OsmoGSMTester - Feature #2202

osmo-gsm-tester: add GPRS data services

04/26/2017 03:50 PM - neels

Status:	Resolved	**Start date:**	04/26/2017
Priority:	Urgent	**Due date:**	
Assignee:	pespin	**% Done:**	100%
Category:			
Target version:			
Spec Reference:			

Description

add osmo-pcu, osmo-sgsn, openggsn to the osmo-gsm-tester to allow testing data services.

- make sure these programs are built by jenkins and provided in the trial packages
- start up above programs for the various platforms

Related issues:

Related to OsmoGSMTester - Feature #2308: prepare GPRS for osmo-gsm-tester: t...	**In Progress**	**05/30/2017**
Blocks OsmoGSMTester - Feature #2203: osmo-gsm-tester: add test case: data te...	**Resolved**	**04/26/2017**
Blocks OsmoGSMTester - Feature #2204: osmo-gsm-tester: add test case: CS pagi...	**Resolved**	**04/27/2017**

History

#1 - 04/26/2017 03:51 PM - neels

- Blocks Feature #2203: osmo-gsm-tester: add test case: data test API added

#2 - 04/27/2017 12:17 AM - neels

- Blocks Feature #2204: osmo-gsm-tester: add test case: CS paging while GPRS is active added

#3 - 04/27/2017 01:38 AM - neels

- Assignee set to Osmocom Developers

#4 - 05/14/2017 11:20 AM - laforge

- Assignee deleted (Osmocom Developers)

#5 - 05/15/2017 12:58 PM - laforge

- Assignee set to osmo-gsm-tester

#6 - 07/03/2017 11:55 AM - neels

- Priority changed from Normal to Urgent

#7 - 07/25/2017 12:03 AM - neels

- Related to Feature #2308: prepare GPRS for osmo-gsm-tester: try to use ofono to connect to an osmocom GPRS service added

#8 - 09/01/2017 04:15 PM - pespin

- Status changed from New to In Progress

- Assignee changed from osmo-gsm-tester to pespin

- % Done changed from 0 to 20

Initial work on adding openggsn, osmo-sgsn and osmo-pcu-sysmo can be found in branch pespin/gprs of osmo-gsm-tester.

Missing:

- Add suppport for regular osmo-pcu to run with osmo-bts-trx.
- Add support for ConnectionManager from ofono API into modems (ofono_client.py).
- Bunch of changes in network topology to have some sort of network namespacing to be able to route packets correctly first through the modem and then through ggsn without looping. This is being tracked in #2308.

#9 - 09/02/2017 02:02 AM - neels

Looking at the branch commits, some comments:

"+# osmo_gsm_tester: specifics for running an openggsn"
The binary is 'ggsn', the project is OpenGGSN... technically 'openggsn' doesn't exist.
I'd prefer to use the actual binary's name 'ggsn', particularly in all log messages ("Starting ggsn")

BTW, in a mail thread we're discussing renaming to osmo-ggsn... but that's still premature.

Have you thought about the tunnel creation?
In my usual test scripts I have to sudo ggsn so that it is allowed to create a tunnel for the GPRS NAT thang.
On the tester we have so far been able to run everything as non-root, and I dimly remember Harald had a comment on how the tunnel problem should be solved...
IIRC it entails having a tunnel readily configured on the system and letting ggsn use that from user land.
Should we have a number of tunnels and treat them as resources?
I don't know much on the subject, just want to make sure you're aware of the need to clarify this.

Otherwise the patches make the impression that you've already run the setup successfully (impressed). Is that the case? Have you run it as root or in user land?

#10 - 09/02/2017 02:04 AM - neels

neels wrote:

> Should we have a number of tunnels and treat them as resources?

(could be less cumbersome if we found a way to give a normal user / group permissions to create tun devices somehow)

#11 - 09/02/2017 03:04 PM - laforge

Hi Neels + Pau,

On Sat, Sep 02, 2017 at 02:02:28AM +0000, neels [REDMINE] wrote:

> Have you thought about the tunnel creation?

"tun device creation". Each PDP context is a tunnel, let's avoid any misunderstanding.

> In my usual test scripts I have to sudo ggsn so that it is allowed to
> create a tunnel for the GPRS NAT thang. On the tester we have so far
> been able to run everything as non-root, and I dimly remember Harald
> had a comment on how the tunnel problem should be solved... IIRC it
> entails having a tunnel readily configured on the system and letting
> ggsn use that from user land.

yes, that's correct. It's the same with any other software that uses
tun devices: You can create persistent tun/tap devices using e.g. tunctl
from the shell (from root/CAP_NET_ADMIN) e.g. at system start or by the
systemd service before dropping privileges and starting the process.
The user/group owner of the persistent tun/tap device is set at its
creation time.

the laforge/osmo-sgsn branch of openggsn.git already has (tested) support for this.

> Should we have a number of tunnels and treat them as resources?

Probably makes sense given how osmo-gsm-tester works, though I'm not the
expert on that. Each tun device then consists of the device name, as
well as the network/netmask to which it is configured.

#12 - 09/04/2017 09:31 AM - pespin

> "+# osmo_gsm_tester: specifics for running an openggsn"
> The binary is 'ggsn', the project is OpenGGSN... technically 'openggsn' doesn't exist.
> I'd prefer to use the actual binary's name 'ggsn', particularly in all log messages ("Starting ggsn")

I'd prefer keeping the name openggsn as it directly points you the project and gives you an idea on what does that class wrap, and not some random/generic ggsn binary: it runs ggsn service from openggsn / OpenGGSN. If at some point we have more than one provider of ggsn service because us or someone else adds it, then I think it's better keeping it like this.

> BTW, in a mail thread we're discussing renaming to osmo-ggsn... but that's still premature.

Indeed, once it's moved openggsn -> osmo-ggsn we will rename it in osmo-gsm-tester too.

> Have you thought about the tunnel creation?
> In my usual test scripts I have to sudo ggsn so that it is allowed to create a tunnel for the GPRS NAT thang.
> On the tester we have so far been able to run everything as non-root, and I dimly remember Harald had a comment on how the tunnel problem should be solved...
> IIRC it entails having a tunnel readily configured on the system and letting ggsn use that from user land.
> Should we have a number of tunnels and treat them as resources?
> I don't know much on the subject, just want to make sure you're aware of the need to clarify this.

Remember too that this is not the only permissions issue we are facing here. On top of it, according to #2308, I plan to run at least openggsn into its own network namespace (but I'd like to move all osmo-* processes running in the main unit too), be it specific per test/suite/trial, I still need to think about this. Changing namespace of a process as far as I know requires root permissions (see: man 2 setns). We can workaround that by using sudo or suid with the nsenter binary for instance, or writing a small suid C binary which launches "ip netns exec $netns $command" and does any required extra check to allow subset of needed commands to be launched. Then wrap that into python function such as foo.run_process_netns(netns, command, env). I think with nsenter you can change the UID in which the user runs, but in case of ip netns exec as you run it as root, then user running the command inside the netns is root, and you need tricks like "sudo ip netns exec $netns sudo -u $user $command", which means we can even create tunnel devices dynamically too because we are root at some point anyway. Or if ggsn supports running as root to create the tun devices and then drop privileges, we can use that too (I guess it can't because then it couldn't clean the tunnels once it finishes?).

> Otherwise the patches make the impression that you've already run the setup successfully (impressed). Is that the case? Have you run it as root or in user land?

I'm not running the full setup end to end yet, as I know there are still some permissions requirements still pending to be able to run it (as discussed above). So far I was able to:
- Run the complete setup end to end working successfully with sysmobts+gobi200 on my local PC, with manual routing rules and moving the modem net iface to a different net namespace (problem is that afterwards you need to move back the iface and re-plug the modem to be able to use it again through ofono due to issues mentioned in #2308).
- Within osmo-gsm-tester, run osmo-pcu inside the sysmobts and it connects successfully to osmo-sgsn. Osmo-sgsn and openggsn are started and stopped when the test ends (probably tun dev is not created).

So, anyway I need this bits running in order to do some tests and see better what is needed. Then I can for instance run osmo-gsm-tester -s interactive -t debug to have everything set up and do the missing system configurations manually step by step, do tests then implement it to be done automatically by osmo-gsm-tester.

--------Reply to Harald's comments----------

> yes, that's correct. It's the same with any other software that uses
> tun devices: You can create persistent tun/tap devices using e.g. tunctl
> from the shell (from root/CAP_NET_ADMIN) e.g. at system start or by the
> systemd service before dropping privileges and starting the process.
> The user/group owner of the persistent tun/tap device is set at its
> creation time.

I didn't know about tunctl, I'll have a look at it and check the openggsn branch you mentioned. There's some part I don't really understand right now though. As far as I know only openggsn is creating a tun device, but what about osmo-sgsn? I think it doesn't create one. How it is expected to do the de/encapsulation of packets coming/going to the ggsn?

> > Should we have a number of tunnels and treat them as resources?
>
> Probably makes sense given how osmo-gsm-tester works, though I'm not the
> expert on that. Each tun device then consists of the device name, as

> well as the network/netmask to which it is configured.

Ok it then probably makes sense and it's easier to create them at startup time, and then handle them as resources in osmo-gsm-tester.

#13 - 09/04/2017 11:02 AM - laforge

- File openggsn.cfg added

On Mon, Sep 04, 2017 at 09:31:38AM +0000, pespin [REDMINE] wrote:

> > yes, that's correct. It's the same with any other software that uses
> > tun devices: You can create persistent tun/tap devices using e.g. tunctl
> > from the shell (from root/CAP_NET_ADMIN) e.g. at system start or by the
> > systemd service before dropping privileges and starting the process.
> > The user/group owner of the persistent tun/tap device is set at its
> > creation time.
>
> I didn't know about tunctl, I'll have a look at it and check the
> openggsn branch you mentioned.

it's super easy.

creating the persistent device:

```
sudo tunctl -3 -u pespin -g pespin -t gtp0
```

deleting it:

```
sudo tunctl -3 -d gtp0
```

The important bit is the '-3', as it says "tun". otherwise it will
create tap by default, which is incompatible.

In osmo-ggsn you then simply use the same device name ("gtp0") and tell
it to not attempt any ifconfig on it. This means no "ip ifconfig" or
"ipv6 ifconfig" lines in the config.

Example osmo-ggsn config file with multiple APNs attached. Please note
this example uses "ifconfig" and hence requires root/CAP_NET_ADMIN. but
you can simpl remove those lines.

> There's some part I don't really understand right now though. As far
> as I know only openggsn is creating a tun device, but what about
> osmo-sgsn? I think it doesn't create one. How it is expected to do the
> de/encapsulation of packets coming/going to the ggsn?

The SGSN is not decapsulating the packets and exposing them to a local
IP stack. It is transparent to the user IP level. The first IP stack
processing the user IP packet is the stack in the MS/UE.

The IP tunnel is between MS and GGSN, anythin in between (BTS/PCU/SGSN/...) is transparent.

> Should we have a number of tunnels and treat them as resources?

> Probably makes sense given how osmo-gsm-tester works, though I'm not the expert on that. Each tun device then consists of the device name, as well as the network/netmask to which it is configured.

> Ok it then probably makes sense and it's easier to create them at startup time, and then handle them as resources in osmo-gsm-tester.

correct. But then, when you're spawning the ggsn in a new netns, you could also simply map your local user (e.g. pespin) to root inside the netns. At that point you can create/change network devices inside the netns. See e.g. "unshare -n -U -r". you then have CAP_NET_ADMIN inside that netns, but in reality to the host you're still only the non-root user.

#14 - 09/04/2017 12:19 PM - neels

pespin wrote:

> I'd prefer keeping the name openggsn as it directly points you the project

Well, when the git trees are called openbsc and libosmo-sccp, we're still not going to log "Starting openbsc" or "Starting libosmo-sccp" but "Starting {osmo-nitb,osmo-msc,osmo-stp}".
So I'd still prefer logging the name of the actual program being launched =)
But not really important, have it your way.

> Have you thought about the tunnel creation?

> Remember too that this is not the only permissions issue we are facing here.

Until now it is, AFAICT.

> On top of it, according to #2308, I plan to run at least openggsn into its own network namespace

That's a new feature that isn't strictly required, right?

> We can workaround that by using sudo

Yes, we can allow particular commands to be run by sudo, as a last resort.
Preferably we don't have to give **anything** root privileges, given that we're running code coming from "somewhere" that is being tested whether it works...

> which means we can even create tunnel devices dynamically

yes ... but rather use tuns that are already configured on the system, since it allows completely dropping the sudo needed otherwise.

#15 - 09/04/2017 12:25 PM - neels

laforge wrote:

> correct. But then, when you're spawning the ggsn in a new netns, you
> could also simply map your local user (e.g. pespin) to root inside
> the netns. At that point you can create/change network devices inside
> the netns.

hmm, if that is even easier, we might as well drop the handling-tun-as-resource part and let openggsn create its own tun **in its own netns**.
But we'd need sudo to create a new netns, right?
...decisions... I'll let you figure it out.

(BTW, for the tester it should pivot on the group 'osmo-gsm-tester', not a particular user name)

#16 - 09/04/2017 12:51 PM - pespin

> Remember too that this is not the only permissions issue we are facing here.

Until now it is, AFAICT.

> On top of it, according to #2308, I plan to run at least openggsn into its own network namespace

That's a new feature that isn't strictly required, right?

It may be not explicitly described in #2308, but according to my research with manual setup of the GPRS chain, we need to use namespaces if we are running all components in the same system, as it's not possible/easy to route traffic correctly 1st through the modem interface, and then route it to somewhere else (internet or specific server) when it gets out of the GGSN.

Have a look at https://osmocom.org/issues/2308#note-9 and comments after it.

#17 - 10/19/2017 12:06 AM - neels

- Blocks Feature #2582: osmo-gsm-tester: add test case: dynamic timeslot data usage added

#18 - 11/23/2017 10:48 AM - laforge

- Priority changed from Urgent to High

#19 - 11/23/2017 04:52 PM - pespin

- % Done changed from 20 to 50

I pushed support for GPRS signaling (no IP data sent through ofono iface yet, but attach and activate context works) to gerrit:

```
remote:   https://gerrit.osmocom.org/4996 hlr: Rename conf_for API
remote:   https://gerrit.osmocom.org/4997 modem: Abstract the list of required ofono interfaces
remote:   https://gerrit.osmocom.org/4998 modem: Move power off sequence to separate method
remote:   https://gerrit.osmocom.org/4999 resources.conf: Add gprs feature for EC20 modem
remote:   https://gerrit.osmocom.org/5000 defaults.conf: Add PDCH channels to test GPRS
```

```
remote:   https://gerrit.osmocom.org/5001 Add OsmoGgsn class
remote:   https://gerrit.osmocom.org/5002 Add OsmoSgsn class
remote:   https://gerrit.osmocom.org/5003 Add class OsmoPcuSysmo
remote:   https://gerrit.osmocom.org/5004 Add OsmoPcu class
remote:   https://gerrit.osmocom.org/5005 pcu_osmo: workaround osmo-pcu failing if bts not ready
remote:   https://gerrit.osmocom.org/5006 OsmoPcuSysmo: Integrate  with Sysmobts and OsmoSgsn
remote:   https://gerrit.osmocom.org/5007 OsmoBtsTrx: Integrate with OsmoPcu and OsmoSgsn
remote:   https://gerrit.osmocom.org/5008 OsmoBtsOctphy: Integrate with OsmoPcu and OsmoSgsn
remote:   https://gerrit.osmocom.org/5009 modem: Add minimal GPRS support
remote:   https://gerrit.osmocom.org/5010 suites: aoip_debug: Start GPRS services
remote:   https://gerrit.osmocom.org/5011 suites: gprs: Introduce suite with ping test
remote:   https://gerrit.osmocom.org/5012 default-suites.conf: Use same order for sysmocell5000 as for other b
ts
remote:   https://gerrit.osmocom.org/5013 default-suites.conf: Add gprs suite
```

What is missing after it becomes merged:
- Get iface information from Context.GetProperotes (already printing those to log), and apply IP and routing to the network interface provided there, probably by some bash script in the main unit which can be enabled to be called by sudo in /etc/sudoers. For instance, all traffic to 1.1.1.1 must go through that ofono iface.
- Create a network namespace from osmo-gsm-tester for a specific suite/test, still to be seen if created dynamically or statically as root. Create a virtual iface (ip link add veth0 type veth peer name veth1) and move one of the pairs inside the namespace, so osmo-ggsn can connect with other processes out of the namespace (osmo-sgsn). Then inside the namespace create a loopback iface with ip 1.1.1.1.
- Create a new python class "TCPServer" or similar, which launches "netcat -l 1.1.1.1" inside the net namespace. We can also run statically a http server there or something similar.

#20 - 05/05/2018 06:22 PM - neels

- Blocks Feature #3239: test dyn TS: switch to voice and back to data, ensure GPRS still works, and ensure switching back to voice still works added

#21 - 05/17/2018 11:40 AM - laforge

- Status changed from In Progress to Stalled

six months no update, but still "in progress"? Please make sure you take the "ticket review monday" more serious, thanks!

#22 - 08/06/2018 08:44 PM - laforge

- Priority changed from High to Urgent

This is dragging on too long, I would love to see this completed finally.

lynxis recently stated he has solved the "many ofono instances / namespace" problem somehow? Maybe he can help?

#23 - 08/27/2018 09:11 AM - pespin

Waiting for working example of setup proposed for lynxis to give it a try.

#24 - 09/18/2018 10:03 AM - pespin

- Blocks deleted (Feature #2582: osmo-gsm-tester: add test case: dynamic timeslot data usage)

#25 - 09/18/2018 10:05 AM - pespin

- Blocks deleted (Feature #3239: test dyn TS: switch to voice and back to data, ensure GPRS still works, and ensure switching back to voice still works)

#26 - 10/29/2018 01:29 PM - pespin

- Status changed from Stalled to Resolved

- % Done changed from 50 to 100

Support for GPRS data services has been added, see #2308. Closing this ticket.

Files

openggsn.cfg	1.59 KB	09/04/2017	laforge