

osmo-gbproxy - Bug #4889

implement truncating of BSSGP STATUS when exceeding the FR MTU

12/06/2020 07:44 PM - laforge

Status: Resolved	Start date: 12/06/2020
Priority: Normal	Due date:
Assignee: daniel	% Done: 100%
Category:	
Target version:	
Spec Reference: 48.018 10.4.14	
Description	
<p>It might be that we receive a BVC-STATUS on the IP side with a UDP packet size that exceeds the FR MTU, at which point we need to truncate it (see "NOTE" in 48.018 10.4.14).</p> <p>The same can also occur when generating a BVC-STATUS in response to a BVC message received on a FR link. Let's assume we e.g. receive a UL-UNITDATA that's exactly the FR MTU of 1600 for a non-existent BVCI. Then the generated BSSGP STATUS must truncate the "PDU IN Error" IE accordingly.</p> <p>To make all of this work, we have to "learn" the MTU from the underlying network device (we receive it together with the link status event messages but currently ignore it). Plus we then need to use this value when generating STATUS or even when passing on STATUS</p>	
Related issues:	
Related to osmo-gbproxy - Feature #4472: Intra-domain connection of OsmoGBPRO... In Progress 03/29/2020	

Associated revisions

Revision 44fa2018 - 02/12/2021 09:39 AM - daniel

gbproxy: Use bssgp2_nsi_tx_ptp in gbprox_relay2nse

Use the function provided by bssgp2 instead of setting up the ns2 prim request ourself.

Related: OS#4889

Change-Id: I0b8926eb903ed972edb2ed7ba3edbb3d77889564

Revision 1ff86f7c - 02/12/2021 02:19 PM - daniel

bssgp_bvc_fsm: Set/get maximum BSSGP PDU length

Add functions to get/set the maximum supported BSSGP PDU size by the NS layer.

IPv4 and IPv6 should not matter since we can just enable IP fragmentation and send NS PDUs up to $2^{16} +$ bytes. Frame relay does not support fragmentation and this is the reason we need to be aware of the maximum PDU size. Luckily with 1600 bytes the MTU in frame relay can hold a regular IP packet including NS/BSSGP overhead.

On the NS layer this corresponds to the size of an NS SDU in NS-UNITDATA (3GPP TS 48.016 Ch. 9.2.10)

Change-Id: I9bb82ead27366b7370c9ff968e03ca2113ec11f0

Related: OS#4889

Revision fa632b8e - 02/12/2021 02:19 PM - daniel

bssgp2_enc_status: Truncate STATUS message to maximum PDU length

Related: OS#4889

Change-Id: Ic39d918c56399ceb0431299ce938e3bf276f678a

Revision 4f1128fc - 02/16/2021 03:29 PM - lynxis

gprs_ns2: inform the NS user (BSSGP) about the MTU of a NSE

The BSSGP layer needs to know the MTU of the NS UNIDATA payload.
The MTU can be 0 if the NSE doesn't contain any NSVC.
Every status indication will contain the mtu value.
The MTU in the status indication contains the maximum transfer unit of a BSSGP message. From NS side the maximum SDU.

Related: OS#4889
Change-Id: I5016b295db6185ec131d83089cf6c806e34ef1b6

Revision a8b61659 - 02/16/2021 03:39 PM - daniel

Add SDU length for an NSE (== BSSGP PDU size)

Prepare tracking the SDU from NS. Initialize with a conservative default.
The value is not yet updated, that will happen in a later patch.

Related: OS#4889
Depends: I5016b295db6185ec131d83089cf6c806e34ef1b6 (libosmocore.git)
Depends: I9bb82ead27366b7370c9ff968e03ca2113ec11f0 (libosmocore.git)
Change-Id: Ic1080abde942ec5a2ae7cdee0ffe716a2fbddb1e

Revision f8cba650 - 02/16/2021 03:42 PM - daniel

gbproxy: Use bssgp2_enc_status when sending STATUS

bssgp_tx_status() is not aware of the MTU and cannot truncate the PDU if needed. Use the newer bssgp2_enc_status() which supports truncating the PDU.

Related: OS#4889
Depends: Ic39d918c56399ceb0431299ce938e3bf276f678a (libosmocore.git)
Change-Id: Id5ddb10385655b339b2a4f04651c1da09b3efb62

Revision 3de1cb0d - 02/19/2021 10:41 AM - lynxis

gprs_ns2_fr: pass MTU changes to the NSE

When the MTU of the frame relay device changes, update the bind and notify all NSEs.

Related: OS#4889
Change-Id: I946f7655c9526ffd98dabdce219c6a419b71e00c

Revision cf1fa635 - 02/19/2021 10:41 AM - lynxis

gprs_ns2: truncate the NS_STATUS to the MTU

A NS Status can contain the original NS message which might result in a NS PDU which exceeds the MTU of the NS-VC.
Truncate the original message to the maximum possible.
Based on truncate BSSGP status message.

Related: OS#4889
Change-Id: I35d8f8bf0eae890f4db56423da0b23b638d24311

Revision 38b9c9a9 - 03/10/2021 12:14 PM - daniel

Update the max_sdu_len from NS

Related: OS#4889
Change-Id: Ie26550198db0cc34ca0a882c137c8685a5662f2a

History

#1 - 12/08/2020 12:11 PM - Iaforge

- Related to Feature #4472: Intra-domain connection of OsmoGBPROXY to multiple SGSNs (pooling) added

#2 - 01/20/2021 04:20 PM - daniel

- Description updated

- Status changed from New to In Progress

#3 - 01/20/2021 05:36 PM - daniel

Some more questions came up while discussing this with lynxis:

- What happens if we receive a BSSGP message, but it is larger than the MTU on the destination NSE? Return with status?
- Do we need to dynamically change the MTU, how should we deal with NSEs where different NS-VCs have different MTUs (think VLAN interface vs. plain ethernet)? Report the lowest?
- What if the MTU changes because one NS-VC with a different MTU became (un)available? We need to update the BSSGP MTU on the fly.

#4 - 01/20/2021 06:50 PM - daniel

Some more observations:

gbproxy still uses a mixture of bssgp and bssgp2/bssgp_fsm code. For status bssgp_tx_status is used which is completely unaware of the BSSGP FSMs.

I'm not sure if and when we want to use osmo primitives in bssgp2, but this would probably be a bigger refactor which we don't have time for right now.

So right now I would make the bssgp_fsm mtu-aware and add a REQ_STATUS event to the BSSGP fsm which then sends a (truncated) STATUS through its BVC. This would then be used by the gbproxy

#5 - 01/21/2021 08:41 AM - daniel

- % Done changed from 0 to 10

Relevant comment from <https://gerrit.osmocom.org/c/libosmocore/+/22343>

also here I'm not sure if we really should worry about the MTU of the ethernet.

In the end, we have potentially larger BSSGP frames and there is no way to influence the size of the upper layer frames. Think of a PDP context with MTU 1500 (or close to that), plus the LLC, SNDCP, BSSGP overhead -> boom.

Yes, we may end up generating IP fragments. But then, the bandwidth of GPRS is ultra low, so what do we care about some more packets in the core network over wired interfaces that likely have more than a thousand times more bandwidth than our radio interface.

If we enforce staying within the MTU of the ethernet device, we would start dropping packets with no way to inform this up the protocol stack so that the LLC/SNDCP XID exchange could negotiate smaller packet sizes. - i.e. we'd effectively break the network completely.

I think the only situation where the MTU matters is in the case of FR, where we have a fixed, well-known MTU and no support from the transport (FR) to do segmentation / fragmentation by itself. Luckily it's 1600, and hence we do have some room for BSSGP/LLC/SNDCP overhead

Since we can fragment in IP the MTU should only ever matter when using frame-relay. There are not NSEs that have mixed II-types (e.g. FR/GRE and FR) so the MTU of an NSE should never change during its lifetime.

#6 - 02/12/2021 04:16 AM - daniel

- % Done changed from 10 to 30

There is an update for support from NS <https://gerrit.osmocom.org/c/libosmocore/+/22343> and I started working on BSSGP support.

See

[https://gerrit.osmocom.org/q/topic:%22os%25234889%22+\(status:open%20OR%20status:merged\)](https://gerrit.osmocom.org/q/topic:%22os%25234889%22+(status:open%20OR%20status:merged))

for the list of patches

#7 - 03/10/2021 02:44 PM - daniel

- Status changed from In Progress to Resolved

- % Done changed from 30 to 100

Fixed with [38b9c9a9](#)