

OsmoGSMTester - Bug #2683

ofono: EC20 failing to create ipv6 or dual contexts

11/27/2017 10:26 AM - pespin

| | |
|--|-------------------------------|
| Status: In Progress | Start date: 11/27/2017 |
| Priority: Normal | Due date: |
| Assignee: lynxis | % Done: 10% |
| Category: | |
| Target version: | |
| Spec Reference: | |
| Description During setup of GPRS tests with osmo-pcu + osmo-sgsn + osmo-ggsn, the ipv4 context is created fine, but if I configure the context with other types, then during Activate it fails: 1. SetProperty 'Protocol' 'ipv6' org.ofono.Error.NotImplemented: Implementation not provided (36) 1. SetProperty 'Protocol' 'dual' org.ofono.Error.Failed: Operation failed (36) | |
| Related issues: | |
| Related to OsmoGSMTester - Bug #2713: ofono: chosen pdp type (v4, v6, dual) ... | New 12/06/2017 |
| Blocked by OsmoGGSN (former OpenGGSN) - Bug #2687: implement ipv4v6 context type | Closed 11/28/2017 |

History

#1 - 11/28/2017 09:35 AM - lynxis

- Status changed from New to Stalled

- % Done changed from 0 to 10

It needs to implement 2x WDS sessions to get ipv4 + ipv6. I'm still unsure, what it needs to be done, to set the PDP type to ip_4_6

#2 - 11/28/2017 09:55 AM - laforge

On Tue, Nov 28, 2017 at 09:35:00AM +0000, lynxis [REDMINE] wrote:

It needs to implement 2x WDS sessions to get ipv4 + ipv6. I'm still unsure, what it needs to be done, to set the PDP type to ip_4_6

v4v6 PDP context were added in rather late/recent 3GPP versions, so it could simply be that the EC20 doesn't yet support it. So if there's no obvious way to do it, probably not worth spending time to do a lot of research on it.

One thing worth trying though is to see if it's possible to somehow define a PDP context of v4v6 type via AT commands. If that works, then the modem likely supports it but you simply don't know yet how to do it via QMI. So if AT command works, then it's worth investing time researching how to do it on QMI.

Please note that I don't think we have tested v4v6 in the OsmoSGSN/GGSN yet, and in fact I'm quite sure it won't work yet with the current code. So testing is a bit of a chicken-and-egg situation.

Regards,
Harald

#3 - 11/28/2017 11:15 AM - lynxis

The ec20 with firmware "EC20EQAR02A13E2G 1 [Jul 5 2017 10:49:17]" is actually requesting a PDP context ipv4v6. ofono doesn't know how to set the PDP context type.

#4 - 11/28/2017 03:13 PM - lynxis

When using ofono on the EC20, the EC20 is falling back from ipv4v6 to ipv4 or ipv6 depending on the requested type. osmo-GGSN doesn't support the dual ipv4 ipv6 support yet and is rejecting with 220

```
#define GTPCAUSE_UNKNOWN_PDP 220 /* Unknown PDP address or PDP type */
```

#5 - 11/28/2017 03:14 PM - lynxis

- Blocked by Bug #2687: implement ipv4v6 context type added

#6 - 11/28/2017 04:11 PM - lynxis

Also interesting: when requesting an pdp ipv6 context, the EC20 and the Sierra are crashing after receiving an pdp context accept.

#7 - 12/12/2017 03:28 PM - pespin

- Related to Bug #2713: ofono: chosen pdp type (v4, v6, dual) not strictly followed in ofono qmi added

#8 - 12/12/2017 03:39 PM - pespin

- Status changed from Stalled to In Progress

Doing some more tests with osmo-gsm-tester now that IPv4v6 is implemented (an ipv4 and an ipv6 addr are returned when an ipv4v6 is requested):

- ipv4: Ofono actually sends an IPv4v6 ctx request. Works fine but I need to set the APN to "inet46" as it is the only one configured with a v4 pool and a v6 pool. If any of the pools is missing, osmo-ggsn fails to allocate an IP for one of the 2 types and then it sends back an error because it cannot fulfill the request. We should check in specs how is the osmo-ggsn expected to behave if IPv4v6 is requested but some parts cannot be fulfilled (ie if apn is configured to use ipv4 only or ipv6 only, if it's fine then returning only 1 address).

```
/sierra_1: context activated {apn='inet46', path='/sierra_1/context2', properties={AccessPointName='inet46', Active=True, AuthenticationMethod='chap', IPv6.Settings={}, Name='Internet', Password='', Protocol='ip', Settings={Address='176.16.46.1', DomainNameServers=['8.8.8.8', '8.8.8.8'], Gateway='176.16.46.2', Interface='wwp0s18f2ulu2i8', Method='static', Netmask='255.255.255.252'}, Type='internet', Username='ogt'}, user='ogt'}
```

- ipv6: Ofono actually send an IPv4v6 ctx request. Same as for ipv4. I can create the context without errors (and without modem crash) but in this case, it seems the ipv6 ip or other settings is not populated to the dbus API, not sure if it is expected, if features are missing or if it's a bug:

```
/sierra_1: context activated {apn='inet46', path='/sierra_1/context3', properties={AccessPointName='inet46', Active=True, AuthenticationMethod='chap', IPv6.Settings={Interface='wwp0s18f2ulu2i8'}, Name='Internet', Password='', Protocol='ipv6', Settings={}, Type='internet', Username='ogt'}, user='ogt'}
```

- dual: Ofono fails and the modem sends nothing over the air:

```
GDBus.Error:org.ofono.Error.Failed: Operation failed (36)
```

#9 - 12/15/2017 11:33 AM - pespin

[lynxis](#): Can you retry creating an ipv6 context with ofono and SierraW using latest osmo-ggsn (37c45e3998fca240b5266abd3ac883d0a35bab50)? It is working fine in osmo-gsm-tester now (using ipv4v6 underneath, both in Prod and R&D). Make sure your osmo-ggsn assigns a link-local address to the tun interface, otherwise force it by using the "ipv6 link-local fe80::1111:1111:1111:1111/64" vty cmd.

Creating the IPv4v6 probably fails due to qmi_activate_primary() and should be addresses as part of [#2713](#), so let's leave it outside from this task:

```
switch (ctx->proto) {
case OFONO_GPRS_PROTO_IP:
    ip_family = 4;
    break;
case OFONO_GPRS_PROTO_IPV6:
    ip_family = 6;
    break;
default:
    goto error;
}
```

Please have a look again with the different modems and close the task if found as working (as it does for me with the SierraW in osmo-gsm-tester).

#10 - 12/15/2017 11:41 AM - pespin

I verified that it also works fine with EC20 (same as SierraW):

```
12:37:15.878983 tst /gobi_0: DBG: 'org.ofono.ConnectionManager'.PropertyChanged() ->
Attached=True
12:37:15.918028 tst /gobi_0: DBG: attached: True
12:37:15.949666 tst /gobi_0: DBG: activate_context {apn='inet46', protocol='ip', user
='ogt'}
12:37:16.839428 tst /gobi_0: context activated {apn='inet46', path='/gobi_0/context2'
, properties={AccessPointName='inet46', A
ctive=True, AuthenticationMethod='chap', IPv6.Settings={}, Name='Internet', Password='', Protocol='ip', Settin
gs={Address='176.16.46.1', DomainNameServ
ers=['8.8.8.8', '8.8.8.8'], Gateway='176.16.46.2', Interface='wvp0s18f2ulu6i4', Method='static', Netmask='255.
255.255.252'}, Type='internet', Username=
'ogt'}, user='ogt'}
12:37:21.883354 tst /gobi_0: DBG: activate_context {path='/gobi_0/context2'}
12:37:23.516130 tst /gobi_0: DBG: activate_context {apn='inet46', protocol='ipv6', us
er='ogt'}
12:37:31.639298 tst /gobi_0: context activated {apn='inet46', path='/gobi_0/context3'
, properties={AccessPointName='inet46', A
ctive=True, AuthenticationMethod='chap', IPv6.Settings={Interface='wvp0s18f2ulu6i4'}, Name='Internet', Passwor
d='', Protocol='ipv6', Settings={}, Type=
'internet', Username='ogt'}, user='ogt'}
12:37:36.677732 tst /gobi_0: DBG: activate_context {path='/gobi_0/context3'}
```

Remember that I have to use inet46 APN due to ofono actually requesting an IPv4v6 ctx underneath ([#2713](#)).