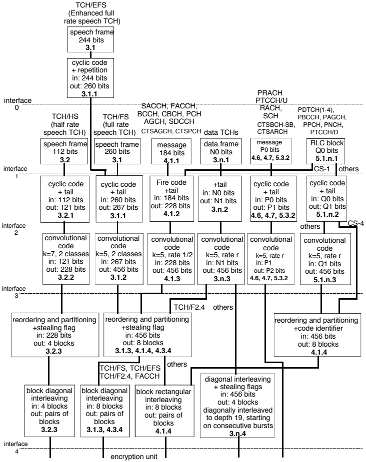


Figure 1a: Channel Coding and Interleaving Organization

Um L1 Interleaving

- Every GSM data frame is spread over 4 or 8 radio bursts.
  - 4-burst block interleave on most channels
  - 8-burst diagonal interleave on TCHs
- Loss of one burst means 1/4 or 1/8 missing channel bits, scattered throughout a frame.
- Allows a slow-hopping system to achieve many performance gains associated with fast-hopping.

Understanding Um

L2

- L1 drops frames, but L3 assumes a reliable link.
- L1 uses fixed-length frames, but L3 uses variable-length messages.
- L2 (Data Link Layer) bridges the gap with segmentation, sequencing and retransmission.
- ISDN uses LAPD for L2, derived from HDLC, derived from SDLC, dating back to IBM's SNA mainframe networks.

Understanding Um

L2

- LAPDm on Dm channels, a HDLC derivative, similar to ISDN's LAPD but simplified.
- LLC on GPRS channels, another HDLC derivative.
- GSM defines no L2 in Bm channels.
  - Speech/fax are just media and have no L2.
  - CSD typically used with PPP for L2.

GSMTAP Interface

It's important to find the right level of the GSMTAP interface

- If we simply pass every GSM burst, then wireshark would need to do the burst-rerassembly, forward error correction, etc - something it traditionally doesn't do
- If we pass every Layer 2 Frame (23 bytes)
  - burst decoding, reassembly, etc. is done in receiver
  - however, every burst might have different RF parameters like ARFCN, RX level, error rate, ...

Current GSMTAP Header

```
struct gsmtap_hdr {
        u_int8_t version;          /* version, set to 0x01 currently */
        u_int8_t hdr_len;          /* length in number of 32bit words */
        u_int8_t type;             /* see GSMTAP_TYPE_* */
        u_int8_t timeslot;         /* timeslot (0..7 on Um) */

        u_int16_t arfcn;           /* ARFCN (frequency) */
        u_int8_t noise_db;         /* noise figure in dB */
        u_int8_t signal_db;        /* signal level in dB */

        u_int32_t frame_number;    /* GSM Frame Number (FN) */

        u_int8_t burst_type;       /* Type of burst, see above */
        u_int8_t antenna_nr;       /* Antenna Number */
        u_int16_t res;             /* reserved for future use (RFU) */

} __attribute__((packed));
```

ip.access wireshark dissectors

- ip.access wrote some wireshark dissectors against an old wireshark version
- they never submtited them upstream, but we have the source under GPL
- meanwhile, upstream wireshark has parts of that functionality
- we now need to port those old dissectors to current wireshark

ip.access wireshark dissectors

- IPA protocol as encapsulation layer
  - different implementation in upstream (packet-gsm_ipa.c)
  - maybe some few bits missing from upstream
  - port the missing bits from ip.access to upstream
- GSM 12.21 (A-bis OML)
  - different implementation in openbsc (abis-oml.patch)
  - quite a number of bits missing from upstream
  - BTS vendor specific decoding preference needed
- GSM 08.58 (A-bis RSL)
  - different implementation in upstream (packet-rsl.c)
  - many ip.access specific bits missing
  - port the missing bits from ip.access to upstream

ip.access wireshark dissectors

- IPA IML (internal management link)
  - no implementation in upstream
  - simply merge it into current upstream
- RTP Multiplex (packet-rtp_mux.c)
  - no implementation in upstream
  - simply merge it into current upstream
- GSM CSD (packet-gsm_csd.c)
  - no implementation in upstream
  - simply merge it into current upstream