

## OsmoMGW - Bug #3731

### TTCN3 TC\_rtpem\_selftest randomly fails

12/13/2018 12:37 PM - daniel

<b>Status:</b>	New	<b>Start date:</b>	12/13/2018
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	daniel	<b>% Done:</b>	0%
<b>Category:</b>			
<b>Target version:</b>			

#### Description

Every once in a while the osmo-mgw TTCN3 test TC\_rtpem\_selftest fails with "Received unexpected type from RTP"

Here are the relevant lines from the log:

```
06:33:37.398872 75 RTP_Emulation.ttcn:341 entering f__IPL4__PROVIDER__listen: 127.0.0.1:10000 / UD
P
06:33:37.399121 75 RTP_Emulation.ttcn:349 entering f__IPL4__PROVIDER__listen: 127.0.0.1:10001 / UD
P
06:33:37.399174 75 RTP_Emulation.ttcn:357 Replied on CTRL to mtc @RTP_Emulation.RTPEM_bind : { loc
al_port := 10000 }
[...]
06:33:37.399487 76 RTP_Emulation.ttcn:341 entering f__IPL4__PROVIDER__listen: 127.0.0.2:20000 / UD
P
06:33:37.399810 76 RTP_Emulation.ttcn:349 entering f__IPL4__PROVIDER__listen: 127.0.0.2:20001 / UD
P
06:33:37.399866 76 RTP_Emulation.ttcn:357 Replied on CTRL to mtc @RTP_Emulation.RTPEM_bind : { loc
al_port := 20000 }
[...]
06:33:37.400095 75 RTP_Emulation.ttcn:365 entering f__IPL4__PROVIDER__connect: 127.0.0.1:10000 ->
127.0.0.2:20000 / UDP
06:33:37.400123 75 RTP_Emulation.ttcn:373 entering f__IPL4__PROVIDER__connect: 127.0.0.1:10001 ->
127.0.0.2:20001 / UDP
06:33:37.400139 75 RTP_Emulation.ttcn:381 Replied on CTRL to mtc @RTP_Emulation.RTPEM_connect : {
}
[...]
06:33:37.400343 76 RTP_Emulation.ttcn:365 entering f__IPL4__PROVIDER__connect: 127.0.0.2:20000 ->
127.0.0.1:10000 / UDP
06:33:37.400368 76 RTP_Emulation.ttcn:373 entering f__IPL4__PROVIDER__connect: 127.0.0.2:20001 ->
127.0.0.1:10001 / UDP
06:33:37.400384 76 RTP_Emulation.ttcn:381 Replied on CTRL to mtc @RTP_Emulation.RTPEM_connect : {
}
[...]
06:33:37.421230 75 RTP_Emulation.ttcn:303 Sent on RTP to system @RTP_CodecPort.RTP_Send : { connId
:= 1, msg := { rtp := { version := 2, padding_ind := '0'B, extension_ind := '0'B, CSRC_count := 0
, marker_bit := '0'B, payload_type := 0, sequence_number := 0, time_stamp := '000000000000000000
0000000000000'B, SSRC_id := '11011110101011011011011111011111'B, CSRCs := omit, ext_header := omit,
data := '01020304'O } } }
06:33:37.421328 76 RTP_Emulation.ttcn:303 Sent on RTP to system @RTP_CodecPort.RTP_Send : { connId
:= 1, msg := { rtp := { version := 2, padding_ind := '0'B, extension_ind := '0'B, CSRC_count := 0
, marker_bit := '0'B, payload_type := 0, sequence_number := 0, time_stamp := '000000000000000000
0000000000000'B, SSRC_id := '11011110101011011011011111011111'B, CSRCs := omit, ext_header := omit,
data := '01020304'O } } }
06:33:37.421404 75 RTP_Emulation.ttcn:303 Outgoing message was mapped to @IPL4asp_Types.ASP_Send :
{ connId := 1, proto := { udp := { } }, msg := '8000000000000000DEADBEEF01020304'O }
06:33:37.421443 76 RTP_Emulation.ttcn:303 Outgoing message was mapped to @IPL4asp_Types.ASP_Send :
{ connId := 1, proto := { udp := { } }, msg := '8000000000000000DEADBEEF01020304'O }
06:33:37.421561 76 RTP_Emulation.ttcn:462 Start timer T_transmit: 0.02 s
06:33:37.421578 75 RTP_Emulation.ttcn:303 Message enqueued on RTP from system @Socket_API_Definiti
ons.PortEvent : { result := { errorCode := ERROR_SOCKET (4), connId := 1, os_error_code := 1, os_e
rror_text := "Operation not permitted" } } id 1
06:33:37.421631 75 RTP_Emulation.ttcn:462 Start timer T_transmit: 0.02 s
```

```
06:33:37.421660 75 RTP_Emulation.ttcn:439 Matching on port RTP failed: Type of the first message i
n the queue is not @RTP_CodecPort.RTP_RecvFrom.
06:33:37.421674 75 RTP_Emulation.ttcn:470 Matching on port RTP succeeded.
06:33:37.421689 75 RTP_Emulation.ttcn:470 Receive operation on port RTP succeeded, message from sy
stem(): @Socket_API_Definitions.PortEvent: { result := { errorCode := ERROR_SOCKET (4), connId :=
1, os_error_code := 1, os_error_text := "Operation not permitted" } } id 1
06:33:37.421705 75 RTP_Emulation.ttcn:470 Message with id 1 was extracted from the queue of RTP.
06:33:37.421723 75 RTP_Emulation.ttcn:471 setverdict(fail): none -> fail reason: "Received unexpec
ted type from RTP", new component reason: "Received unexpected type from RTP"
```

Both sockets get bound and connected to each other, but the send on one of them then fails with -EPERM.

Not sure why this is going on here. It seems that trying to send a packet that is dropped through firewalling rules will result in -EPERM, but since this only happens once in a while I don't think that's it.

Maybe the fact that both sockets are on the same machine and trying to send to each other at the same time has something to do with it?