

## OsmoMGW - Feature #3429

### idea: auto-cleanup endpoints after long period of inactivity?

07/28/2018 07:18 PM - neels

<b>Status:</b> Resolved	<b>Start date:</b> 07/28/2018
<b>Priority:</b> Normal	<b>Due date:</b>
<b>Assignee:</b> osmith	<b>% Done:</b> 100%
<b>Category:</b>	
<b>Target version:</b>	
<b>Spec Reference:</b>	
<b>Description</b>	
<p>To avoid a scenario where osmo-mgw maintains open endpoints for a voice stream that long ceased without the mgw being told, we might want to automatically close down rtpbridge/* endpoints if we have seen no MGCP message and also not a single RTP package for a long time. A long time could be 10 minutes, or also as low as half a minute.</p> <p>What happens when osmo-bsc or osmo-msc crash hard with RTP streams open? Do osmo-bsc / osmo-mgw send a wildcard DLCX for all endpoints? Maybe not a good idea in case several osmo-bsc and/or osmo-msc share the same MGW?</p> <p>(Inspired by an IRC question about osmo-mgw saying "Not able to find a free endpoint")</p>	
<b>Related issues:</b>	
Related to OsmoMGW - Feature #3507: allow shorter Connection Identifier 'l:'	<b>Resolved</b> <b>08/28/2018</b>
Related to OsmoMGW - Feature #3508: compare Connection Identifier 'l:' case i...	<b>Resolved</b> <b>08/28/2018</b>
Related to OsmoMGW - Feature #3509: match MGCP "l:" Connection ID also when l...	<b>Resolved</b> <b>08/29/2018</b>
Related to OsmoMSC - Feature #3655: Introduce self-destruction timer for SS/U...	<b>Resolved</b> <b>10/16/2018</b>
Related to OsmoBSC - Feature #3659: handover during LCLS directly between BTSs	<b>Stalled</b> <b>10/17/2018</b>
Related to OsmoMGW - Feature #3783: Make conn-timeout compatible with LCLS	<b>Resolved</b> <b>02/06/2019</b>

## History

### #1 - 08/28/2018 06:05 PM - neels

Seeing issues like [#3507](#) and [#3508](#), where an MGCP client fails to clean up its own endpoint connections due to Connection Identifier mismatches (due to size limits or case insensitivity) makes me return to this issue. I see this as an important sanity feature, especially when the osmo-mgw is a long running entity serving various clients. At first this was only about clients failing to send DLCX messages, but also seeing us failing to accept DLCX that weren't forgotten at all makes me think that it is rather important that we magically clean up unused endpoint connections.

If endpoints that failed to be cleaned pile up, it will eventually exhaust all of the ports or the permitted number of endpoints, besides leaving unused state in osmo-mgw's memory.

I still think something like one minute of neither RTP nor MGCP messages received for a given endpoint connection is a sane limit to discard a connection automatically.

To be very conservative, I guess three minutes of permitted inactivity could be a good default configuration.

### #2 - 08/28/2018 06:05 PM - neels

- Related to Feature #3507: allow shorter Connection Identifier 'l:' added

### #3 - 08/28/2018 06:05 PM - neels

- Related to Feature #3508: compare Connection Identifier 'l:' case insensitively added

### #4 - 08/28/2018 06:17 PM - neels

RFC3435

#### 2.1.1.3.2 Names of Connections

Connection identifiers are created by the gateway when it is requested to create a connection. They identify the connection within the context of an endpoint. Connection identifiers are

treated in MGCP as hexadecimal strings. The gateway MUST make sure that a proper waiting period, at least 3 minutes, elapses between the end of a connection that used this identifier and its use in a new connection for the same endpoint (gateways MAY decide to use identifiers that are unique within the context of the gateway). The maximum length of a connection identifier is 32 characters.

Hah, I did say 3 minutes, didn't I, even though for a slightly different aim.

**#5 - 08/29/2018 12:11 AM - neels**

- Related to Feature #3509: match MGCP "l:" Connection ID also when leading zeros are omitted added

**#6 - 11/22/2018 11:04 AM - laforge**

- Assignee set to osmith

**#7 - 01/22/2019 03:00 PM - osmith**

- Status changed from New to In Progress

**#8 - 01/24/2019 10:26 AM - osmith**

- Related to Feature #3655: Introduce self-destruction timer for SS/USSD connections added

**#9 - 01/29/2019 03:30 PM - osmith**

- % Done changed from 0 to 20

Status update:

One osmo-mgw can have multiple endpoints (mgcp\_endpoint), and each endpoint can have multiple connections (mgcp\_conn, always RTP at this point, possibly other protocols in the future).

So far I've implemented a watchdog for mgcp\_conn. Next up is implementing another watchdog for mgcp\_endpoint.

**#10 - 01/31/2019 09:32 AM - msuraev**

- Related to Feature #3659: handover during LCLS directly between BTSs added

**#11 - 01/31/2019 09:34 AM - msuraev**

While implementing this we've got to make sure it doesn't break BTS-variant of LCLS.

I'm not sure if we can determine LCLS status in MGW so this should be optional feature off by default which can be enabled via vty with corresponding notice regarding LCLS incompatibility.

**#12 - 01/31/2019 04:27 PM - neels**

Hm, I didn't consider LCLS keeping inactive endpoints open for a long period of time.

Keeping unused endpoints open is still a bit of a problem, we need some sort of sanity there in the MGW.

[msuraevlaforge](#) Can LCLS tear down unused MGW endpoints, i.e., on breaking LCLS, can it re-create endpoints and transmit a new RTP port?

We could also tell the MGW that the endpoint is still in use by sending no-op MDCX to the MGW once a minute; or investigate for some other keep-alive message in MGCP.

[osmith](#) btw, for me, having the inter-MS-C GSUP messages is more urgent than this feature.

**#13 - 01/31/2019 06:10 PM - laforge**

On Thu, Jan 31, 2019 at 04:27:19PM +0000, [redmine@lists.osmocom.org](mailto:redmine@lists.osmocom.org) wrote:

[msuraevlaforge](#) Can LCLS tear down unused MGW endpoints, i.e., on breaking LCLS, can it re-create endpoints and transmit a new RTP port?

in theory one could write the LCLS code to do that, but of course it comes at a lot of extra complexity.

The "easy" solution would be ..

We could also tell the MGW that the endpoint is still in use by sending no-op MDCX to the MGW once a minute; or investigate for some other keep-alive message in MGCP.

exactly my thinking. The guard timer of the MGCP connect (not endpoint, right?) would be re-freshed not only by RTP but by any MGCP message related to that connection.

**#14 - 01/31/2019 06:20 PM - laforge**

Also, LCLS explicitly defines that the media path/plane remains active but is simply not used until LCLS instructs any of the network elements to do otherwise.

So definitely, all resources/ports/etc. should be allocated so that with no additional latency in response to only a single LCLS message the media can be re-activated.

Also, keep in mind that there are LCLS configurations that support "locally switched call until a RTP packet arrives again from the core network" at which point the local RTP will be dropped in favor of the RTP from the core.

**#15 - 02/01/2019 09:34 AM - osmith**

- % Done changed from 20 to 90

I had already created a patch, but forgot to post it here and update the status:

<https://gerrit.osmocom.org/#/c/osmo-mgw/+12730/>

In the current state, this patch disables the timeout by default and notes that it should not be enabled together with LCLS in the VTY config.

The guard timer of the MGCP connect (not endpoint, right?) would be re-freshed not only by RTP but by any MGCP message related to that connection.

It is already implemented like this in the patch, receiving MDCX will update the guard timer too.

We could also tell the MGW that the endpoint is still in use by sending no-op MDCX to the MGW once a minute; or investigate for some other keep-alive message in MGCP.

Sounds like a good solution to me, how about I create a new issue for that?

I think this issue is done when the patch gets its second +1.

**#16 - 02/06/2019 12:49 PM - osmith**

- Related to Feature #3783: Make conn-timeout compatible with LCLS added

**#17 - 02/06/2019 12:50 PM - osmith**

- Status changed from In Progress to Resolved

- % Done changed from 90 to 100