Distributed GSM - Feature #4261

facilitate GSUP proxy routability: osmo-hlr needs to copy rx source_name to tx destination_name

11/08/2019 03:00 AM - neels

Status:	Resolved	**Start date:**	11/08/2019
Priority:	Urgent	**Due date:**	
Assignee:	neels	**% Done:**	100%
Category:			
Target version:			
Spec Reference:			

Description

Testing GSUP proxy behavior of osmo-hlr, I notice this pickle we're currently in with GSUP:

For any given IMSI, there can be more than one VLR: typically one MSC VLR, and one SGSN VLR.
In short, there can be more than one GSUP client sending requests to osmo-hlr for a given IMSI.

When there is now a GSUP proxy between these multiple VLRs and the servicing home HLR, requests for each need to be routed back reliably.
Two ideas:

1) use source_name and destination_name. (todo: make sure we don't interfere with inter-MSC handover use of those IEs, in doubt add separate similar IEs)
When a proxy receives a message from a VLR, it adds a source_name IE to that GSUP message before passing it on.
So, say, a LU Request coming directly from an MSC will not contain a source_name,
but as soon as it went through a proxy, the presence of a source_name a) indicates that a proxy is involved and b) which is the original MSC that asked.
Now, if osmo-hlr would always take a source_name if it finds one, and placed that into the responses sent back, we could always route back to the right MSC.
The problem with this is that the osmo-hlr code is not so trivial to adjust in this manner.
I would most prefer a central place that makes this decision, but the code composing replies and sending them back to the peer is scattered throughout the code base.
Unless we want to edit every single of those message compositions (error prone), it would require an effort to refactor osmo-hlr so that each response passes through one specific point in the code to be encoded and sent, which always still knows the requesting message.
One fundamental problem is that there are GSUP requests that are serviced asynchronously, and that it is "hard" to keep a GSUP message asynchronously (struct osmo_gsup_message has lots of pointers pointing at memory with ambiguous ownership).
So we would need to solve this problem of keeping a GSUP request -> response association across async servicing in general, and then we could in one simple step add the source_name -> destination_name routing information.

2) The other idea is to separate CS and PS domains.
The great advantage is that we don't need to modify what osmo-hlr responds.
We just figure out who is the current CS peer and who is the PS peer, and according to the gsup_msg->cn_domain always route back like that.
The practical problem here is that actually most of our GSUP messages completely lack the cn_domain field, osmo-msc sends even the initial SendAuthInfo Request without a CS domain indication...
The theoretical problem here is that as soon as there is a mixup with more than one peer claiming to be CS (or PS), responses from the one CS peer get routed back to the other CS peer.
So we would need to make sure all clients restlessly always send a cn_domain IE,
and would still hit an ambiguity for multiple peers on a given cn_domain.

Writing this out has clarified my mind that option 1) is clearly better.
If we ensure a source_name always comes back as destination_name, we only need to modify osmo-hlr, the GSUP server,
and then we avoid routing failures if more than one client claims to be CS (or PS).
The refactoring in osmo-hlr could be a bit irritating, but it is worth the additional effort,
since it probably also forces the code paths into more clarity about handling of GSUP messages.

History

#1 - 11/11/2019 06:23 AM - neels

- File short_trace.pcapng added

- Status changed from New to In Progress

- % Done changed from 0 to 80

i've got a working patch for osmo-hlr that makes proxy routing work.
Added ttcn3-hlr-tests that verify source_name and destination_name for routing.
Ran a successful LU with an actual phone using an osmo-hlr as proxy and mDNS to forward to the "home" osmo-hlr.

The patch still needs to be separated from non-specific refactoring to prepare for this change.
Among other things, I found it better to move the osmo-hlr's luop.c into an osmo_fsm implementation.

#2 - 11/13/2019 06:26 AM - neels

- File success_ussd_via_two_hlr_proxies.tgz added

#3 - 11/29/2019 11:50 AM - neels

- % Done changed from 80 to 90

#4 - 11/29/2019 11:51 AM - neels

https://gerrit.osmocom.org/c/osmo-hlr/+/16205

#5 - 05/06/2020 12:52 AM - neels

- Status changed from In Progress to Resolved

- % Done changed from 90 to 100

merged

Files

short_trace.pcapng	1.11 MB	11/11/2019	neels
success_ussd_via_two_hlr_proxies.tgz	300 KB	11/13/2019	neels