OsmoGSMTester - Bug #2413

osmo-gsm-tester: smpp: ESME traffic not logged in nitb pcap file

07/31/2017 03:32 PM - pespin

Status:	Closed	**Start date:**	07/31/2017
Priority:	Low	**Due date:**	
Assignee:	pespin	**% Done:**	0%
Category:			
Target version:			
Spec Reference:			

Description

We need to make sure this information is stored correctly in order to debug SMPP issues more easily.

After a quick check, I couldn't find why this conn is not in the pcap file. The filter applied to tcpdump seems to be OK, it filters for same host which is used by the ESME to connect, but there's not TCP connection port=2775 to it on the pcap file.

Test checked which contains the issue (other may probably too): smpp/esme_ms_sms_storeforward.py

Related issues:

Related to OsmoNITB - Bug #2414: SMSC: deliverSM message with no user_message...	**New**	**07/31/2017**
Related to OsmoNITB - Bug #2429: SMSC: deiverSM message with bad user_message...	**Closed**	**08/10/2017**

History

#1 - 07/31/2017 03:33 PM - pespin

- Assignee set to osmo-gsm-tester

#2 - 07/31/2017 03:41 PM - pespin

- Description updated

#3 - 07/31/2017 04:08 PM - pespin

- Related to Bug #2414: SMSC: deliverSM message with no user_message_reference added

#4 - 08/10/2017 10:02 AM - pespin

It seems the traffic can be sniffed through "lo" interface with "tcpdump -n -vv -i lo tcp and port 2775". However, it doesn't appear on the interface osmo-nitb is listening to (eth1:0, osmo-nitb listens on 10.42.42.2), or in the main one (eth1, 10.42.42.1):

```
~# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0d:b9:35:3a:41 brd ff:ff:ff:ff:ff:ff
    inet 10.42.42.1/24 brd 10.42.42.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet 10.42.42.2/24 brd 10.42.42.255 scope global secondary eth1:0
       valid_lft forever preferred_lft forever
    inet 10.42.42.3/24 brd 10.42.42.255 scope global secondary eth1:1
       valid_lft forever preferred_lft forever
    inet 10.42.42.4/24 brd 10.42.42.255 scope global secondary eth1:2
       valid_lft forever preferred_lft forever
    inet 10.42.42.5/24 brd 10.42.42.255 scope global secondary eth1:3
       valid_lft forever preferred_lft forever
    inet 10.42.42.6/24 brd 10.42.42.255 scope global secondary eth1:4
       valid_lft forever preferred_lft forever
    inet6 fe80::20d:b9ff:fe35:3a41/64 scope link
       valid_lft forever preferred_lft forever
```

```
11:58:26.032946 IP (tos 0x0, ttl 64, id 13046, offset 0, flags [DF], proto TCP (6), length 60)
```

```
    10.42.42.1.46654 > 10.42.42.2.2775: Flags [S], cksum 0x6885 (incorrect -> 0xcf10), seq 1934627237, win 436
90, options [mss 65495,sackOK,TS val 40444755 ecr 0,nop,wscale 10], length 0
11:58:26.033013 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 60)
    10.42.42.2.2775 > 10.42.42.1.46654: Flags [S.], cksum 0x6885 (incorrect -> 0xb0b6), seq 1266789643, ack 19
34627238, win 43690, options [mss 65495,sackOK,TS val 40444755 ecr 40444755,nop,wscale 10], length 0
11:58:26.033070 IP (tos 0x0, ttl 64, id 13047, offset 0, flags [DF], proto TCP (6), length 52)
    10.42.42.1.46654 > 10.42.42.2.2775: Flags [.], cksum 0x687d (incorrect -> 0x8429), seq 1, ack 1, win 43, o
ptions [nop,nop,TS val 40444755 ecr 40444755], length 0
```

Not sure if we can modify the routing to have it show up in the osmo-nitb interface instead of going through lo.

#5 - 08/10/2017 10:38 AM - pespin

- Related to Bug #2429: SMSC: deiverSM message with bad user_message_reference added

#6 - 08/10/2017 11:22 AM - pespin

osmo-nitb is only listening on 10.42.42.2:2775

```
# netstat -l -n -t -p
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
...
tcp        0      0 10.42.42.2:2775         0.0.0.0:*               LISTEN      11367/osmo-nitb
...
```

And here's the actual ESME<->NITB connection:

```
tcp        0      0 10.42.42.1:46953        10.42.42.2:2775         ESTABLISHED 11352/python3
```

#7 - 08/10/2017 11:32 AM - pespin

Looks like the "local" routing table sets the conn to be used locally, for any conn going from the master iface (10.42.42.1, eth1) to the secondary ones (10.42.42.2, eth1:*):

```
# ip route show table local
local 10.3.0.17 dev tun0  proto kernel  scope host  src 10.3.0.17
broadcast 10.9.25.0 dev eth0  proto kernel  scope link  src 10.9.25.101
local 10.9.25.101 dev eth0  proto kernel  scope host  src 10.9.25.101
broadcast 10.9.25.255 dev eth0  proto kernel  scope link  src 10.9.25.101
broadcast 10.42.42.0 dev eth1  proto kernel  scope link  src 10.42.42.1
local 10.42.42.1 dev eth1  proto kernel  scope host  src 10.42.42.1
local 10.42.42.2 dev eth1  proto kernel  scope host  src 10.42.42.1
local 10.42.42.3 dev eth1  proto kernel  scope host  src 10.42.42.1
local 10.42.42.4 dev eth1  proto kernel  scope host  src 10.42.42.1
local 10.42.42.5 dev eth1  proto kernel  scope host  src 10.42.42.1
local 10.42.42.6 dev eth1  proto kernel  scope host  src 10.42.42.1
broadcast 10.42.42.255 dev eth1  proto kernel  scope link  src 10.42.42.1
broadcast 10.64.1.0 dev eth2  proto kernel  scope link  src 10.64.1.1
local 10.64.1.1 dev eth2  proto kernel  scope host  src 10.64.1.1
broadcast 10.64.1.255 dev eth2  proto kernel  scope link  src 10.64.1.1
broadcast 127.0.0.0 dev lo  proto kernel  scope link  src 127.0.0.1
local 127.0.0.0/8 dev lo  proto kernel  scope host  src 127.0.0.1
local 127.0.0.1 dev lo  proto kernel  scope host  src 127.0.0.1
broadcast 127.255.255.255 dev lo  proto kernel  scope link  src 127.0.0.1
broadcast 192.168.1.0 dev eth2  proto kernel  scope link  src 192.168.1.200
local 192.168.1.200 dev eth2  proto kernel  scope host  src 192.168.1.200
broadcast 192.168.1.255 dev eth2  proto kernel  scope link  src 192.168.1.200

# ip route get 10.42.42.2
local 10.42.42.2 dev lo  src 10.42.42.1
    cache <local>
```

#8 - 08/10/2017 04:36 PM - pespin

- Assignee changed from osmo-gsm-tester to pespin

#9 - 08/25/2017 12:51 PM - pespin

This is starting to become annoying as I see the same issue appears for traffic in MSC pcap file, and I think the same may happen with some osmo-bts-trx / osmo-trx traffic.

I think it basically happens on any connection which is started with the main eth address (10.42.42.1), basically that one is selected automatically by kernel/routing when no speciifc bind() is done before connect() is called.

We should change the network setup/config inside the host + document it in order to be able to record all traffic using the specific interface with the specific IP address.

#10 - 10/29/2017 06:33 PM - laforge

- Priority changed from Normal to Low

#11 - 11/20/2017 04:52 PM - pespin

- Status changed from New to Resolved

Fixed by recording on "any" interface and relying on filtering based on IP. See https://gerrit.osmocom.org/#/c/4937/

#12 - 02/06/2018 08:26 AM - laforge

- Status changed from Resolved to Closed