

## libosmo-netif - Bug #3006

### Osmux: improvements on lost/dropped RTP packets

02/26/2018 12:48 PM - pespin

|                        |     |                    |            |
|------------------------|-----|--------------------|------------|
| <b>Status:</b>         | New | <b>Start date:</b> | 02/26/2018 |
| <b>Priority:</b>       | Low | <b>Due date:</b>   |            |
| <b>Assignee:</b>       |     | <b>% Done:</b>     | 0%         |
| <b>Category:</b>       |     |                    |            |
| <b>Target version:</b> |     |                    |            |
| <b>Spec Reference:</b> |     |                    |            |

#### Description

During recent analysis of Osmux related code, a few issues with handling of lost RTP packets were found. See OS#2439 [1] and OS#2440 [2] for more information.

The summary: In some of the cases in which an RTP packet (or a big amount of consecutive ones) is lost, Osmux will currently fail to account correctly about it and seq/timestamp won't be updated correctly when re-generating RTP packets on the other side, which would make seq/timestamps to be shifted and from RTP receiver point of view than can be seen as a constant increment in delay every time there's this issue. I'm in the process of improving some related code to fix the issue mentioned above (OS-[#2440](#)).

As part of this process, I created OS#2439 which attempts to potentially improve the quality of the audio during this scenarios in which RTP audio is lost. Currently, when an input RTP packet is detected to be missing (gap in seq), Osmux still allocates a slot for it and then clones there the payload of a packet received before/after the lost packet, and on the receiver side, Osmux doesn't know how to differentiate between lost/replicated packets or good ones. This means instead of dropping the replicated payload, it is sent to the RTP receiver. This is bad because we don't let the RTP receiver take smarter decisions on how to fix the loss of the packet.

To fix this issue, I can foresee 2 solutions, each one with pros and cons.

#### Solution 1:

When a loss of RTP packet is detected in RTP->Osmux converter, we still allocate an AMR frame for it but we fill it with 0 instead of copying the payload from some received payload nearby, as voice frames cannot be all-zeros. On the Osmux->RTP converter (receiver side), we check if the packet is all 0 and in that case we drop the payload and increment seq/timestamp of RTP sender, this way the RTP receiver accounts the RTP packet as lost correctly.

#### pros:

- No major protocol compatibility issues with older versions of Osmux.

#### cons:

- 17 bytes (AMR 5.9) are still sent over the sat link every time an RTP packet is lost.
- CPU power consumption increases on the receiver quite a lot as we need to compare every payload against a string of zeroes.

#### Solution 2:

A new field of 8 bits is added to the Osmux frame header. This field contains a bitmask, where each bit states whether there's an AMR payload allocated for it (1), or not (0).

On the Osmux->RTP converter (receiver side), we check if a payload has its bit set to 0 and in that case we increment seq/timestamp of RTP sender and we send nothing, this way the RTP receiver accounts the RTP packet as lost correctly.

#### pros:

- Less CPU power used: We avoid memcpying in the sender for each lost RTP packet, and we avoid memcmp() on all Osmux payloads.

#### cons:

- 1 byte is sent over the sat link with every Osmux frame.
- It breaks compatibility with current versions of the protocol, which means they cannot talk together and all systems must be upgraded to use the same version (or have different bsc-nat for each version).

As far as I understand, Solution 1 sends less data over the sat link as long as the ratio of lost RTP packets is lower than  $1/17 = 5.88\%$ . If this threshold is normally reached, Solution 2 saves more data. Does ONW have any measurements on regular packet

lost?

PS: In case someone wants to look at some reference on the protocol: <http://ftp.osmocom.org/docs/latest/osmux-reference.pdf>

[1] <https://osmocom.org/issues/2439>

[2] <https://osmocom.org/issues/2440>

---

## History

#1 - 09/30/2018 08:34 AM - laforge