

External USSD interface

Vadim Yanitskiy, OsmoDevCon 2018, 2018-04-20, Berlin

Introduction into Supplementary Services and USSD

Supplementary Services:

- Call related (`pdisc = GSM48_PDISC_CC`)
- Call independent (`pdisc = GSM48_PDISC_NC_SS`)

What is USSD?

- Unstructured Supplementary Services Data
- A type of call independent Supplementary Services
- Similar to SMS, but a real-time communication is assumed
- Usually, USSD requests look like: `*FOO*BAR#` or `*#FOO#`
 - e.g. `*#100#` in both OsmoMSC and OpenBSC
- Up to 160 bytes payload length

Use cases of USSD:

- Activation / deactivation of custom (paid) network services
- Getting information about custom (paid) network services
- Getting information about subscriber data (e.g. balance)

How do we support USSD in Osmocom

libosmocore GSM 04.80 API:

- Abstract representation of SS/USSD request `struct ss_request`:
 - Transaction Identifier (see 04.07 for details),
 - Invoke ID (matches a request with the response),
 - Operation code (e.g. `GSM0480_OP_CODE_PROCESS_USS_REQ`),
 - USSD payload (up to 160 octets) and its actual length,
 - USSD DCS (Data Coding Scheme).
- Decoding of 'raw' L3 message into `struct ss_request`
- Encoding of some USSD message types into *raw* L3 message:
 - USSD notification,
 - USSD response.

Future plans:

- Improve decoding of call related SS
- Implement encoding of MS-originated USSD messages

How do we support USSD in Osmocom

OsmoNiTB & OsmoMSC:

- Own number request (`*#100#`) only
- Adding support of new codes requires:
 - modification the source code (`ussd.c`),
 - (re)compilation of the source code,
 - restarting the network if active,
 - ⇒ too complicated for end user :/

External USSD interface

We have external interfaces for:

- SMS (via SMPP)
- voice calls (via MNCC)

External USSD interface? (OS#1597)

- Which protocol should be used?
 - MAP is assumed by the specs., but:
 - there is no stable Osmocom implementation of MAP,
 - complex protocol...
 - SMPP has some provisions for USSD, but:
 - not session-oriented nature...
 - GSUP?

GSUP?

What is GSUP?

- Generic Subscriber Update Protocol
- Osmocom-specific non-standard protocol
- Aimed to keep protocol complexity out of its users:
 - OsmoHLR,
 - OsmoMSC,
 - OsmoSGSN.
- GSUP → MAP conversation is assumed

Advantages:

- Easy to use
- Easy to extend
- Easy to convert to MAP

Disadvantages:

- Not session-oriented :(
 - but can be extended!

Current work around GSUP implementation

Session management (TCAP emulation):

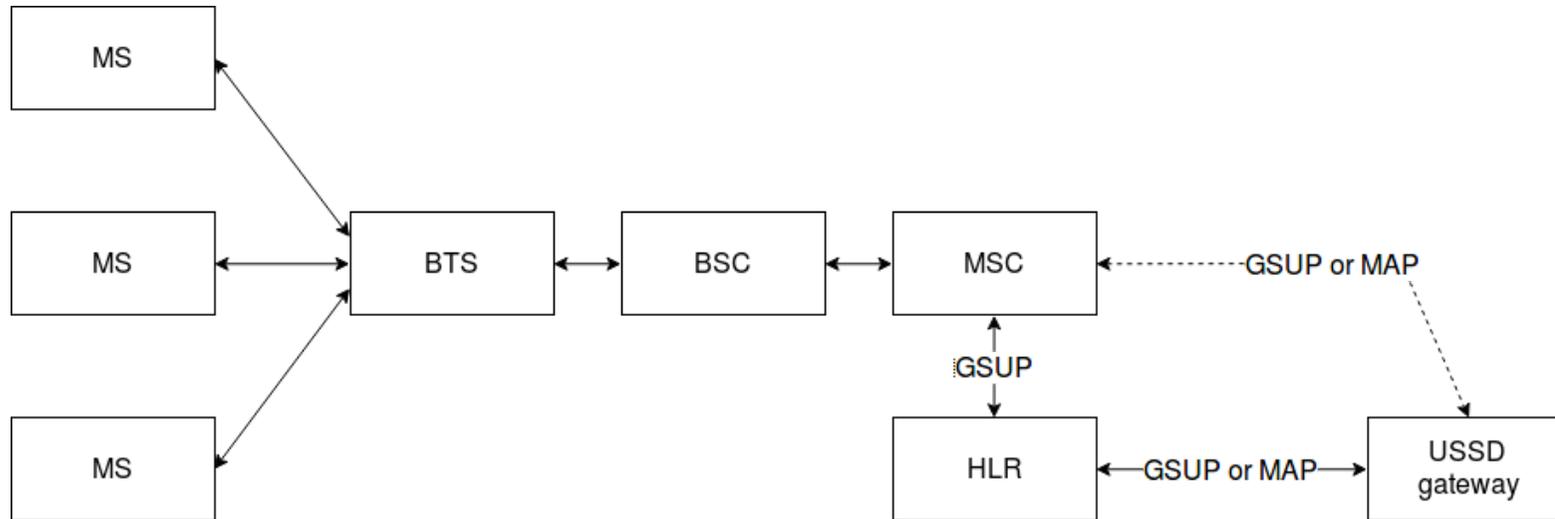
- The following new IEs are going to be introduced:
 - `OSMO_GSUP_SESSION_STATE_IE` (BEGIN, CONTINUE, END),
 - `OSMO_GSUP_SESSION_ID_IE` (i.e. unique session ID).
- Session ID generation:
 - Master-slaves relation is assumed:
 - A slave requests an unused session ID,
 - Master allocates unique session IDs,
 - Master sends it to a slave.
- The following GSUP message types are going to be introduced:
 - `OSMO_GSUP_MSGT_SESSION_ID_REQUEST`,
 - `OSMO_GSUP_MSGT_SESSION_ID_ERROR`,
 - `OSMO_GSUP_MSGT_SESSION_ID_RESULT`.

Current work around GSUP implementation

USSD payload support:

- The following new messages are going to be introduced:
 - USSD REQ / RSP:
 - OSMO_GSUP_MSGT_PROC_USS_REQ_*
 - OSMO_GSUP_MSGT_USS_REQ_*
 - Subscriber notification:
 - OSMO_GSUP_MSGT_USS_NOTIFY_*
- The following IEs are going to be introduced:
 - OSMO_GSUP_SS_INFO_IE
 - OSMO_GSUP_SS_INVOKE_ID_IE
 - OSMO_GSUP_SS_ALERTING_PATTERN_IE
 - OSMO_GSUP_USSD_STRING_DCS_IE
 - OSMO_GSUP_USSD_STRING_IE

Communication diagram



To be discussed:

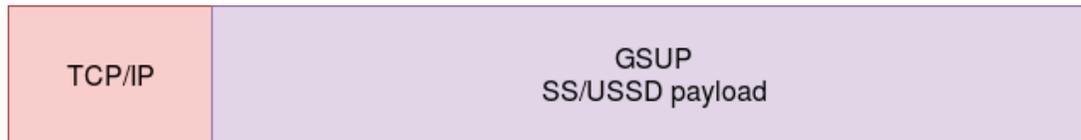
- Should we support direct USSD GW - MSC communication?

USSD - GSUP / MAP message conversation

A message coming from MS to MSC (or vice versa):



A message coming from MSC to HLR (or vice versa):



A message coming from HLR to USSD GW (or vice versa):



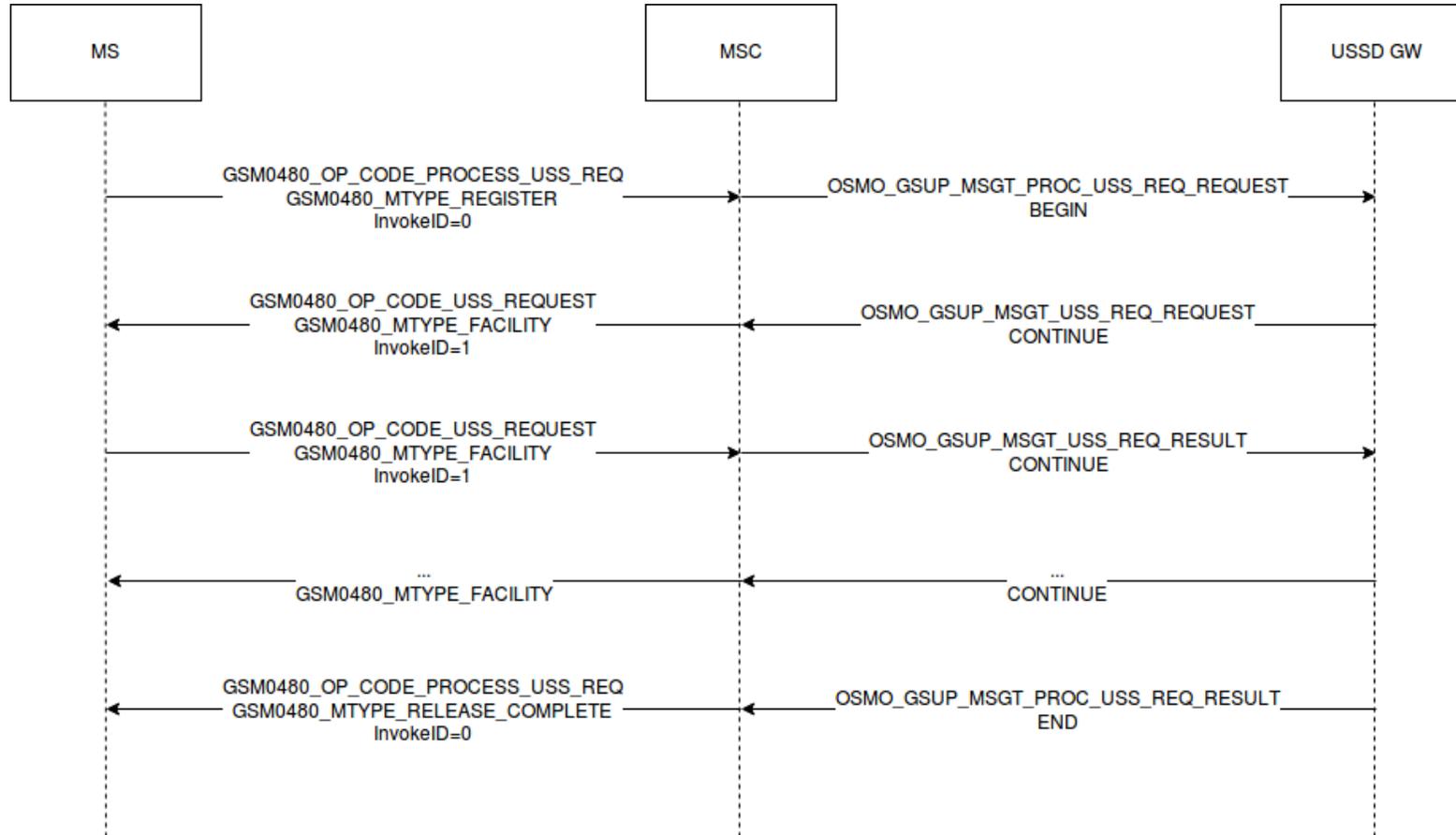
USSD - GSUP / MAP message conversation

Why MSC should parse SS/USSD requests?

- We need to know GSM 04.80 message type:
 - REGISTER → TCAP BEGIN
 - FACILITY → TCAP CONTINUE
 - RELEASE_COMPLETE → TCAP END
- We need to know the operation code:
 - It is SS?
 - GSM0480_OP_CODE_REGISTER_SS
 - GSM0480_OP_CODE_ERASE_SS
 - Is it USSD?
 - GSM0480_OP_CODE_PROCESS_USS_DATA
 - GSM0480_OP_CODE_USS_NOTIFY
- We need to know DCS of USSD payload
- Finally, we need the USSD payload itself

Message flow example:

Mobile initiated USSD-session



Status of the current implementation

What is already done:

- libosmcore GSM 04.80 API improvement:
 - Support of optional FACILITY IE in RELEASE_COMPLETE,
 - Raw USSD payload access,
 - DCS decoding.

Work in progress now:

- OsmoMSC:
 - Subscriber connection ref-counting
 - Transaction management
 - GSUP message coding

Future plans:

- TTCN3 test cases (OS#2931)
- Counters for USSD events:
 - Total number of USSD-/SS-requests
 - Number of active USSD-/SS-connections
 - Number of rejected / successful USSD-/SS-requests

Questions?

Thanks!

References:

- <https://osmocom.org/issues/1597>
- 3GPP TS 04.10 "Supplementary Services Specification"
- 3GPP TS 04.80 "Formats and coding of Supplementary Services"
- GSM TS 09.02 "Mobile Application Part (MAP) specification"
 - Section 11 "Supplementary services related services"
 - Section 7.6.4 "Supplementary services parameters"
 - Section 7.6.3 "Subscriber management parameters"
 - Section 7.6.1 "Common parameters"