

OsmoGSMTester - Bug #3881

ofono: Sometimes provides wrong net interface in DBus API

04/01/2019 09:54 AM - pespin

Status:	Resolved	Start date:	04/01/2019
Priority:	Normal	Due date:	
Assignee:	pespin	% Done:	100%
Category:			
Target version:			
Spec Reference:			

Description

I saw some osmo-gsm-tester tests that failed:

```
04:15:11.807003 run          osmo-hlr_10.42.42.2: Add subscriber {algo=1, algo_str='comp128v1',
, imsi='901700000010978', msisdn='1820', subscriber_id=1}
04:15:12.839312 bus          /gobi_10: Setting Online False
04:15:13.857703 bus          /gobi_10: Setting Powered False
04:15:20.353010 bus          /gobi_10: Setting Powered True
04:15:21.545365 bus          /gobi_10: Setting Online True
04:15:23.631041 tst          /gobi_10: Connect to ('901', '70')
04:15:23.665366 tst          /gobi_10: Manufacturer: 'Quectel', Model: 'EC20', Revision: 'EC20EQAR02A13E2G 1 [Jul 5 2017 10:49:17]', Serial: '20120822'
04:15:23.666671 tst          ping_idle_ping.py:47: waiting for modems to attach...
04:15:30.904372 tst          /gobi_10: Registering with operator /gobi_10/operator/90170 ('901', '70')
04:15:36.831376 tst          ping_idle_ping.py:51: waiting for modems to attach to data service
s...
04:15:44.260744 tst          /gobi_10: context activated {apn='inet46', path='/gobi_10/context2',
properties={AccessPointName='inet46', Active=True, AuthenticationMethod='chap', IPv6.Settings={}, Name='Internet', Password='', Protocol='ip', Settings={Address='176.16.46.1', DomainNameServers=['192.168.100.1', '8.8.8.8'], Gateway='176.16.46.2', Interface='wwan0', Method='static', Netmask='255.255.255.252'}, Type='internet', Username='ogt'}, user='ogt'}
04:15:44.262922 tst          ping_idle_ping.py:56: Setting up data plane for "/gobi_10/context2"
04:15:44.415972 run          ifup(pid=7019): Launched
04:15:45.421749 run          ifup(pid=7019): ERR: Terminated: ERROR {rc=1} [trial-1810→gprs:oc2g+mod-bts0-dyntos-osmo→/gobi_10→ifup(pid=7019)]
04:15:45.423708 run          ifup(pid=7019): stderr:
| (launched: 2019-04-01_04:15:44.404070)
| Cannot find device "wwan0"
04:15:45.427225 tst          ping_idle_ping.py:57: ERR: Error: Exited in error 1 [trial-1810→gprs:oc2g+mod-bts0-dyntos-osmo→ping_idle_ping.py:57]
04:15:45.433608 tst          ping_idle_ping.py:57: Test FAILED (70.0 sec)
```

So I went and checked configuration of the net namespaces (no interesting netiface in main netns):

```
root@osmo-gsm-tester-prod:~# for ns in `ip netns`; do echo "== $ns =="; ip netns exec $ns ip l; done
== 1-5.3.2 ==
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
18: wwan1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 10
    00
    link/ether 7a:3e:ed:a5:84:9a brd ff:ff:ff:ff:ff:ff
== 1-5.3.3 ==
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
19: wwan2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 10
    00
    link/ether 7a:3e:ed:a5:84:9a brd ff:ff:ff:ff:ff:ff
== 1-5.3.5 ==
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 1
```

```

    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
20: wwan3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 10
00
    link/ether 7a:3e:ed:a5:84:9a brd ff:ff:ff:ff:ff:ff
== 1-5.3.6 ==
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
21: wwan4: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 10
00
    link/ether 7a:3e:ed:a5:84:9a brd ff:ff:ff:ff:ff:ff
== 1-5.4.1.2 ==
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
22: wwan6: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 10
00
    link/ether 7a:3e:ed:a5:84:9a brd ff:ff:ff:ff:ff:ff
23: wwan7: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 10
00
    link/ether 7a:3e:ed:a5:84:9a brd ff:ff:ff:ff:ff:ff
== 1-5.4.1.3 ==
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
24: wwan8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 10
00
    link/ether 7a:3e:ed:a5:84:9a brd ff:ff:ff:ff:ff:ff
25: wwan9: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 10
00
    link/ether 7a:3e:ed:a5:84:9a brd ff:ff:ff:ff:ff:ff
== 1-5.4.1.5 ==
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
16: wwan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 10
00
    link/ether 7a:3e:ed:a5:84:9a brd ff:ff:ff:ff:ff:ff
== 1-5.4.1.6 ==
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2343: dev2343: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ether 7a:3e:ed:a5:84:9a brd ff:ff:ff:ff:ff:ff

```

According to `./show_usb_device.py`, 1-5.4.1.5 containing `wwan0` is the only Gobi2000 modem in the setup, but `ofono` states in the test that it is using an EC20 modem (as expected, since it's the only in `resources.conf` to be used on `gprs` tests).

The first EC20 modem in `resources.conf.prod` (the one which should be used and thus is used) is the one in 1-5.4.1.6, which interestingly has a "dev2343" net iface instead of usual "wwanX" one. That may be related regarding how `ofono` decided to provide "wwan0" for it instead of "dev2343".

Running "`modem-netns-setup.py stop`" to return all net ifaces to main netns, then running `./show_usb_device.py` shows how net ifaces relate to each modem:

```

root@osmo-gsm-tester-prod:~# ./show_usb_device.py
{<DEVICE ID 05c6:9215 on Bus 001 Address 041>: {'cdc': ['cdc-wdm5'],
      'name': 'Quectel Quectel LTE '
      'Module',
      'net': ['dev2343'],
      'path': '/sys/bus/usb/devices/1-5.4.1.6/',
      'serial': ['ttyUSB26',
      'ttyUSB24',
      'ttyUSB27',
      'ttyUSB25']},
<DEVICE ID 05c6:9205 on Bus 001 Address 031>: {'cdc': ['cdc-wdm0'],
      'name': 'Qualcomm Incorporated '
      'Qualcomm Gobi 2000',
      'net': ['wwan0'],
      'path': '/sys/bus/usb/devices/1-5.4.1.5/',

```

```

        'serial': ['ttyUSB7',
                  'ttyUSB5',
                  'ttyUSB6']],
<DEVICE ID 1199:68c0 on Bus 001 Address 038>: {'cdc': ['cdc-wdm9', 'cdc-wdm8'],
        'name': 'Sierra Wireless, '
              'Incorporated MC7304',
        'net': ['wwan9', 'wwan8'],
        'path': '/sys/bus/usb/devices/1-5.4.1.3/',
        'serial': ['ttyUSB32',
                  'ttyUSB31',
                  'ttyUSB33']],
<DEVICE ID 1199:68c0 on Bus 001 Address 037>: {'cdc': ['cdc-wdm7', 'cdc-wdm6'],
        'name': 'Sierra Wireless, '
              'Incorporated MC7304',
        'net': ['wwan7', 'wwan6'],
        'path': '/sys/bus/usb/devices/1-5.4.1.2/',
        'serial': ['ttyUSB29',
                  'ttyUSB28',
                  'ttyUSB30']],
<DEVICE ID 05c6:9215 on Bus 001 Address 036>: {'cdc': ['cdc-wdm4'],
        'name': 'Quectel Quectel LTE '
              'Module',
        'net': ['wwan4'],
        'path': '/sys/bus/usb/devices/1-5.3.6/',
        'serial': ['ttyUSB23',
                  'ttyUSB21',
                  'ttyUSB22',
                  'ttyUSB20']],
<DEVICE ID 05c6:9215 on Bus 001 Address 035>: {'cdc': ['cdc-wdm3'],
        'name': 'Quectel Quectel LTE '
              'Module',
        'net': ['wwan3'],
        'path': '/sys/bus/usb/devices/1-5.3.5/',
        'serial': ['ttyUSB18',
                  'ttyUSB16',
                  'ttyUSB19',
                  'ttyUSB17']],
<DEVICE ID 05c6:9215 on Bus 001 Address 034>: {'cdc': ['cdc-wdm2'],
        'name': 'Quectel Quectel LTE '
              'Module',
        'net': ['wwan2'],
        'path': '/sys/bus/usb/devices/1-5.3.3/',
        'serial': ['ttyUSB15',
                  'ttyUSB13',
                  'ttyUSB14',
                  'ttyUSB12']],
<DEVICE ID 05c6:9215 on Bus 001 Address 033>: {'cdc': ['cdc-wdm1'],
        'name': 'Quectel Quectel LTE '
              'Module',
        'net': ['wwan1'],
        'path': '/sys/bus/usb/devices/1-5.3.2/',
        'serial': ['ttyUSB11',
                  'ttyUSB9',
                  'ttyUSB10',
                  'ttyUSB8']],
<DEVICE ID 10c4:ea60 on Bus 001 Address 008>: {'cdc': [],
        'name': 'Silicon Labs CP2104 '
              'USB to UART Bridge '
              'Controller',
        'net': [],
        'path': '/sys/bus/usb/devices/1-5.2/',
        'serial': ['ttyUSB4']],
<DEVICE ID 0403:6001 on Bus 001 Address 006>: {'cdc': [],
        'name': 'FTDI FT232R USB UART',
        'net': [],
        'path': '/sys/bus/usb/devices/1-5.1/',
        'serial': ['ttyUSB1']],

```

```

<DEVICE ID 0403:6001 on Bus 001 Address 007>: {'cdc': [],
      'name': 'FTDI FT232R USB UART',
      'net': [],
      'path': '/sys/bus/usb/devices/1-1.4/',
      'serial': ['ttyUSB2']},
<DEVICE ID 0403:6001 on Bus 001 Address 005>: {'cdc': [],
      'name': 'FTDI FT232R USB UART',
      'net': [],
      'path': '/sys/bus/usb/devices/1-1.3/',
      'serial': ['ttyUSB0']},
<DEVICE ID 10c4:ea60 on Bus 001 Address 004>: {'cdc': [],
      'name': 'Silicon Labs CP2102 '
      'USB to UART Bridge '
      'Controller',
      'net': [],
      'path': '/sys/bus/usb/devices/1-1.1/',
      'serial': ['ttyUSB3']}]

```

So definitely ofono is announcing wwan0 for 1-5.4.1.6 instead of dev2343. Could it be related to our patch to avoid dropping information about the modem netns in ofono (to allow moving net ifaces to other netns):

<https://git.sylom.com/de/ofono/commit/?h=osmo-gsm-tester&id=e77b82a1bd1986aab42a119e0556c5941795992e>

Related issues:

Related to OsmoGSMTester - Support #3931: Contact quectel to get info about m...	New	04/15/2019
--	------------	-------------------

History

#1 - 04/01/2019 10:20 AM - pespip

Interestingly, according to dmesg wwan0 iface was first assigned to Gobi2000 (1-5.4.1.5) and later to EC20 (1-5.4.1.6):

```

root@osmo-gsm-tester-prod:~# dmesg | grep wwan0
[ 17.028189] qmi_wwan 1-5.4.1.5:1.0 wwan0: register 'qmi_wwan' at usb-0000:00:12.2-5.4.1.5, WWAN/QMI device,
7a:3e:ed:a5:84:9a
[ 4121.598616] qmi_wwan 1-5.4.1.5:1.0 wwan0: unregister 'qmi_wwan' usb-0000:00:12.2-5.4.1.5, WWAN/QMI device
[ 4126.115197] qmi_wwan 1-5.4.1.5:1.0 wwan0: register 'qmi_wwan' at usb-0000:00:12.2-5.4.1.5, WWAN/QMI device,
7a:3e:ed:a5:84:9a
[29263.362075] qmi_wwan 1-5.4.1.6:1.4 wwan0: register 'qmi_wwan' at usb-0000:00:12.2-5.4.1.6, WWAN/QMI device,
7a:3e:ed:a5:84:9a
[216205.198406] qmi_wwan 1-5.4.1.6:1.4 wwan0: unregister 'qmi_wwan' usb-0000:00:12.2-5.4.1.6, WWAN/QMI device
[216215.028310] qmi_wwan 1-5.4.1.6:1.4 wwan0: register 'qmi_wwan' at usb-0000:00:12.2-5.4.1.6, WWAN/QMI device,
7a:3e:ed:a5:84:9a
[314476.953665] qmi_wwan 1-5.4.1.6:1.4 wwan0: unregister 'qmi_wwan' usb-0000:00:12.2-5.4.1.6, WWAN/QMI device
[314487.046418] qmi_wwan 1-5.4.1.6:1.4 wwan0: register 'qmi_wwan' at usb-0000:00:12.2-5.4.1.6, WWAN/QMI device,
7a:3e:ed:a5:84:9a

```

Even more interesting, it seems it was registered twice without unregistering in the middle, and once for each modem:

```

[ 4126.115197] qmi_wwan 1-5.4.1.5:1.0 wwan0: register 'qmi_wwan' at usb-0000:00:12.2-5.4.1.5, WWAN/QMI device,
7a:3e:ed:a5:84:9a
[29263.362075] qmi_wwan 1-5.4.1.6:1.4 wwan0: register 'qmi_wwan' at usb-0000:00:12.2-5.4.1.6, WWAN/QMI device,
7a:3e:ed:a5:84:9a

```

And actually now there exists wwan0 and dev2342:

```

16: wwan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ether 7a:3e:ed:a5:84:9a brd ff:ff:ff:ff:ff:ff
2343: dev2343: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ether 7a:3e:ed:a5:84:9a brd ff:ff:ff:ff:ff:ff

```

Here's full log of dmesg filtered for both modems:

```

root@osmo-gsm-tester-prod:~# dmesg | grep -e 1-5.4.1.5 -e 1-5.4.1.6
[ 12.612269] usb 1-5.4.1.5: new high-speed USB device number 17 using ehci-pci
[ 12.722635] usb 1-5.4.1.5: config 1 has an invalid interface number: 1 but max is 0
[ 12.722648] usb 1-5.4.1.5: config 1 has no interface number 0
[ 12.724575] usb 1-5.4.1.5: New USB device found, idVendor=05c6, idProduct=9204
[ 12.724583] usb 1-5.4.1.5: New USB device strings: Mfr=3, Product=2, SerialNumber=0

```

```
[ 12.724587] usb 1-5.4.1.5: Product: Qualcomm Gobi 2000
[ 12.724592] usb 1-5.4.1.5: Manufacturer: Qualcomm Incorporated
[ 12.743874] qcserial 1-5.4.1.5:1.1: Qualcomm USB modem converter detected
[ 12.744968] usb 1-5.4.1.5: Qualcomm USB modem converter now attached to ttyUSB5
[ 15.872346] usb 1-5.4.1.5: USB disconnect, device number 17
[ 15.872893] qcserial 1-5.4.1.5:1.1: device disconnected
[ 16.864362] usb 1-5.4.1.5: new high-speed USB device number 18 using ehci-pci
[ 16.975228] usb 1-5.4.1.5: New USB device found, idVendor=05c6, idProduct=9205
[ 16.975237] usb 1-5.4.1.5: New USB device strings: Mfr=3, Product=2, SerialNumber=0
[ 16.975242] usb 1-5.4.1.5: Product: Qualcomm Gobi 2000
[ 16.975246] usb 1-5.4.1.5: Manufacturer: Qualcomm Incorporated
[ 16.980643] qcserial 1-5.4.1.5:1.1: Qualcomm USB modem converter detected
[ 16.981008] usb 1-5.4.1.5: Qualcomm USB modem converter now attached to ttyUSB5
[ 16.982576] qcserial 1-5.4.1.5:1.2: Qualcomm USB modem converter detected
[ 16.982929] usb 1-5.4.1.5: Qualcomm USB modem converter now attached to ttyUSB6
[ 16.984345] qcserial 1-5.4.1.5:1.3: Qualcomm USB modem converter detected
[ 16.984680] usb 1-5.4.1.5: Qualcomm USB modem converter now attached to ttyUSB7
[ 17.025904] qmi_wwan 1-5.4.1.5:1.0: cdc-wdm0: USB WDM device
[ 17.028189] qmi_wwan 1-5.4.1.5:1.0 wwan0: register 'qmi_wwan' at usb-0000:00:12.2-5.4.1.5, WWAN/QMI device,
7a:3e:ed:a5:84:9a
[ 21.984710] usb 1-5.4.1.6: new high-speed USB device number 23 using ehci-pci
[ 22.099053] usb 1-5.4.1.6: New USB device found, idVendor=05c6, idProduct=9215
[ 22.099064] usb 1-5.4.1.6: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ 22.099070] usb 1-5.4.1.6: Product: Quectel LTE Module
[ 22.099075] usb 1-5.4.1.6: Manufacturer: Quectel
[ 22.122215] qcserial 1-5.4.1.6:1.0: Qualcomm USB modem converter detected
[ 22.122670] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB24
[ 22.128759] qcserial 1-5.4.1.6:1.1: Qualcomm USB modem converter detected
[ 22.129224] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB25
[ 22.130140] qcserial 1-5.4.1.6:1.2: Qualcomm USB modem converter detected
[ 22.130525] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB26
[ 22.131639] qcserial 1-5.4.1.6:1.3: Qualcomm USB modem converter detected
[ 22.132028] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB27
[ 22.133221] qmi_wwan 1-5.4.1.6:1.4: cdc-wdm5: USB WDM device
[ 22.133875] qmi_wwan 1-5.4.1.6:1.4 wwan5: register 'qmi_wwan' at usb-0000:00:12.2-5.4.1.6, WWAN/QMI device,
7a:3e:ed:a5:84:9a
[ 4121.598354] usb 1-5.4.1.5: USB disconnect, device number 18
[ 4121.598616] qmi_wwan 1-5.4.1.5:1.0 wwan0: unregister 'qmi_wwan' usb-0000:00:12.2-5.4.1.5, WWAN/QMI device
[ 4121.614522] qcserial 1-5.4.1.5:1.1: device disconnected
[ 4121.614897] qcserial 1-5.4.1.5:1.2: device disconnected
[ 4121.615301] qcserial 1-5.4.1.5:1.3: device disconnected
[ 4121.842007] usb 1-5.4.1.5: new high-speed USB device number 28 using ehci-pci
[ 4121.952447] usb 1-5.4.1.5: config 1 has an invalid interface number: 1 but max is 0
[ 4121.952457] usb 1-5.4.1.5: config 1 has no interface number 0
[ 4121.954440] usb 1-5.4.1.5: New USB device found, idVendor=05c6, idProduct=9204
[ 4121.954453] usb 1-5.4.1.5: New USB device strings: Mfr=3, Product=2, SerialNumber=0
[ 4121.954458] usb 1-5.4.1.5: Product: Qualcomm Gobi 2000
[ 4121.954464] usb 1-5.4.1.5: Manufacturer: Qualcomm Incorporated
[ 4121.957389] qcserial 1-5.4.1.5:1.1: Qualcomm USB modem converter detected
[ 4121.957828] usb 1-5.4.1.5: Qualcomm USB modem converter now attached to ttyUSB5
[ 4125.258476] usb 1-5.4.1.5: USB disconnect, device number 28
[ 4125.259353] qcserial 1-5.4.1.5:1.1: device disconnected
[ 4125.993312] usb 1-5.4.1.5: new high-speed USB device number 31 using ehci-pci
[ 4126.108622] usb 1-5.4.1.5: New USB device found, idVendor=05c6, idProduct=9205
[ 4126.108633] usb 1-5.4.1.5: New USB device strings: Mfr=3, Product=2, SerialNumber=0
[ 4126.108639] usb 1-5.4.1.5: Product: Qualcomm Gobi 2000
[ 4126.108644] usb 1-5.4.1.5: Manufacturer: Qualcomm Incorporated
[ 4126.114494] qmi_wwan 1-5.4.1.5:1.0: cdc-wdm0: USB WDM device
[ 4126.115197] qmi_wwan 1-5.4.1.5:1.0 wwan0: register 'qmi_wwan' at usb-0000:00:12.2-5.4.1.5, WWAN/QMI device,
7a:3e:ed:a5:84:9a
[ 4126.116429] qcserial 1-5.4.1.5:1.1: Qualcomm USB modem converter detected
[ 4126.116756] usb 1-5.4.1.5: Qualcomm USB modem converter now attached to ttyUSB5
[ 4126.118217] qcserial 1-5.4.1.5:1.2: Qualcomm USB modem converter detected
[ 4126.118598] usb 1-5.4.1.5: Qualcomm USB modem converter now attached to ttyUSB6
[ 4126.119903] qcserial 1-5.4.1.5:1.3: Qualcomm USB modem converter detected
[ 4126.120268] usb 1-5.4.1.5: Qualcomm USB modem converter now attached to ttyUSB7
[ 4130.634130] usb 1-5.4.1.6: USB disconnect, device number 23
[ 4130.634860] qcserial 1-5.4.1.6:1.0: device disconnected
[ 4130.635413] qcserial 1-5.4.1.6:1.1: device disconnected
[ 4130.635870] qcserial 1-5.4.1.6:1.2: device disconnected
[ 4130.640193] qcserial 1-5.4.1.6:1.3: device disconnected
[ 4130.640530] qmi_wwan 1-5.4.1.6:1.4 wwan5: unregister 'qmi_wwan' usb-0000:00:12.2-5.4.1.6, WWAN/QMI device
[ 4130.880569] usb 1-5.4.1.6: new high-speed USB device number 32 using ehci-pci
[ 4130.990495] usb 1-5.4.1.6: New USB device found, idVendor=05c6, idProduct=9215
[ 4130.990506] usb 1-5.4.1.6: New USB device strings: Mfr=1, Product=2, SerialNumber=0
```

```
[ 4130.990512] usb 1-5.4.1.6: Product: Quectel LTE Module
[ 4130.990517] usb 1-5.4.1.6: Manufacturer: Quectel
[ 4131.010294] qcserial 1-5.4.1.6:1.0: Qualcomm USB modem converter detected
[ 4131.010833] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB24
[ 4131.014332] qcserial 1-5.4.1.6:1.1: Qualcomm USB modem converter detected
[ 4131.014804] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB25
[ 4131.020926] qcserial 1-5.4.1.6:1.2: Qualcomm USB modem converter detected
[ 4131.021384] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB26
[ 4131.022133] qcserial 1-5.4.1.6:1.3: Qualcomm USB modem converter detected
[ 4131.022452] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB27
[ 4131.023803] qmi_wwan 1-5.4.1.6:1.4: cdc-wdm5: USB WDM device
[ 4131.024474] qmi_wwan 1-5.4.1.6:1.4 wwan5: register 'qmi_wwan' at usb-0000:00:12.2-5.4.1.6, WWAN/QMI device,
7a:3e:ed:a5:84:9a
[29253.269410] usb 1-5.4.1.6: USB disconnect, device number 32
[29253.270225] qcserial 1-5.4.1.6:1.0: device disconnected
[29253.273356] qcserial 1-5.4.1.6:1.1: device disconnected
[29253.274099] qcserial 1-5.4.1.6:1.2: device disconnected
[29253.274689] qcserial 1-5.4.1.6:1.3: device disconnected
[29253.284031] qmi_wwan 1-5.4.1.6:1.4 wwan5: unregister 'qmi_wwan' usb-0000:00:12.2-5.4.1.6, WWAN/QMI device
[29263.219314] usb 1-5.4.1.6: new high-speed USB device number 39 using ehci-pci
[29263.329609] usb 1-5.4.1.6: New USB device found, idVendor=05c6, idProduct=9215
[29263.329619] usb 1-5.4.1.6: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[29263.329625] usb 1-5.4.1.6: Product: Quectel LTE Module
[29263.329630] usb 1-5.4.1.6: Manufacturer: Quectel
[29263.347866] qcserial 1-5.4.1.6:1.0: Qualcomm USB modem converter detected
[29263.348302] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB24
[29263.352125] qcserial 1-5.4.1.6:1.1: Qualcomm USB modem converter detected
[29263.352602] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB25
[29263.359027] qcserial 1-5.4.1.6:1.2: Qualcomm USB modem converter detected
[29263.359538] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB26
[29263.360253] qcserial 1-5.4.1.6:1.3: Qualcomm USB modem converter detected
[29263.360567] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB27
[29263.361434] qmi_wwan 1-5.4.1.6:1.4: cdc-wdm5: USB WDM device
[29263.362075] qmi_wwan 1-5.4.1.6:1.4 wwan0: register 'qmi_wwan' at usb-0000:00:12.2-5.4.1.6, WWAN/QMI device,
7a:3e:ed:a5:84:9a
[216205.188995] usb 1-5.4.1.6: USB disconnect, device number 39
[216205.189841] qcserial 1-5.4.1.6:1.0: device disconnected
[216205.190687] qcserial 1-5.4.1.6:1.1: device disconnected
[216205.191495] qcserial 1-5.4.1.6:1.2: device disconnected
[216205.192310] qcserial 1-5.4.1.6:1.3: device disconnected
[216205.198406] qmi_wwan 1-5.4.1.6:1.4 wwan0: unregister 'qmi_wwan' usb-0000:00:12.2-5.4.1.6, WWAN/QMI device
[216214.884456] usb 1-5.4.1.6: new high-speed USB device number 40 using ehci-pci
[216214.993613] usb 1-5.4.1.6: New USB device found, idVendor=05c6, idProduct=9215
[216214.993625] usb 1-5.4.1.6: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[216214.993630] usb 1-5.4.1.6: Product: Quectel LTE Module
[216214.993635] usb 1-5.4.1.6: Manufacturer: Quectel
[216215.011726] qcserial 1-5.4.1.6:1.0: Qualcomm USB modem converter detected
[216215.012182] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB24
[216215.016500] qcserial 1-5.4.1.6:1.1: Qualcomm USB modem converter detected
[216215.016992] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB25
[216215.023360] qcserial 1-5.4.1.6:1.2: Qualcomm USB modem converter detected
[216215.025394] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB26
[216215.026280] qcserial 1-5.4.1.6:1.3: Qualcomm USB modem converter detected
[216215.026637] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB27
[216215.027590] qmi_wwan 1-5.4.1.6:1.4: cdc-wdm5: USB WDM device
[216215.028310] qmi_wwan 1-5.4.1.6:1.4 wwan0: register 'qmi_wwan' at usb-0000:00:12.2-5.4.1.6, WWAN/QMI device
, 7a:3e:ed:a5:84:9a
[314476.941924] usb 1-5.4.1.6: USB disconnect, device number 40
[314476.942968] qcserial 1-5.4.1.6:1.0: device disconnected
[314476.944105] qcserial 1-5.4.1.6:1.1: device disconnected
[314476.944983] qcserial 1-5.4.1.6:1.2: device disconnected
[314476.946091] qcserial 1-5.4.1.6:1.3: device disconnected
[314476.953665] qmi_wwan 1-5.4.1.6:1.4 wwan0: unregister 'qmi_wwan' usb-0000:00:12.2-5.4.1.6, WWAN/QMI device
[314486.892590] usb 1-5.4.1.6: new high-speed USB device number 41 using ehci-pci
[314487.006696] usb 1-5.4.1.6: New USB device found, idVendor=05c6, idProduct=9215
[314487.006707] usb 1-5.4.1.6: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[314487.006712] usb 1-5.4.1.6: Product: Quectel LTE Module
[314487.006717] usb 1-5.4.1.6: Manufacturer: Quectel
[314487.037244] qcserial 1-5.4.1.6:1.0: Qualcomm USB modem converter detected
[314487.038110] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB24
[314487.039345] qcserial 1-5.4.1.6:1.1: Qualcomm USB modem converter detected
[314487.039959] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB25
[314487.041322] qcserial 1-5.4.1.6:1.2: Qualcomm USB modem converter detected
[314487.041966] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB26
[314487.043429] qcserial 1-5.4.1.6:1.3: Qualcomm USB modem converter detected
```

```
[314487.043999] usb 1-5.4.1.6: Qualcomm USB modem converter now attached to ttyUSB27
[314487.045720] qmi_wwan 1-5.4.1.6:1.4: cdc-wdm5: USB WDM device
[314487.046418] qmi_wwan 1-5.4.1.6:1.4 wwan0: register 'qmi_wwan' at usb-0000:00:12.2-5.4.1.6, WWAN/QMI device
, 7a:3e:ed:a5:84:9a
```

So it can clearly be seen that:

- During initial boot, gobi2000 is assigned wwan0 and ec20 is assigned wwan5.
- Both of them crash from time to time (devices are unregistered and then registered again).
- At some point (29263.362075), while gobi2000 is registered, then EC20 crashes and when registering again, it gets assigned wwan0 despite it should be for gobi2000?! (dev2343 doesn't show up in dmesg)

So my guess is that in 29263.362075 when ec20 crashed, it was assigned wwan0 because gobi2000's wwan0 was not in the main netns. Then I possibly spotted the crash and moved it to its netns again (./netns-modem-setup.py start), probably at some point returning all of them to the main netns and thus creating some kind of collision (2 ifaces named wwan0) and the kernel renaming one of them, and ofono not catching the rename.

What I'd investigate: See how this naming-collision is handled in the kernel between netns, and how udev announced them, then see if ofono implements a catch for this renaming.

#2 - 04/01/2019 10:38 AM - pespin

I confirm ofono doesn't handle iface name renaming. I tried changing the name of one iface manually with following command, and ofono didn't output any information:

```
ip link set wwan9 name wwan9pespin
```

That's what "udev monitor" shows while calling the command to move wwan9-old -> wwan9-new:

```
KERNEL[598426.927211] move /devices/pci0000:00/0000:00:12.2/usb1/1-5/1-5.4/1-5.4.1/1-5.4.1.3/1-5.4.1.3:1.1
0/net/wwan9-new (net)
UDEV [598426.959213] move /devices/pci0000:00/0000:00:12.2/usb1/1-5/1-5.4/1-5.4.1/1-5.4.1.3/1-5.4.1.3:1.1
0/net/wwan9 (net)
```

#3 - 04/01/2019 10:56 AM - pespin

I did test manually what happened to prove the concept (renaming to existing name in default netns inside another iface, then move it back to default):

```
$ ip link set wwan9-old netns tmp
KERNEL[598537.320874] remove /devices/pci0000:00/0000:00:12.2/usb1/1-5/1-5.4/1-5.4.1/1-5.4.1.3/1-5.4.1.3:1.1
0/net/wwan9-old (net)
UDEV [598537.332401] remove /devices/pci0000:00/0000:00:12.2/usb1/1-5/1-5.4/1-5.4.1/1-5.4.1.3/1-5.4.1.3:1.1
0/net/wwan9 (net)

$ ip netns exec tmp ip link set wwan9-old name wwan8
(nothing shown in udev monitor since it's in a different netns)
$ ip netns delete tmp
KERNEL[598659.888878] add /devices/pci0000:00/0000:00:12.2/usb1/1-5/1-5.4/1-5.4.1/1-5.4.1.3/1-5.4.1.3:1.1
0/net/wwan8 (net)
KERNEL[598659.889040] move /devices/pci0000:00/0000:00:12.2/usb1/1-5/1-5.4/1-5.4.1/1-5.4.1.3/1-5.4.1.3:1.1
0/net/dev25 (net)
UDEV [598660.058499] add /devices/pci0000:00/0000:00:12.2/usb1/1-5/1-5.4/1-5.4.1/1-5.4.1.3/1-5.4.1.3:1.1
0/net/wwan8 (net)
UDEV [598660.070552] move /devices/pci0000:00/0000:00:12.2/usb1/1-5/1-5.4/1-5.4.1/1-5.4.1.3/1-5.4.1.3:1.1
0/net/dev25 (net)
```

And indeed ofono didn't notice the change (I don't see related output in its stdout/stderr).

#4 - 04/01/2019 04:28 PM - pespin

I'm working on an ofono patch to update the interface send over Dbus upon iface rename, but I think I am hitting this systemd-udev bug: <https://github.com/systemd/systemd/issues/9426>

#5 - 04/01/2019 06:41 PM - pespin

- Status changed from New to In Progress
- % Done changed from 0 to 30

I confirm I see the systemd bug in debian9 (up-to-date systems), "udevadm monitor -p" is quite useful for that.

I submitted 2 patches to ofono Mailing lists which allows ofono to track network interface renames (if udev works without bugs).

So given we have the bug, I see several paths here:

- Add udev rules to rename modem network interfaces to always have the same name, this way even if a modem crashes it will keep the same name and ofono will submit the correct name.
- Our own ofono has a patch to avoid dropping the net device (and the whole modem) once it detects the net device gets out. If my patches are accepted, we need to change this patch to flag the net device in order to be able to drop it when the iface reappears in default netns after modem crash, otherwise after detecting the "move" the new device name is added but the old one is not dropped, so we end up with 2 devices and then the old one can override the new one when setting NetworkInterface property.
- Contact quectel and try to get rid of modem crashes. See if we can upgrade the firmware (ask for downgrade path possibility).

#6 - 04/02/2019 06:19 AM - laforge

On Mon, Apr 01, 2019 at 06:41:49PM +0000, pespin [REDMINE] wrote:

- Contact quectel and try to get rid of modem crashes. See if we can upgrade the firmware (ask for downgrade path possibility).

While I think it's useful to investigate updates for all kinds of reasons, I don't think you will ever be able to make the assumption that the baseband stack won't crash, particularly in our type of usage. And I would say that for any type of baseband/vendor.

#7 - 04/02/2019 04:20 PM - pespin

- % Done changed from 30 to 50

pespin wrote:

- Add udev rules to rename modem network interfaces to always have the same name, this way even if a modem crashes it will keep the same name and ofono will submit the correct name.

It took a big effort, but finally was able to make it work. dhcpcd was racing against udev and bringing interfaces up before udev tried renaming, which failed because iface was already up.

So, on top of the udev rule, one needs to configure dhcpcd to prevent it from managing those net ifaces you want to rename through udev. In /etc/dhcpcd.conf append:

```
# Don't ifup on modem ifaces, it prevents renaming
denyinterfaces ww*
```

Then the udev rule /etc/udev/rules.d/70-net-setup-link-modems.rules:

```
SUBSYSTEM!="net", GOTO="net_setup_link_end"
ACTION!="add", GOTO="net_setup_link_end"

IMPORT{builtin}="net_id"
ENV{DEVTYPE}=="wwan", ENV{ID_NET_NAME_PATH}!="", NAME="$env{ID_NET_NAME_PATH}"

LABEL="net_setup_link_end"
```

For some reason, SierraW modems don't get property ID_NET_NAME_PATH during call to "net_id" builtin, and so rule to rename the device doesn't apply, but we don't care since anyway we don't use gprs on those devices during testing.

#8 - 04/02/2019 05:47 PM - pespin

In Prod system none of the modems gets ID_NET_NAME_PATH. The reason is the same as per SierraW ones in RnD: It's devpath is too long/deep (too many usb-hub levels on it). That makes the resulting ID_NET_NAME_PATH > IFNAMESIZ (15 chars) which prevents it from setting ID_NET_NAME_PATH.

This scenario is actually documented in systemd source code: https://github.com/systemd/systemd/blob/master/src/udev/udev-builtin-net_id.c#L43

So we need to either:

- Drop one usb-hub level in the quadmem usb path, so it fits in 15 chars for ec20 modems
- Adding an explicit UDEV rule matching a specific devpath and providing a manual name for each modem

#9 - 04/02/2019 07:14 PM - pespin

- % Done changed from 50 to 60

I finally went for a 3rd option: if path is too long and ID_NET_NAME_PATH is not set, we generate a unique iface name ourselves by hashing the devpath of the net iface (prepending character "r").

<https://gerrit.osmocom.org/#/c/osmo-ci/+13495> ansible: osmo-gsm-tester: Make modem net ifaces persistent

TODO:

- follow-up on ofono submitted patches
- Contact quectel to get info about updates
- Make sure osmo-gsm-tester moved iface to its correct interface during test, in case a previous test made the modem crash (and its iface is re-registered on the default netns).

#10 - 04/04/2019 02:54 PM - pespin

- % Done changed from 60 to 70

- I submitted a patch with a fix for the udev rule which was generating variadic hashes on some scenarios:
<https://gerrit.osmocom.org/#/c/osmo-ci/+13518> ansible: gsm-tester: Remove net iface name from hash input
- osmo-gsm-tester now makes sure the iface is moved to correct netns before using it: <https://gerrit.osmocom.org/#/c/osmo-gsm-tester/+13526/>

TODO:

- merge patches
- follow-up on ofono submitted patches (I think not really needed by us anymore since we have persistent interface names)
- Contact quectel to get info about updates (let's see if we can get modems crashing less).

#11 - 04/15/2019 01:41 PM - pespin

- Related to Support #3931: Contact quectel to get info about modem updates (hopefully less crashes) added

#12 - 04/15/2019 01:41 PM - pespin

- Status changed from In Progress to Resolved

- % Done changed from 70 to 100

Handling quectel contact to upgrade firmware in [#3931](#), we can close this ticket.