

## OsmoMGW - Bug #4554

### Clean up stale endpoints after OsmoBSC crash

05/16/2020 08:52 PM - ipse

|   |        |                    |            |
|---|--------|--------------------|------------|
| <b>Status:</b>  | New    | <b>Start date:</b> | 05/16/2020 |
| <b>Priority:</b>  | Normal | <b>Due date:</b>   |            |
| <b>Assignee:</b>  |        | <b>% Done:</b>     | 0%         |
| <b>Category:</b>  |        |                    |            |
| <b>Target version:</b>  |        |                    |            |
| <b>Spec Reference:</b>  |        |                    |            |
| <b>Description</b>  |        |                    |            |
| <p>When OsmoBSC crashes, OsmoMGW doesn't know about that and continues to maintain endpoints, allocated by the old OsmoBSC process forever.</p> <p>I see two possible ways to resolve this:</p> <ol style="list-style-type: none"><li>1) RTP activity timeout (a standard feature of many VoIP servers). If we don't see RTP activity on a certain endpoint, assume it dead, file an alarm to O&amp;M and deallocate resources. The trick here is that this might happen for various reasons even if OsmoBSC is live and kicking. So we should somehow let it know that the endpoint has failed. Otherwise, this is an important alarm in itself, besides handling crashed OsmoBSC.</li><li>2) Somehow maintain a connection between OsmoBSC and OsmoMGW. Maybe a form of a ping/pong with a random OsmoBSC ID. So that if OsmoBSC crashes and is restarted, it send a different ID and OsmoMGW knows it's time to deallocate endpoints allocated by the previous OsmoBSC instance.</li></ol> <p>PS I don't know MGCP but I would think it should have ways to handle such situations?</p> <p>PPS A workaround in the situation with a single OsmoMGW handling a single OsmoBSC is to always restart them simultaneously. So that if one crashes, the other one is forced to restart as well.</p> |        |                    |            |