

## OsmoBTS - Bug #4444

### ports not unique when running multiple osmo-bts on the same machine

03/08/2020 08:20 PM - laforge

<b>Status:</b> Resolved	<b>Start date:</b> 03/08/2020
<b>Priority:</b> Urgent	<b>Due date:</b>
<b>Assignee:</b> laforge	<b>% Done:</b> 100%
<b>Category:</b>	
<b>Target version:</b>	
<b>Spec Reference:</b>	
<b>Description</b>	
This is with running two osmo-bts[-virtual] on the same machine and a number of active voice calls:	
<pre>root@36bd59f48d7c:/docker# ./netstat -uan Active Internet connections (servers and established) Proto Recv-Q Send-Q Local Address           Foreign Address         State udp      768    0 172.18.23.4:16384       0.0.0.0:* udp      0      0 172.18.23.4:16384       0.0.0.0:* udp      0      0 172.18.23.4:16385       0.0.0.0:* udp      0      0 172.18.23.4:16385       0.0.0.0:* udp      768    0 172.18.23.4:16386       0.0.0.0:* udp      0      0 172.18.23.4:16386       0.0.0.0:* udp      0      0 172.18.23.4:16387       0.0.0.0:* udp      0      0 172.18.23.4:16387       0.0.0.0:* udp      0      0 172.18.23.4:16388       0.0.0.0:* udp      0      0 172.18.23.4:16388       0.0.0.0:* udp      0      0 172.18.23.4:16389       0.0.0.0:* udp      0      0 172.18.23.4:16389       0.0.0.0:* udp      0      0 172.18.23.4:16390       0.0.0.0:* udp      0      0 172.18.23.4:16391       0.0.0.0:* udp      0      0 172.18.23.4:16392       0.0.0.0:* udp      0      0 172.18.23.4:16393       0.0.0.0:* udp      0      0 172.18.23.4:16394       0.0.0.0:* udp      0      0 172.18.23.4:16395       0.0.0.0:* udp      0      0 172.18.23.4:16396       0.0.0.0:* udp      0      0 172.18.23.4:16397       0.0.0.0:*</pre>	
16384 is bts->rtp_port_range_start. So both of them happily start to bind ports from 16384 upwards without creating any error whatsoever. This means that the bind_rtp() loop in <a href="https://source.sr.common/rsl.c">source:src/common/rsl.c</a> is not working at all.	
<b>Related issues:</b>	
Related to OsmoBTS - Bug #4446: osmo-bts doesn't actually ever connect() its ...	<b>New</b> <b>03/08/2020</b>

### Associated revisions

#### Revision 54c919ec - 03/09/2020 10:24 AM - laforge

ortp: disable SO\_REUSEADDR + SO\_REUSEPORT

ortp >= 0.24.0 doesn't differentiate between SO\_REUSEADDR and SO\_REUSEPORT, and has both enabled by default. The latter means that we can end up with non-unique port bindings as we will not fail to bind the same port twice.

This should have caused visible problems not only when operating multiple osmo-bts on one machine (rare), but also with a single osmo-bts. Once the range (default 16384-17407) wraps, there is a risk of new sockets (for new calls) colliding with old ones. As two ports (RTP+RTCP) are used per call, this means every 512 voice calls we expect the BTS to wrap. And from that point onwards there's a risk of overlapping with previously allocated sockets.

Change-Id: I4fc9eee561c7958c70c63b4ffdc6cb700b795e28

Closes: OS#4444

## Revision fb3a57f7 - 03/09/2020 11:00 AM - laforge

ortp: disable SO\_REUSEADDR + SO\_REUSEPORT

ortp >= 0.24.0 doesn't differentiate between SO\_REUSEADDR and SO\_REUSEPORT, and has both enabled by default. The latter means that we can end up with non-unique port bindings as we will not fail to bind the same port twice.

This should have caused visible problems not only when operating multiple osmo-bts on one machine (rare), but also with a single osmo-bts. Once the range (default 16384-17407) wraps, there is a risk of new sockets (for new calls) colliding with old ones. As two ports (RTP+RTCP) are used per call, this means every 512 voice calls we expect the BTS to wrap. And from that point onwards there's a risk of overlapping with previously allocated sockets.

Change-Id: I4fc9eee561c7958c70c63b4ffdc6cb700b795e28  
Closes: OS#4444

## History

---

### #1 - 03/08/2020 08:31 PM - laforge

It turns out that libortp uses not only SO\_REUSEADDR, but actually also SO\_REUSEPORT since 0.24.0 in 2015. This means that any attempt to bind multiple sockets to the same local port will succeed, and the logic to bind to a unique port in osmo-bts is completely rendered useless.

This should have caused visible problems not only when operating multiple osmo-bts on one machine (rare), but also with a single osmo-bts. Once the range (default 16384-17407) wraps, there is a risk of new sockets (for new calls) colliding with old ones. As two ports (RTP+RTCP) are used per call, this means every 512 voice calls we expect the BTS to wrap. And from that point onwards there's a risk of overlapping with previously allocated sockets.

### #2 - 03/08/2020 09:44 PM - laforge

- % Done changed from 0 to 80

- test case: <https://gerrit.osmocom.org/c/libosmo-abis/+/17429>
- fix: <https://gerrit.osmocom.org/c/libosmo-abis/+/17430>

### #3 - 03/08/2020 09:50 PM - laforge

- Related to Bug #4446: osmo-bts doesn't actually ever connect() its RTP/UDP sockets added

### #4 - 06/13/2020 05:58 PM - laforge

- Status changed from In Progress to Resolved

- % Done changed from 80 to 100

patches merged back in march