osmocom.org - Core Network Protocols

Harald Welte <laforge@gnumonks.org>

gnumonks.org
hmw-consulting.de
sysmocom GmbH

March 2012, OsmoDevCon 2012, Berlin / Germany

Outline

GSM core network components

MSC (Mobile Switching Center): The central switch

HLR (Home Location Register): Database of subscribers

AUC (Authentication Center): Database of authentication keys

VLR (Visitor Location Register): For roaming users

EIR (Equipment Identity Register): To block stolen phones

GSM network structure

MSC Actual call switching and top-level mobility functions. May serve dozens of location areas

VLR Temporary cache of subscriber data from HLR + TMSI

HLR Subscriber databases + subscriber location information

AUC Generation of authentication tuples

SMSC SMS Service Centre, store+forward for SMS

GSM core network integration

- VLR often integrated into MSC
- AUC often integrated with AUC
- integration so common, many graphs/diagrams are actually not 100% correct

GSM network interfaces

- C Interface between GMSC and HLR
- D Interface between MSC and HLR
- E Interface between MSC and MSC

All of them based on MAP, so C/D/E not commonly distinguished

core network protocol stack

Traditional telephony based on SS7 / CS7, GSM too

- Lower layers (MTP2/MTP3) re-used
- ISUP used for actual call control signalling
- SCCP for routing / GTT
- TCAP for transaction supprt
- MAP for actual GSM related signalling

SS7 networks

- STP - Signalling Transfer Point
  - *Router* for SCCP
  - performs GTT (see below)
- SCP - Signalling Control Point
  - *End-node* like MSC/HLR
  - SCP has GT, PC, ..

SS7 addresses

- Point Code (PC)
  - typically unique within PLMN / country
- Global Title (GT)
  - world-wide unique address
  - translated into PC by GTT at STP
- Subsystem Number (SSN)
  - logical function address inside network (MSC, VLR, HLR, ...)
  - not used on international links

SS7 GTT (Global Title Translation)

Global Title Translation

- can happen at any STP
- translates a Destination GT into new destination address
- new dest address can be any address, such as
  - new global title (GT)
  - point code (PC)
  - sub-system number (SSN)
- GTT rules explicitly configured by operator, e.g.
  - prefix or range based match
  - (inter)nationalize numbering plan
  - add digits at beginning or end

SS7 physical layer

- classic SS7 signalling over TDM circuits
  - E1 timeslot (64kbps)
  - multiple E1 timeslots (N*64kbps)
  - MTP Level 2 / MTP Level 3
- modern networks use SIGTRAN
  - IP as network layer replaces E1 lines
  - SCTP on top(no TCP/UDP!)
  - many different SIGTRAN stacking options
- some vendor-proprietary protocols like SCCPlite

SIGTRAN stacking options

SIGTRAN != SIGTRAN

- IP/SCTP/M2PA/MTP2/MTP3/SCCP/TCAP/MAP
- IP/SCTP/M2UA/MTP3/SCCP/TCAP/MAP
- IP/SCTP/M3UA/SCCP/TCAP/MAP
- IP/SCTP/SUA/TCAP/MAP

SCCP

SCCP takes care of

- Global Title based addressing
- Global Title Translation
- connection-oriented or connectionless semantics
- GSM core network interfaces with MAP/CAP only use connection-less UDT service

TCAP

- Idea: decouple transaction logic from actual application
- transaction semantics can be used by multiple higher-layer protocols
- state machines on both sides maintained outside of application
- protocol specified in ASN.1, BER encoding

MAP - Mobile Application Part

- used between all classic GSM core network components
- application protocol on top of TCAP
- protocol specified in ASN.1, BER encoding

CAP - Camel Application Part

- used for CAMEL entities (gsmSCF, gsmSSF, gprsSSF, gsmSRF)
- application protocol on top of TCAP
- protocol specified in ASN.1, BER encoding

Introduction to Roaming

Roaming enables subscribers to use other operators' networks

- Home Network is called HPLMN
- Visited Network is called VPLMN
- Roaming requres between HPLMN and VPLMN
  - Roaming agreement (contract)
  - SS7 connectivity (ISUP/MAP/CAP)
  - IP connectivity (for packet data)

Roaming principle

- MS, MSC, VLR and SGSN are in VPLMN
- HLR, AUC, GMSC and GGSN are in HPLMN
- they talk to each other via MAP, just like in non-roaming case
- selection of HPLMN based on IMSI of subscriber
- non-roaming caes: HPLMN == VPLMN

MVNO - Mobile Virtual Network Operators

A MVNO setup is a special case of roaming

- MNO operates PLMN with RAN and CN
- MVNO operates HPLMN without RAN (BSC/BTS)
- MVNO subscribers always roam into MNO network

Traditional Billing

Initially, GSM was designed for business users

- Billing was always post-paid
- Each PLMN simply logs all call/sms
- Logs called CDR (Call Data Record)
- At the end of the month, invoices are generated
- CDR records are exchanged between roaming partners

Billing for Roaming

- CDR files often vendor-specific / custom
- GSMA established a standard called TAP
- TAP is the standard for exchange of billing records between roaming partners
- Summary: Intra-PLMN: CDR, Inter-PLMN: TAP
- TAP has many versions/generations
- Specified in ASN.1

The advent of pre-paid

- At some point, users wanted pre-paid services
- Difficult to implement in traditional billing architecture
- In HPLMN, every operator could come up with custom solution
- Thus, pre-paid initially not supported in roaming
- In the early pre-paid days, there were lots of ways to exceed pre-paid balance

Pre-paid required fundamental changes

- The pre-paid balance / account is maintained in HPLMN
- HPLMN needs much more control over user while roaming
- A new protocol (CAMEL) was introduced, as well as new entities in the network
- Lots of changes all over netowrk elements (MSC, SGSN, HLR)

CAMEL - Customized Applications Mobile Enhanced Logic

- gsmSCF - Service Control Function
  - receives per-subscriber specific config from HLR (CSI: CAMEL Subscription Information)
  - remotely controls call, SMS, etc. processing
- gsmSSF - Service Switching Function
  - built into MSC
  - hooks / triggers at key state changes
  - allows gsmSCF to alter/override/abort transactions
- gprsSSF provides similar feature inside SGSN

Erlang osmo_ss7

- Signalling link management
- Signalling linkset management
- MTP-level routing
- Protocol codecs
  - BSSMAP, ISUP, M2PA, M2UA, M3UA, MTP3, SCCP, SUA
- Various different protocol implementations
  - SIGTRAN: M3UA, M2PA, M2UA, SUA
  - IPA multiplex / SCCP lite

Erlang osmo_sccp

SCCP implementation, typcially used on top of osmo_sccp

- SCCP connectionless (SCLC)
- SCCP connection oriented (SCOC)
- SCCP routing / gtt (SCRC)
- applications can bind to SSN numbers

Erlang osmo_map

- Not a full-blown MAP end-user implementation
- Primarily a set of integrated TCAP+MAP codec
- Used for protocol analysis/dissection
- Used for transparent MAP mangling engines
- Think of FTP/IRC NAT in TCP/IP, where you need to modify addresses contained in the payload (not header) of the messages

Erlang mgw_nat

- Strange transparent SCCP/TCAP/MAP gateway
- Supports all kinds of strange operations
  - SCCP Global Title Masquerade (dynamic GT pool)
  - Replace VLR/MSC GT inside MAP payload
  - Supported Camel Phase patching
  - 1:1 IMSI mapping in MAP payload
  - ISUP GT mangling
  - national/international numbering plan conversions
- Used in multiple production installations for 1 year

Erlang signerl TCAP

- Full ITU-T TCAP implementation
- 1:1 mapping of ITU-T TCAP state machines to Erlang gen_fsm
  - DHA - Dialogue Handling
  - TSM - Transaction State Machine
  - ISM - Invocation State Machine
- 1:1 mapping of other ITU-T entities to Erlang gen_server
  - CCO - Componen Coordinator
  - TCO - Transaction Coordinator
- Some old/incomplete/bit-rotten ANSI TCAP code

Erlang signerl TCAP

- properly implements the N-primitives to lower level
- properly implements all TR-primitives internally (TC / TR split)
- properly implements all TC-primitives towards the TCAP user
- Can be used on top of osmo_sccp
- Can be used directly by application servers or via signerl MAP

Erlang signerl MAP

- Interface between MAP primitives and TCAP primitives
- Provides very little benefit over using TCAP directly
- Not used much so far, I always use TCAP user API instead

Erlang application servers

- No complete implementation of any GSM core network node yet
- Lots of testing / experimentation code for generating single MAP transactions against existing/proprietary core network components
- Work on a HLR based on Mnesia DB should be starting soon

libosmo-sccp

- minimalistic SCCP implementation
- only used inside IPA multiplex / SCCP lite
- no retransmissions / GT routing / translation
- stable, used in production (osmo-bsc)

libosmo-asn1-tcap

- asn1c-generated TCAP codec
- almost no manual code
- built as shared library

libosmo-tcap

- First attempt of Harald to implement TCAP (before Erlang)
- 1:1 mapping of ITU-T TCAP components to C source files
- Heavily based on asn1c-generated data structures
- Uses libosmo-asn1-tcap

libosmo-asn1-map

- asn1c-generated MAP code
- almost no manual code
- built as shared library

Future of C implementation?

- unclear at this point
- first finish testing/deploying Erlang implementations
- possible use case for Gc interface of osmo-sgsn (SGSN-HLR)
- Do we interface C code with Erlang MAP or maintain C implementation in parallel?