OsmoGGSN (former OpenGGSN) - Feature #2418
IPv6 PDP context support
08/01/2017 01:54 PM - laforge

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>High</td>
</tr>
<tr>
<td>Assignee:</td>
<td>dexter</td>
</tr>
<tr>
<td>Category:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
<tr>
<td>Spec Reference:</td>
<td></td>
</tr>
</tbody>
</table>

**Description**
Even while the transport (outer) layer is still v4-only, it would be great to finally have IPv6 PDP context (inner) layer support.

There is a user-contributed patch at [https://gerrit.osmocom.org/#/c/2870/](https://gerrit.osmocom.org/#/c/2870/)

This patch needs to be tested if it actually works at all (dexter) and the code review related modifications need to be made.

dexter: I’ve already done a first round of 12 comments on June 11, so after the testing you could also go through those, address them and re-submit a new version of the patch. I’ll do another round after the testing has concluded and the old comments are adressed.

**History**

#1 - 08/02/2017 12:22 PM - laforge
- Status changed from New to In Progress
- Assignee changed from dexter to laforge

the patch is against a >= 10 years old openggsn version, will require lots of manual porting work :/

#2 - 08/02/2017 11:37 PM - laforge
- % Done changed from 0 to 70

remote:   https://gerrit.osmocom.org/3403 ippool: Add IPv6 support to IP pool implementation
remote:   https://gerrit.osmocom.org/3404 lib/tun.h: Remove non-endian-safe redefinition of IP header
remote:   https://gerrit.osmocom.org/3405 ippool_new(): const-ify input arguments
remote:   https://gerrit.osmocom.org/3406 IPv6 support for user IP
remote:   https://gerrit.osmocom.org/3407 ggsn: Send proper errors in create_context_ind()
remote:   https://gerrit.osmocom.org/3408 Support setting TUN device IPv6 address + prefix

#3 - 08/03/2017 12:30 PM - dexter
- File ipv6_pdp_context_reject.pcapng added

I have configured the ggsn as suggested in:

debug
fg
listen 127.0.0.1
net 2001:780:44:2000:0:0:0:0/120

First I expected a problem with the sgsn, since this is the untested component, but it may also be a problem with the ggsn as well. The debug output of openggsn shows:

<0000> ippool.c:454 MS requested unsupported PDP context type
<0002> ggsn.c:202 Cannot allocate IP address in pool

This means that the ggsn is the entity that rejects the pdp context.
The attached trace also confirms that.

The sgsn log also displays the reject:
I wonder what could be wrong here. I have tried the sgsn emulator, but I can only get it to a point where it requests an IPv4 pdp context (which also gets rejected, which is expected with the current configuration). I start sgsnemu as follows:

```
./sgsnemu --l 127.0.0.2 -r 127.0.0.1
```

Maybe there is commandline argument missing. commandline argument?

I suspect either a problem with my ggsn configuration. It might also be that the pdp context request that the sgsn generates might have some issues.

#4 - 08/03/2017 10:41 PM - laforge
It was working once but I broke it later in some cleanup :/

I've just (force-)pushed a fix to laforge/ipv6, please retry using the new version.

The v6 activation has now been tested against my brand-new TTCN-3 testcase for PDP context activation, which supports both v4 and v6 contexts. I’ll expand on this and hopefully we can integrate this with Jenkins soon.

Regards,
Harald

#5 - 08/04/2017 05:39 PM - dexter
- File trace_v6pdpctx_act_and_deact_04072017.tar added
I gave it another try. I have tested it with IPV4 and IPV6. For IPV4 everything is fine so far. But IPV6 seems not to work.

The IPV6 pdpd context is successfully established, but it is not possible to transfer any data. I would have expected that at least pinging the phone is possible. There is data flowing between the phone and the network, but as I can see so far by the log its only Rx UPLINK-UNITDATA. There is no TX data.

On the tun0 interface, as well as on the GTP link, the only messages that appear are Router Solicitation and Neighbor Solicitation. When I look at the GB interface I see the messages there as well, so these messages seem to be mobile originated. However, I do not understand why source and destination address is does not fall in the assigned range. When I look at the pdp context ack. I can see the expected address is assigned.

The behaviour is cyclic, the phone establishes a pdp context, for a short time, data is transferred. Then the pdp context is deactivated. After some time the same happens again.

I have also checked with the communicator again. The communicator now shows also the same behaviour. I also tried with a qualcomm modem. The same there too.

Trades, logs and configs can be found in:
trace_v6pdpctx_act_and_deact_04072017.tar

regards,
Philipp

#6 - 08/05/2017 09:15 AM - laforge
It's good to see that the signaling plane seems to work end-to-end, i.e.
the v6 PDP context is established successfully and a dynamic IPv6 address is allocated.

The only odd part I can see on the signaling/control plane is the fact that the PCO contain 0.0.0.0 as DNS server IP adresses. This is odd in two ways:

a) if there are no DNS servers, why report 0.0.0.0 instead of simply not sending any DNS Server PCO back to the MS
b) if the PDP context is IPv6, the DNS Server addresses inherently also must be IPv6. The MS has no way to talk to an IPv4 DNS server if it has an IPv6 (only) PDP context

The MS then subsequently sends router and neighbor discovery requests which are unanswered. The big question is how that is supposed to work on a point-to-point interface anyway. In IPv4 there is no broadcast/ARP involved on such interfaces.

I'll read up with the relevant specs and will extend the TTCN-3 tests to transmit the exact same router and neighbor discovery requests to be able to develop and test without having to set up the entire cellular network + phone for each attempt.

#7 - 08/05/2017 03:15 PM - laforge
Ok, I found Section 11.2.1.3.2 of 3GPP TS 29.061 which details the process. For some strange reason, they decided to make it significantly more complex than in the IPv4 case, while at the same time making it not entirely compatible/compliant with what IETF IPv6 usually does.

Other apparently related documents: RFC3314, RFC7066, RFC6459, TS 23.060 9.2.1

My understanding at this point:

- the IPv6 address sent in the EUA IE of the PDP CTX ACT ACK from the GGSN is not used for actual traffic, but is only used for allocating an "interface identifier" to the MS/UE. This "interface identifier" is formed from the lower 64 bits of the prefix, while the upper 64 bits are being discarded.

- the MS/UE uses RFC2373 Section 2.5.8 to construct a link-local unicast address from this "interface identifier". This means: * lower 64 bits are the "interface identifier" * upper 64 bits are fe80:0000:0000:0000

- the MS may send a "router solicitation" icmpv6 message to trigger the GGSN to send a router advertisement to the MS (optional)
The GGSN sends a Router Advertisement message. The Router Advertisement messages shall contain the same prefix as the one provided in the PDP CTX ACT ACK. A given prefix shall not be advertised on more than one PDP context on a given APN, or set of APNs, within the same addressing scope. The GGSN shall be configured to advertise only one prefix per PDP context.

After the MS has received the Router Advertisement message, it constructs its full IPv6 address by concatenating the interface identifier received in PDP CTX ACT ACK, or a locally generated interface identifier, and the prefix received in the Router Advertisement. If the Router Advertisement contains more than one prefix option, the MS shall only consider the first one and silently discard the others.

Because any prefix that the GGSN advertises in a PDP context is unique within the scope of the prefix (i.e. site-local or global), there is no need for the MS to perform Duplicate Address Detection for this IPv6 address. Therefore, the GGSN shall silently discard Neighbor Solicitation messages that the MS may send to perform Duplicate Address Detection.

It is possible for the MS to perform Neighbor Unreachability Detection towards the GGSN, as defined in RFC 2461; therefore if the GGSN receives a Neighbor Solicitation as part of this procedure, the GGSN shall provide a Neighbor Advertisement as described in RFC 2461.

So now we know we need to implement router advertisements inside the GGSN before we can proceed...

---

Hi Philipp,

it should be working now, at least more than before. In my test case, I can now see a seemingly-valid router advertisement in response to the router solicitation. Could you give current laforge/ipv6 another [quick] try, please?

My ggsn config file is attached. The important part is to have a pool with a prefix shorter than 64 bits, i.e. something like /56, so you have 256 prefixes that can be allocated to individual MSs.

---

Hello Harald,

I have tried it again. I can see the router advertisements going up the gb interface. However, the pdp context still closes after a few seconds. I have attached the archive with the logs and the config files.

regards.
Philipp

---

Hi Philipp,

On Wed, Aug 09, 2017 at 10:53:24AM +0000, dexter [REDMINE] wrote:

I have tried it again. I can see the router advertisements going up the gb interface. However, the pdp context still closes after a few seconds. I have attached the archive with the logs and the config files.

Ok, it seems the MS needs to do neighbor solicitation after the router advertisement. I'll investigate. Meanwhile I've also found some phones using which I should hopefully be able to test myself.

---

So, finally progress. I could make the current "laforge/ipv6" branch work with my Fairphone2: It establishes the v6 PDP context successfully, gets
through PDP context activation and router solicitation/advertisement and then successfully establishes bi-directional TCP traffic to the public IPv6 internet.

Changes compared to yesterday's version were:

- implement PCO for IPv6 DNS server address assignment
- fix some bits in router advertisement message that were not in-line with 3GPP requirements (L/A bit)
- make sure that the upper 64bit of the EUA in the PDP Context Establishment matches the /64 prefix of the router advertisement that follows

Philip: Please reproduce locally with your phone(s) and confirm it works for you, too

#12 - 08/10/2017 09:39 AM - dexter
- File trace_ipv6_test_android_10082017.pcapng added
- File trace_ipv6_test_nokiaE90_10082017.pcapng added

Attached the traces of the recent tests.

- Android: PDP context still closes immediately, but rest looks good. We suspect that the pdp context is closed because the DNS server is unreachable from the test host.
- nokiaE90: PDP context stays open until it is manually closed by the user.

#13 - 08/12/2017 08:48 PM - laforge
- Status changed from In Progress to Closed
- % Done changed from 80 to 100

merged to master.

Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Date</th>
<th>Username</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipv6_pdp_contextReject.pcapng</td>
<td>480 Bytes</td>
<td>08/03/2017</td>
<td>dexter</td>
</tr>
<tr>
<td>trace_v6pdpctx_act_and_deact_04072017.tar</td>
<td>1.62 MB</td>
<td>08/04/2017</td>
<td>dexter</td>
</tr>
<tr>
<td>v6.conf</td>
<td>2.41 KB</td>
<td>08/09/2017</td>
<td>laforge</td>
</tr>
<tr>
<td>trace_v6pdpctx_act_and_deact_09082017.tar</td>
<td>2.36 MB</td>
<td>08/09/2017</td>
<td>dexter</td>
</tr>
<tr>
<td>trace_ipv6_test_android_10082017.pcapng</td>
<td>14.1 KB</td>
<td>08/10/2017</td>
<td>dexter</td>
</tr>
<tr>
<td>trace_ipv6_test_nokiaE90_10082017.pcapng</td>
<td>29.4 KB</td>
<td>08/10/2017</td>
<td>dexter</td>
</tr>
</tbody>
</table>