OsmoGGSN (former OpenGGSN) - Bug #4434
sgsnemu: Fix wrong assumption always Interface Identifier == IPv6 public prefix

osmo-ggsn currently (20539f02713fc5afaf8f2248555d4a199e57cda98) uses same 64 bits value in the Interface Identifier (IPv6 GTP EUA sent during Create Pdp Ctx Response) as well as later for the IPv6 prefix (during Router Advertisement). That's really an implementation details and it doesn't necessarily need to be the case as per spec (they can have totally different values). For instance, ergw uses a constant Interface identifier of ::1 for all UEs.

sgsnemu currently expects the Interface Identifier to be the same as the IPv6 public prefix later received, which means it won't work properly when used against ergw (on IPv6).

Probably this diff for osmo-ggsn allows quickly running into a similar scenario in sgsnemu (receiving different values):

diff --git a/ggsn/icmpv6.c b/ggsn/icmpv6.c
index 12119b8..240c89c 100644
--- a/ggsn/icmpv6.c
+++ b/ggsn/icmpv6.c
@@ -142,6 +142,7 @@
 struct msgb *icmpv6_construct_ra(const struct in6_addr *saddr,
 ra_opt_pref->preferred_lifetime = htonl(GGSN_AdvPreferredLifetime);
 ra_opt_pref->res2 = 0;
+ ra_opt_pref->prefix[2] ^= 0xff;
 /* checksum */
 skb_csum = csum_partial(msgb_data(msg), msgb_length(msg), 0);

Proposed solution:
Disable handling of routing advertisement by the kernel (SLAAC) and do it within sgsnemu [1].

This would work similar to what we already do in osmo-ggsn, but the other way around:

- Upon IPv6 packet received in GTP tunnel from SGSN, where normally we'd then forward it to the tun device (encaps_tun()), instead we pass it to handle_router_mcast() (tun.c).
- In handle_router_mcast(), if what we received is a RouterSolicitiation, parse it and send back on the GTP tunnel a RouterAdvertisement: gtp_data_req(icmpv6_construct_ra())

So in sgsnemu case, the steps required:

1. Have a working setup with sgsnemu and osmo-ggsn
2. Trigger the issue by applying the patch above mentioned to study the behavior
3. Disable SLAAC (RouterAdvertisement) in the tun interface:
   1. Add a proc_ipv6_conf_write() function similar to the alreaday available proc_ipv6_conf_read()
   2. Redo the checks on "accept_ra" and "forwarding" proc properties in create_pdp_conf().
4. Send a RouterSolicitiation over the GTP tunnel using the IPv6 EUA received in PDPCreateContextResponse (create_pdp_conf()).
5. Wait for RouterAdvertisement coming from the GTP tunnel, then set the IP address prefix from there (cb_tun_ind()).

Extra: This allows us to implement IPv6 builtin ping in sgsnemu (like we do already for IPv4) without requiring the tun iface.


History
WIP in osmo-ggsn.git branch pespin/sgsnemu.

I already disabled kernel automatic SLAAC and I have sending RouterSolicitation and received RouterAdvertisement working (parsing the prefix bits still missing).

Related spec: [https://tools.ietf.org/html/rfc4861#section-4.2](https://tools.ietf.org/html/rfc4861#section-4.2)

Also interesting and related to the topic: [1] 3GPP TS 29.061 sec 11.2.1.3.2 "Ipv6 Stateless Address Autoconfiguration"

ETSI ETSI TS 129 061 V15.5.0 (2019-04)3GPP TS 29.061 version 15.5.0 Release 151) After the first phase of setting up Ipv6 access to an Intranet or ISP, the MS shall use the ipv6 Interface-Identifier, as provided by the GGSN, to create its Ipv6 Link-Local Unicast Address according to RFC 4291 [82]. Before the MS can communicate with other hosts or Mses on the Intranet/ISP, the MS must obtain an Ipv6 Global or Site-Local Unicast Address. The simplest way is the Ipv6 Stateless Address Autoconfiguration procedure described below and in 3GPP TS 23.060 [3]. The procedure is consistent with RFC 4862 [83].

2) After the GGSN has sent a Create PDP Context Response message to the SGSN, it shall start sending Router Advertisements periodically on the new MS-GGSN link established by the PDP Context. The MS may issue a Router Solicitation directly after the user plane establishment. This shall trigger the GGSN to send a Router Advertisement immediately.

The Prefix sent in the Router Advertisements shall be identical to the Prefix returned in the Create PDP Context Response

3) When creating a Global or Site-Local Unicast Address, the MS may use the Interface-Identifier received during the PDP Context Activation phase or it may generate a new Interface-Identifier. There is no restriction on the value of the Interface-Identifier of the Global or Site-Local Unicast Address, since the Prefix is unique. Interface-Identifiers shall in any case be 64-bit long.

Since the GGSN guarantees that the Prefix is unique, the MS does not need to perform any Duplicate Address Detection on addresses it creates. That is, the ‘DupAddrDetectTransmits’ variable in the MS should have a value of zero. If the MS finds more than one Prefix in the Router Advertisement message, it shall only consider the first one and silently discard the others. The GGSN shall not generate any globally unique Ipv6 addresses for itself using the Prefix assigned to the MS in the Router Advertisement.

[1] [https://www.etsi.org/deliver/etsi_ts/129000_129099/129061/15.05.00_60/ts_129061v150500p.pdf](https://www.etsi.org/deliver/etsi_ts/129000_129099/129061/15.05.00_60/ts_129061v150500p.pdf)

Several commits with improvements and fixes have already been merged.

I still have to 2 important commits WIP implementing related features:

- branch pespin/sgsnemu: Implements all manual SLAAC. afaict almost everything is working. There's one blocker: netdev_addroute6() I implemented (to add the default route towards the tun) always returns "errno=22/Invalid argument ioctl(SIOCADDRT)". I tried for a while changing a bit some parameters but I always get the same.
- branch pespin/sgsmeu-ping: Contains an extra commit on top of pespin/sgsnemu implementing ping for IPv6 APNs without requiring a tun interface. Sending ping messages is working, I still need to implement reading the pong messages. And I actually need to use the link-global IP address from SLAAC rather than the one from the PDP EUA.

On Tue, Apr 14, 2020 at 07:02:42PM +0000, pespin [REDMINE] wrote:

- branch pespin/sgsnemu: Implements all manual SLAAC. afaict almost everything is working. There's one blocker: netdev_addroute6() I implemented (to add the default route towards the tun) always returns "errno=22/Invalid argument ioctl(SIOCADDRT)". I tried for a while changing a bit some parameters but I always get the same.
have you tried to add the route via RTNL (routing netlink)? libnl-route might be used for it.

#7 - 04/15/2020 02:43 PM - pespin
- % Done changed from 70 to 80

The SIOCADDRT issue for AF_INET6 is solved now. Apparently it requires the ifindex to be set to the tun interface, leaving it as 0 doesn't work. Strangely that is not needed for AF_INET (IPv4) routing....

Related patches:

remote:   https://gerrit.osmocom.org/c/osmo-ggsn/+/17812 sgsnemu: Avoid adding extra autogenerated local link ipv6 addr to tun iface
remote:   https://gerrit.osmocom.org/c/osmo-ggsn/+/17817 sgsnemu: Fix ping transmitted statistics output
remote:   https://gerrit.osmocom.org/c/osmo-ggsn/+/17819 icmpv6.c: Mark internal function as static
remote:   https://gerrit.osmocom.org/c/osmo-ggsn/+/17820 sgsnemu: Get rid of duplicated options.destaddr
remote:   https://gerrit.osmocom.org/c/osmo-ggsn/+/17821 sgsnemu: Get rid of duplicated options.net
remote:   https://gerrit.osmocom.org/c/osmo-ggsn/+/17822 sgsnemu: tun_addaddr: Don't set local addr as dstaddr
remote:   https://gerrit.osmocom.org/c/osmo-ggsn/+/17823 icmpv6.c: Move code generating ipv6 hdr to its own function
remote:   https://gerrit.osmocom.org/c/osmo-ggsn/+/17824 Rename netdev_*route to end in route4
remote:   https://gerrit.osmocom.org/c/osmo-ggsn/+/17825 sgsnemu: Handle IPv6 SLAAC in tun iface manually
remote:   https://gerrit.osmocom.org/c/osmo-ggsn/+/17826 sgsnemu: Implement ping on IPv6 APNs

#8 - 04/15/2020 06:02 PM - pespin
- % Done changed from 80 to 90

And after all new features required, the actual fix for the actual issue is here:
https://gerrit.osmocom.org/c/osmo-ggsn/+/17828 sgsnemu: Fix assumption ipv6 Interface-Identifier of public addr == ...

#9 - 04/22/2020 09:38 AM - pespin
- Status changed from In Progress to Feedback

Patches merged. Waiting for user who internally reported the issue to test it before closing the ticket.