

OsmoSTP - Feature #4220

osmo_ss7: support traffic-mode load-share

10/08/2019 03:37 PM - laforge

Status: Closed	Start date: 10/08/2019
Priority: Urgent	Due date:
Assignee: pespin	% Done: 100%
Category:	
Target version:	
Spec Reference:	
Description The SIGTRAN RFCs mention different traffic-modes (broadcast, loadshare, roundrobin), which affect how messages received by the STP are distributed across multiple ASPs within one AS. osmo-stp currently doesn't support those traffic modes yet. We require the "load-share" traffic mode to be implemented. A simple implementation using a round-robin method of distributing inbound signaling messages across all currently active ASPs within the AS is sufficient.	
Related issues:	
Related to libosmo-sccp + libosmo-sigtran - Feature #2002: Implement xUA traf...	Resolved 04/10/2017
Precedes OsmoSTP - Feature #4218: dynamic IPA ASPs	Closed 10/09/2019 10/09/2019

History

#1 - 10/09/2019 03:54 PM - pespin

Following osmo-stp user manual section needs to be updated when this is implemented:

4.5.1.2 Application Server (AS) An Application Server is basically a logical entity representing one particular external application (from the SS7 point of view) which is interfaced with the SS7 network by means of one of the SIGTRAN protocols. An Application Server can have one or more Application Server Processes associated with it. This functionality (currently not implemented in Osmocom) can be used for load-balancing or fail-over scenarios.

#2 - 10/16/2019 10:31 AM - pespin

Seems the expected place for this to happen is in libosmo-sccp.git xua_as_transmit_msg().

#3 - 10/16/2019 11:12 AM - pespin

- Status changed from New to In Progress

Found related information in <https://tools.ietf.org/html/rfc4666> (RFC for M3UA):

4.3.4.3. ASP Active Procedures

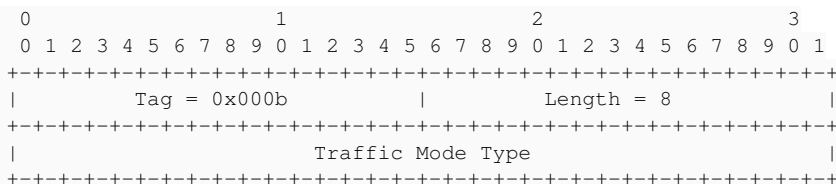
There are three modes of Application Server traffic handling in the SGP M3UA layer: Override, Loadshare and Broadcast. When included, the Traffic Mode Type parameter in the ASP Active message indicates the traffic handling mode to be used in a particular Application Server. If the SGP determines that the mode indicated in an ASP Active message is unsupported or incompatible with the mode currently configured for the AS, the SGP responds with an Error message ("Unsupported / Invalid Traffic Handling Mode"). If the traffic handling mode of the Application Server is not already known via configuration data, then the traffic handling mode indicated in the first ASP Active message causing the transition of the Application Server state to AS-ACTIVE MAY be used to set the mode.

3.6.1. Registration Request (REG REQ):

Traffic Mode Type: 32-bit (unsigned integer)

The optional Traffic Mode Type parameter identifies the traffic mode

of operation of the ASP(s) within an Application Server. The format of the Traffic Mode Type Identifier is as follows:



The valid values for Traffic Mode Type are shown in the following table:

1	Override
2	Loadshare
3	Broadcast

3.7.1. ASP Active:

Traffic Mode Type: 32-bit (unsigned integer)

The Traffic Mode Type parameter identifies the traffic mode of operation of the ASP within an AS. The valid values for Traffic Mode Type are shown in the following table:

1	Override
2	Loadshare
3	Broadcast

Within a particular Routing Context, Override, Loadshare, and Broadcast SHOULD NOT be mixed. The Override value indicates that the ASP is operating in Override mode, in which the ASP takes over all traffic in an Application Server (i.e., primary/backup operation), overriding any currently active ASPs in the AS. In Loadshare mode, the ASP will share in the traffic distribution

Similar information can be found in RFC for SUA:

<https://tools.ietf.org/html/rfc3868>

Found also some references in code:

```

enum m3ua_traffic_mode {
    M3UA_TMOD_OVERRIDE    = 1,
    M3UA_TMOD_LOADSHARE  = 2,
    M3UA_TMOD_BCAST      = 3,
};

enum osmo_ss7_as_traffic_mode {
    OSMO_SS7_AS_TMOD_OVERRIDE = 0, /* default */
    OSMO_SS7_AS_TMOD_BCAST,
    OSMO_SS7_AS_TMOD_LOADSHARE,
    OSMO_SS7_AS_TMOD_ROUNDROBIN,
    _NUM_OSMO_SS7_ASP_TMOD
};

```

And function `osmo_ss7_tmode_to_xua()` and `osmo_ss7_tmode_from_xua()` provides a way to translate between them.

Traffic mode is already parsed into "as->cfg.mode" in "as_traf_mode_cmd". It is set by default to `OSMO_SS7_AS_TMOD_LOADSHARE` in `osmo_ss7_as_find_or_create()`.

So far it seems out client code is submitting traffic mode = override in REG REQ messages, which probably changes the mode. That traffic mode is parsed and stored into `as->cfg.mode` in `handle_rkey_reg()`.

#4 - 10/16/2019 03:02 PM - pespin

Related patches:

- remote: <https://gerrit.osmocom.org/c/libosmo-sccp/+15800> ss7: Fix trailing whitespace
- remote: <https://gerrit.osmocom.org/c/libosmo-sccp/+15801> vty: Print traffic mode during show cs7 instance as
- remote: <https://gerrit.osmocom.org/c/libosmo-sccp/+15802> ss7: Implement AS traffic mode loadshare using round robin ASP selection

I'm now working on the config/VTY part in order to test it properly (I did some previous tests with loadshare and the sending part looks fine i think). I have a commit fixing several related things in libosmo-sccp, but still some stuff seems to be lacking:

- STP uses the "listen" node and I'm not sure if it can use the "as" node where the traffic-mode can be configured. I need to check if "as" can be used or I need to add the traffic-mode command to the "listen" node, so a preferred method can be forced.
- In general, the traffic mode is requested by ASPs towards the STP. Currently, If I set "traffic-mode loadshare" in "as" node of osmo-bsc.cfg, it still seems to be sending "override" during REG REQ (and nothing during AS ACT?). Once this is fixed if ASPs require "loadshare" then STP should be able to provide it.

#5 - 10/16/2019 03:04 PM - roch

Hi Pau,

Will that apply to IPA nodes?

Best regards

Roch.

Roch-Alexandre Nominé | Principal Telecommunications Architect

#6 - 10/16/2019 07:44 PM - laforge

Hi Pau,

On Wed, Oct 16, 2019 at 11:12:59AM +0000, pespun [REDMINE] wrote:

Traffic mode is already parsed into "as->cfg.mode" in "as_traf_mode_cmd". It is set by default to OSMO_SS7_AS_TMOD_LOADSHARE in osmo_ss7_as_find_or_create().

So far it seems out client code is submitting traffic mode = override in REG REQ messages, which probably changes the mode. That traffic mode is parsed and stored into as->cfg.mode in handle_rkey_reg().

Correct. It's just that nothing ever happens with that information, so we're in compliance to the spec in that you can configure different traffic-modes, rather than rejecting what is not supported.

Feel free to focus on the actual implementation of the load-sharing. I will try to find time today or tomorrow to work on a test case.

The normal case in the world out there is without RKM, i.e. the traffic mode is configured by the VTY / config file. RKM is specified by IETF but I haven't seen it in use anywhere outside of Osmocom. In either way, both methods provide a means to set the desired traffic-mode, the implementation of the related behavior is identical irrespective of the method of configuration.

#7 - 10/17/2019 11:47 AM - pespun

- % Done changed from 0 to 20

Hi Roch,

I didn't start with the IPA-side implementation yet, it's in the TODO list.

So far I have been implementing the round robin loadshare part in M3UA and making sure I can configure through VTY and use it with RKM in an osmocom setup with BSC<->STP<->MSC, which was the quickest way to have something to start with.

I'll now focus on statically-configured AS/ASP in osmo-stp for both M3UA and IPA.

Using RKM traffic mode configured in VTY fixed here:

<https://gerrit.osmocom.org/c/libosmo-sccp/+15803> ss7: Set configured AS traffic mode when sending Routing Key Register msg

Kind regards,

Pau

#8 - 10/18/2019 08:00 PM - laforge

Hi Pau,

I just force-pushed an update to STP_Tests.ttcn in the laforge/stp branch of osmo-ttcn3-hacks.git. It contains a test like TC_tmt_loadshare which should test functionality with traffic-mode load-share. I haven't tested the test yet, but I'll see if I can test it with a proprietary STP (Cisco ITP).

#9 - 10/18/2019 09:12 PM - laforge

- Precedes Feature #4218: dynamic IPA ASPs added

#10 - 10/29/2019 08:21 AM - laforge

- File loadshare.pcap added

I just re-tested with current laforge/stp. I tested two configurations:

- with traffic-mode loadshare set in the vty
- without any traffic-mode set in the VTY

The outcome is the same: The first of the two ASPs always gets all the traffic, despite "show" in the VTY showing two ASP_ACTIVE ASPs within an AS that has loadshare:

AS Name	State	Routing Context	Routing Dpc	Key Si	Key Opc	Cic Ssn	Cic Min	Cic Max	Traffic Mode
as-sender	AS_ACTIVE	1023	23						override
as-receiver	AS_ACTIVE	1042	42						loadshare

ASP Name	AS Name	State	Type	Remote IP	Addr:Rmt	Port	SCTP
asp-sender	?	ASP_ACTIVE	m3ua	127.0.0.1		9999	
asp-receiver0	?	ASP_ACTIVE	m3ua	127.0.0.1		10000	
asp-receiver1	?	ASP_ACTIVE	m3ua	127.0.0.1		10001	

pcap file attached

#11 - 11/05/2019 04:40 PM - laforge

- Checklist item [x] load-share in m3ua added
- Checklist item [x] tests for m3ua added
- Checklist item [] load-share in ipa added
- Checklist item [] tests for ipa added
- % Done changed from 20 to 50

#12 - 11/05/2019 04:48 PM - laforge

- Priority changed from Normal to Urgent

#13 - 11/07/2019 04:18 PM - pespin

- Checklist item [x] load-share in ipa set to Done
- Checklist item [x] tests for ipa set to Done
- % Done changed from 50 to 90

current osmo-stp master has been tested with TTCN3 STP_Tests_IPA.tcn from osmo-ttcn3-hacks.git branch laforge/stp and seems to support loadsharing fine on IPA ASs.

Only missing thing is to fix the TTCN3 hack required to have tests working in order to be able to merge the TTCN3 tests branch into master: <https://gerrit.osmocom.org/c/osmo-ttcn3-hacks/+/15821>

#14 - 11/07/2019 06:57 PM - pespin

- Status changed from In Progress to Feedback

libosmo-sccp.git fix to have TTCN3 tests working without the HACK patch available here:

<https://gerrit.osmocom.org/c/libosmo-sccp/+/16005> xua: ipa_asp_fsm: Allow receiving IPA ID ACK before IPA ID RESP

Once all commit are merged and related TTCN3 tests are green (should be after everything's merged) we can probably close this ticket.

#15 - 11/13/2019 04:48 PM - laforge

- Status changed from Feedback to Closed
- % Done changed from 90 to 100

all commits merged. tests passing. closed.

#16 - 08/27/2020 07:51 AM - laforge

- Related to Feature #2002: Implement xUA traffic modes for load-balance and broadcast added

Files

loadshare.pcap	7.99 KB	10/29/2019	laforge
----------------	---------	------------	---------