OsmoMSC - Bug #3355

OsmoMSC doesn't provide unique IDTAG_SERNR in IPA CCM

06/23/2018 05:45 PM - laforge

Status:	Resolved	**Start date:**	06/23/2018
Priority:	High	**Due date:**	
Assignee:	stsp	**% Done:**	0%
Category:			
Target version:			
Resolution:			

Description

OsmoMSC currently identifies itself simply as "MSC" over the IPA/GSUP link to OsmoHLR.

This is wrong as it results in all OsmoMSCs (in a multi-MSC network) to register using the same identity, which in turn will render the HLR unable to store which MSC is actually serving the subscriber, and/or to route mesages accordingly.

It appears the change to "MSC" was made in Change-Id: I0a60681ab4a4d73e26fe8f0637447db4b6fe6eb2 from "SGSN-00-00-00-00-00-00" before. Both the old and the new name are wrong though, as they're not unique.

The correct behavior AFAICT is "MSC-" prefix with whatever unque identifier behind. We could make that suffix configurable in the VTY, if there's no other way to derive any other locally unique identifier. MAC adresses work fine for BTSs, but not for MSCs: There might be any number of OsmoMSCs running on the same physical / virtual machine with only one MAC address...

Related issues:

Related to OsmoHLR - Bug #2796: OsmoHLR doesn't update VLR during UpdateLocation	**In Progress**	**12/30/2017**	
Related to OsmoSGSN - Bug #3356: OsmoSGSN doesn't provide unique IDTAG_SERNR ...	**In Progress**	**06/23/2018**	
Related to OsmoHLR - Feature #3644: allow arbitrary bytes in Global Title (GT...	**New**	**10/11/2018**	
Related to OsmoHLR - Bug #3710: hlr_ussd.c hardcodes "MSC-00-00-00-00-00-00" ...	**Resolved**	**11/26/2018**	
Blocks OsmoMSC - Feature #3618: Inter-MSC hand-over support	**In Progress**	**10/02/2018**	
Blocks OsmoMSC - Feature #1609: Inter-BSC hand-over is missing (MSC side)	**Resolved**	**11/21/2016**	**11/21/2016**

History

#1 - 06/23/2018 05:45 PM - laforge

- Related to Bug #2796: OsmoHLR doesn't update VLR during UpdateLocation added

#2 - 06/23/2018 05:49 PM - laforge

- Related to Bug #3356: OsmoSGSN doesn't provide unique IDTAG_SERNR in IPA CCM added

#3 - 06/23/2018 05:51 PM - laforge

Note: The identifier must stay unique even across re-starts of the MSC.

#4 - 07/30/2018 02:49 PM - stsp

Would uuid_generate be a reasonable solution?
https://linux.die.net/man/3/uuid_generate

#5 - 08/20/2018 09:37 AM - neels

stsp wrote:

> Would uuid_generate be a reasonable solution?
> https://linux.die.net/man/3/uuid_generate

The difficulty is to get the same UUID **across program restarts**, i.e. no random involved.
We should probably use an ID to be set in the config file (and loudly error-log if missing)?

#6 - 09/28/2018 01:21 AM - neels

it would be good if the identification were at most 15 characters wide, so it fits in the HLR's vlr_number.

#7 - 10/02/2018 02:42 PM - neels

neels wrote:

> it would be good if the identification were at most 15 characters wide, so it fits in the HLR's vlr_number.

OK, turns out the GSUP clients are free to send pretty much anything they wants (in the sense of arbitrary Global Title),
and osmo-hlr should be able to deal with that.

#8 - 11/26/2018 10:47 PM - neels

- Related to Feature #3644: allow arbitrary bytes in Global Title (GT), a.k.a. the VLR,SGSN's identification towards the HLR added

#9 - 11/26/2018 11:09 PM - neels

This becomes a necessary prerequisite for inter-MSC handover:

We will connect between MSCs via GSUP, and hence the individual MSCs need individual identification.
I believe with multiple MSCs and the need to associate each MSC's BSCs with specific BSSMAP Cell Identifiers, the MSCs' identifications should be configurable in osmo-msc.cfg.
Looking at the code, though, I'm a bit puzzled how the current 'MSC-00-00-00-00-00-00' identification is even created:

- I see vlr_start() sets the unit_name to "MSC", calling osmo_gsup_client_create("MSC",...)
- but I can't find the place that concatenates the two to form the IDTAG_SERNR that arrives at OsmoHLR:
  - there is libosmo-netif/src/ipa.c "case IPAC_IDTAG_UNITNAME: osmo_ipa_unit_snprintf_name(str, sizeof(str), dev);" which contains a "%s-%02x-..." fmt;
  - there is libosmo-abis/src/input/ipaccess.c "case IPAC_IDTAG_UNITNAME: snprintf(str, sizeof(str), "%s-%02x-%02x-%02x-%02x-%02x-%02x", dev->unit_name, dev->mac_addr[0]...";
- but at OsmoHLR we actually retrieve the IPAC_IDTAG_SERNR, not the UNITNAME!? see lu_op_alloc_conn() in osmo-hlr/src/luop.c

We need to decide how to resolve this soon, and someone needs to figure out what code we're actually using, and how the IDTAG_UNITNAME mysteriously becomes the IDTAG_SERNR on its way to OsmoHLR.

#10 - 11/26/2018 11:11 PM - neels

- Blocks Feature #3618: Inter-MSC hand-over support added

#11 - 11/26/2018 11:11 PM - neels

- Blocks Feature #1609: Inter-BSC hand-over is missing (MSC side) added

#12 - 11/26/2018 11:29 PM - neels

- Related to Bug #3710: hlr_ussd.c hardcodes "MSC-00-00-00-00-00-00" twice added

#13 - 12/03/2018 04:26 PM - stsp

neels wrote:

> Looking at the code, though, I'm a bit puzzled how the current 'MSC-00-00-00-00-00-00' identification is even created:
>
> - I see vlr_start() sets the unit_name to "MSC", calling osmo_gsup_client_create("MSC",...)
> - but I can't find the place that concatenates the two to form the IDTAG_SERNR that arrives at OsmoHLR:
>   - there is libosmo-netif/src/ipa.c "case IPAC_IDTAG_UNITNAME: osmo_ipa_unit_snprintf_name(str, sizeof(str), dev);" which contains a "%s-%02x-..." fmt;
>   - there is libosmo-abis/src/input/ipaccess.c "case IPAC_IDTAG_UNITNAME: snprintf(str, sizeof(str), "%s-%02x-%02x-%02x-%02x-%02x-%02x", dev->unit_name, dev->mac_addr[0]...";

It is constructed there (ipaccess.c line 621) at the sender.
The mac adddress of the MSC is initialized to zeros; so the resulting string is "MSC-00-00-00-00-0-00".

The unit name is added because the IPA peer requests it:

```
<0013> input/ipaccess.c:705 received ID get from 0/0/0

<0013> input/ipaccess.c:639  tag 8: 0/0/0
<0013> input/ipaccess.c:639  tag 7: 00:00:00:00:00:00
<0013> input/ipaccess.c:639  tag 2: 00:00:00:00:00:00
<0013> input/ipaccess.c:639  tag 3: 00:00:00:00:00:00
<0013> input/ipaccess.c:639  tag 4: 00:00:00:00:00:00
<0013> input/ipaccess.c:639  tag 5: 00:00:00:00:00:00
<0013> input/ipaccess.c:639  tag 1: MSC-00-00-00-00-00-00
<0013> input/ipaccess.c:639  tag 0: MSC-00-00-00-00-00-00
```

- but at OsmoHLR we actually retrieve the IPAC_IDTAG_SERNR, not the UNITNAME!? see lu_op_alloc_conn() in osmo-hlr/src/luop.c

Well, we retrieve all tags sent by the IPA peer and store them in parsed form.

The corresponding debug log of the peer (i.e. osmo-hlr) looks like this:

```
20181129132858893 DLINP DEBUG 127.0.0.1:36286 message received (ipa.c:337)
Unit_ID='0/0/0' MAC_Address='00:00:00:00:00:00' Location_1='00:00:00:00:00:00' Location_2='00:00:00:00:00:00'
Equipment_Version='00:00:00:00:0
0:00' Software_Version='00:00:00:00:00:00' Unit_Name='MSC-00-00-00-00-00-00' Serial_Number='MSC-00-00-00-00-00
-00' 20181129132858893 DLGSUP IN
FO CCM Callback (gsup_server.c:182)
20181129132858893 DLGSUP INFO 0: MSC-00-00-00-00-00-00 (gsup_server.c:138)
20181129132858893 DLGSUP INFO 0: 4d 53 43 2d 30 30 2d 30 30 2d 30 30 2d 30 30 2d 30 30 2d 30 30 00  (gsup_serv
er.c:141)
20181129132858893 DLGSUP INFO 1: MSC-00-00-00-00-00-00 (gsup_server.c:138)
20181129132858893 DLGSUP INFO 1: 4d 53 43 2d 30 30 2d 30 30 2d 30 30 2d 30 30 2d 30 30 2d 30 30 00  (gsup_serv
er.c:141)
20181129132858893 DLGSUP INFO 2: 00:00:00:00:00:00 (gsup_server.c:138)
20181129132858893 DLGSUP INFO 2: 30 30 3a 30 30 3a 30 30 3a 30 30 3a 30 30 3a 30 30 00  (gsup_server.c:141)
20181129132858893 DLGSUP INFO 3: 00:00:00:00:00:00 (gsup_server.c:138)
20181129132858893 DLGSUP INFO 3: 30 30 3a 30 30 3a 30 30 3a 30 30 3a 30 30 3a 30 30 00  (gsup_server.c:141)
```

```
20181129132858893 DLGSUP INFO 4: 00:00:00:00:00:00 (gsup_server.c:138)
20181129132858893 DLGSUP INFO 4: 30 30 3a 30 30 3a 30 30 3a 30 30 3a 30 30 3a 30 30 00  (gsup_server.c:141)
20181129132858893 DLGSUP INFO 5: 00:00:00:00:00:00 (gsup_server.c:138)
20181129132858893 DLGSUP INFO 5: 30 30 3a 30 30 3a 30 30 3a 30 30 3a 30 30 3a 30 30 00  (gsup_server.c:141)
20181129132858893 DLGSUP INFO 7: 00:00:00:00:00:00 (gsup_server.c:138)
20181129132858893 DLGSUP INFO 7: 30 30 3a 30 30 3a 30 30 3a 30 30 3a 30 30 3a 30 30 00  (gsup_server.c:141)
20181129132858893 DLGSUP INFO 8: 0/0/0 (gsup_server.c:138)
20181129132858893 DLGSUP INFO 8: 30 2f 30 2f 30 00  (gsup_server.c:141)
20181129132858893 DMAIN INFO Adding GSUP route for MSC-00-00-00-00-00-00 (gsup_router.c:64)
```

> We need to decide how to resolve this soon, and someone needs to figure out what code we're actually using, and how the IDTAG_UNITNAME mysteriously becomes the IDTAG_SERNR on its way to OsmoHLR.

As explained above, it finds its way there via TLV parsing.

#14 - 12/03/2018 04:30 PM - stsp

I suppose we could invent a configuration option which sets the "MAC address" (or rather "MSC ID") of the MSC.
We don't need to call it "MAC address" in user-facing documentation and the UI, but it could be stored in the same place so that the above code which generates the IPA ID request response will work without modification.
And this "MSC ID" could look just like a MAC address syntactically.

#15 - 12/03/2018 08:08 PM - neels

TLDR: I would like to see a patch that changes the code so that we fully control at least one of the ID tags from the osmo-msc.cfg file.

Details:

So, currently, due to some magic (which I still don't fully understand), IDTAG_SERNR == IDTAG_UNITNAME.

```
Unit_Name='MSC-00-00-00-00-00-00' Serial_Number='MSC-00-00-00-00-00-00'
```

What are we going to do next?

- I need to be able to identify one specific MSC by one tag or the other,
  in a way that is transparent and obvious to the user. Best: fully user provided,
  without magic "MAC address" stuff happening.
- Currently the unit name is fed into the API as just "MSC", and internally extended by the "MAC address",
  which isn't really a MAC address to begin with, plus we don't really want to use a MAC address either.

- To me the tag "Unit_Name" makes more sense to identify an MSC than "Serial_Number"? (right?)
  Or is Unit_Name like a generic "MSC" or "SGSN", and Serial_Number would be the universally unique identifier?

- I think it's nonsense to send the MAC address N times over in Serial_Number, Unit_Name, Software_Version, Equipment_Version as well as Location_{1,2},
  especially since the MAC address is all-zero for all clients. If we made this more sensible, our GSUP messages could have a shorter IPA header.

- Suggestion:
  - Unit_Name="MSC-901-70-0" -- leave the unit name entirely up to user configuration, only recommending this format in the docs (PLMN + redundancy server nr?)
  - Software_Version="osmo-msc-1.2.0.96-56891a" (or even empty/omit)
  - Location1="00:00:00:00:00:00" Location2="192.168.0.1" (MAC + IP of the interface used to route to osmo-hlr? omit?)
  - Equipment_Version="" Serial_Number="" (omit?).

I am completely unsure about what the individual ID_TAGs are supposed to mean.
All I know is the current content is completely useless :P and that I need at least one useful one.

#16 - 12/04/2018 10:26 AM - stsp

neels wrote:

> So, currently, due to some magic (which I still don't fully understand), IDTAG_SERNR == IDTAG_UNITNAME.

The "magic" is reuse of local buffer 'str' in ipa_bts_id_resp().
Both of these lines log the same buffer:

```
<0013> input/ipaccess.c:639  tag 1: MSC-00-00-00-00-00-00
<0013> input/ipaccess.c:639  tag 0: MSC-00-00-00-00-00-00
```

The tags are defined as follows:

```
        IPAC_IDTAG_SERNR                = 0x00,
        IPAC_IDTAG_UNITNAME             = 0x01,
```

The tags are iterated in the order given by this definition in libosmocore/src/gsm/ipa.c:

```
static const uint8_t ipa_id_req_msg[] = {
        0, 17, IPAC_PROTO_IPACCESS, IPAC_MSGT_ID_GET,
        0x01, IPAC_IDTAG_UNIT,
        0x01, IPAC_IDTAG_MACADDR,
        0x01, IPAC_IDTAG_LOCATION1,
        0x01, IPAC_IDTAG_LOCATION2,
        0x01, IPAC_IDTAG_EQUIPVERS,
        0x01, IPAC_IDTAG_SWVERSION,
        0x01, IPAC_IDTAG_UNITNAME,
        0x01, IPAC_IDTAG_SERNR,
};
```

So the unit name is parsed before the serial number.
When the unit name is parsed, the buffer 'str' in ipa_bts_id_resp() is populated with the string "MAC-00-00-00-00-00-00".
Below the switch statement, the tag gets logged:

(./libosmo-abis/src/input/ipaccess.c:639)

```
 LOGP(DLINP, LOGL_INFO, " tag %d: %s\n", data[1], str);
```

Then the serial number is parsed, but apparently nothing gets copied to 'str' because 'dev->serno' is NULL?

(./libosmo-abis/src/input/ipaccess.c:629)

```
                case IPAC_IDTAG_SERNR:
                        if (dev->serno)
                                osmo_strlcpy(str, dev->swversion, sizeof(str));
                        break;
```

Then the content of 'str' gets logged again.

So that explains the "magic". Just please don't ask me what this code intends to do...

#17 - 12/04/2018 10:46 AM - stsp

neels wrote:

> I am completely unsure about what the individual ID_TAGs are supposed to mean.
> All I know is the current content is completely useless :P and that I need at least one useful one.

It seems out biggest problem is that, currently, the content of tags is mostly zeroes in the MSC case.

I'd rather not change the meaning and parsing of these tags at all if we can avoid doing so.

This code is shared between several components, not just between MSC and HLR.
There are also multiple copies of related code which have proliferated which makes such changes even more risky.
E.g. compare ipa_bts_id_resp() in libosmo-abis/src/input/ipaccess.c to ipa_ccm_make_id_resp() in libosmocore/src/gsm/ipa.c.

Ultimately, the content of tags is derived from the contents of this structure:

```
struct ipaccess_unit {
        uint16_t site_id;
        uint16_t bts_id;
        uint16_t trx_id;
        char *unit_name;
        char *equipvers;
        char *swversion;
        uint8_t mac_addr[6];
        char *location1;
        char *location2;
        char *serno;
};
```

I suppose we should investigate how we can make the MSC fill this structure with values that end up generating useful tags.

#18 - 12/04/2018 11:09 AM - neels

> apparently nothing gets copied to 'str' because 'dev->serno' is NULL?

oh! ah. ouch. That's really interesting. Excellent, I finally understand what is going on.
Maybe it was even intended this way, but I still think it makes no sense.

> I'd rather not change the meaning and parsing of these tags at all if we can avoid doing so.

Cleaning up the tags isn't urgent; seems sane to me but we don't need to rock that boat now.

> I suppose we should investigate how we can make the MSC fill this structure with values that end up generating useful tags.

The requirement is: fully control at least one of the ID tags from the osmo-msc.cfg file.
Related: IIUC normally such an identification is done by GT (Global Title), which is an arbitrary blob.
Can you try placing an arbitrary string in IDTAG_SERNR?

#19 - 12/04/2018 12:23 PM - stsp

neels wrote:

> Can you try placing an arbitrary string in IDTAG_SERNR?

I found where the MSC's (client-side) ipaccess_unit is coming from.
It's a local definition in osmo-hlr/src/gsupclient/gsup_client.c:

```
167 static int gsup_client_read_cb(struct ipa_client_conn *link, struct msgb *msg)
168 {
169         struct ipaccess_head *hh = (struct ipaccess_head *) msg->data;
170         struct ipaccess_head_ext *he = (struct ipaccess_head_ext *) msgb_l2(msg);
171         struct osmo_gsup_client *gsupc = (struct osmo_gsup_client *)link->data;
172         int rc;
173         struct ipaccess_unit ipa_dev = {
174                 /* see gsup_client_create() on const vs non-const */
175                 .unit_name = (char*)gsupc->unit_name,
176         };
```

```
(gdb)
173             struct ipaccess_unit ipa_dev = {
(gdb)
175                     .unit_name = (char*)gsupc->unit_name,
(gdb)
173             struct ipaccess_unit ipa_dev = {
(gdb)
178             OSMO_ASSERT(ipa_dev.unit_name);
(gdb) p ipa_dev
$1 = {site_id = 0, bts_id = 0, trx_id = 0, unit_name = 0x60b0001327e0 "MSC", equipvers = 0x0, swversion = 0x0,
 mac_addr = "\000\000\000\000\000", location1 = 0x0, location2 = 0x0, serno = 0x0}
```

It shouldn't be hard to allow callers to override some of those fields.

#20 - 12/04/2018 01:01 PM - stsp

As a first step, the GSUP client API could allow callers to pass in a 'struct ipaccess_unit' instead of a unit name.
See https://gerrit.osmocom.org/#/c/osmo-hlr/+/12098

#21 - 12/04/2018 01:12 PM - stsp

- Status changed from New to In Progress

#22 - 12/06/2018 05:18 PM - stsp

This patch allows users to configure an IPA name in osmo-msc.cfg: https://gerrit.osmocom.org/c/osmo-msc/+/12177

This patch (depends on the former) makes osmo-msc announce its software version to gsup peers: https://gerrit.osmocom.org/c/osmo-msc/+/12178

#23 - 12/11/2018 10:21 AM - stsp

This issue is still waiting for review by neels of https://gerrit.osmocom.org/c/osmo-msc/+/12177

#24 - 12/11/2018 01:08 PM - stsp

- Status changed from In Progress to Resolved

Above patches have been merged.