

## OsmoSTP - Bug #3838

### connecting two BSCs to OsmoSTP will both end up with the same routing key

03/13/2019 06:53 AM - laforge

<b>Status:</b>	Feedback	<b>Start date:</b>	03/13/2019
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	laforge	<b>% Done:</b>	0%
<b>Category:</b>			
<b>Target version:</b>			

#### Description

From IRC:

```
01:43 < neels> 20190313014039861 DLSS7 INFO asp-asp-dyn-16: RKM: Registering routing key 4 for DP
C 0.23.3 (xua_rkm.c:197)
01:43 < neels> 20190313014039862 DLSS7 NOTICE asp-asp-dyn-16: RKM: Found existing AS for RCTX 4 (
xua_rkm.c:215)
01:43 < neels> 20190313014039862 DLSS7 NOTICE asp-asp-dyn-16: RKM: DPC doesn't match (2236 != 187
) (xua_rkm.c:218)
01:44 < neels> If I configure one of the two with a different routing key, both can connect.

07:31 <@LaF0rge> neels: routing keys are at the M3UA/SIGTRAN level. Point codes at the MTP level
and SCCP addresses at the SCCP level. And every protocol level needs unique identifiers.
07:32 <@LaF0rge> neels: however, in the pcap files you shared with me around last midnight, the
different BSCs were indeed using different routing keys
07:40 <@LaF0rge> neels: two M3UA connections with the same routing key *intentionally* map to the
same AS. This is used for round-robin/load-sharing setups or alternatively, depending on the traf
fic mode, for fail-over
```

The interesting bit here is why, when using dynamic routing key registration, both BSCs would get the same routing key. That would clearly be a bug. The entire point of dynamic registration is to avoid any "a priori" manual configuration of settings like routing keys on all related systems. That's what the "RKM" (routing key management) sub-system of SIGTRAN/M3UA is all about.

If you look at the attached pcap file and specifically look for the "RKM" messages "m3ua.message\_class == 9", then you will see:

- system with point code 188 registers for **hard coded** routing context 4
- system with point code 185 registers for routing context 0 (dynamic) and gets allocated 2
- system with point code 189 registers for routing context 0 (dynamic) and gets allocated 3
- system with point code 187 registers for **hard coded** routing context 4
- system with point code 188 registers for **hard coded** routing context 4

So somehow, some of the M3UA clients use **overlapping** statically-configured routing keys while others do the right thing and ask the STP to allocate one dynamically. This could either be a configuration mistake, or it could be bugs in the respective code base. \*

#### History

##### #1 - 03/21/2019 02:17 AM - neels

- Status changed from New to Feedback

- Assignee changed from neels to laforge

I actually had explicit routing keys in the .cfg files.

This comes from the need to configure distinct point-codes for the two BSC.

We actually still lack a similar section in the osmo-bsc manual, but looking at the osmo-msc manual section 5.4.1:

```
A full configuration needs an asp on an as – an Application Server Process running on an Application Server –
as well as a
local point code and routing configuration. The SCCP VTY automatically creates those parts that are missing, b
y assuming sane
defaults. A complete configuration would look like this:
```

```
cs7 instance 0
```

```
point-code 0.23.1
asp my-OsmoMSC-A-Iu 2905 0 m3ua
  remote-ip 127.0.0.1
as my-as-for-OsmoMSC-A-Iu m3ua
  asp my-OsmoMSC-A-Iu
  routing-key 0 0.23.1
```

i.e. for some weird reason the local point-code apparently needs to be repeated in the AS config. Even though I probably wrote the manual for this, I can't claim to understand any of this config in detail. This cfg shown in the manual probably came from the 'show running-config' output.

I just checked, I cannot just drop the 'routing-key' line, the result is that both BSCs constantly go through their RESET cycle, and neither of them ever receive a RESET ACK.

What works though is that I completely drop the entire 'as' section in both osmo-bsc cfgs, then the point-code configured on top gets applied to the 'as' auto config, and things work out fine.

What I still wonder, what if a user for whatever reason wants to manually configure an 'as', it then seems impossible to still use automatic RKM?

To summarize:

- a config fix is to drop the 'as' section from 'cs7',
- the osmo-msc manual needs fixing,
- the osmo-bsc manual needs a cs7 configuration section in the first place.

## Files

---

no_reset_ack_one_bsc_at_a_time.pcap	385 KB	03/13/2019	laforge
-------------------------------------	--------	------------	---------