

2.1.15 Enable Measurement reports

The reporting of the measurement results needs to be enabled through the *trap-register* node. After it has been enabled, measurement TRAPs will be generated once measurement reports arrive.

```
$ ./bsc_control.py -d localhost -p 4249 -s trap-register measurement-report.v1  
Got message: SET_REPLY 1 trap-register Dummy implementation
```

2.1.16 Measurement Notification

There will be a TRAP for each measurement report that is received from the BTS. The information available from the *struct gsm_meas_rep* will be emitted as a JSON string. Depending on the state of the radio connection the IMSI/TMSI might not yet be available. The variable for the TRAP is *bts.NR.trx.NR.measurement-report.v1* and the JSON document has the following entries:

```
{  
  "timeslot" : TS_NR,  
  "subslot" : Subslot_NR,  
  "tmsi" : tmsi,  
  "imsi" : "IMSI",  
  "report_nr" : Nr,  
  "flags" : data_indicating_available_flags, "ul" : {  
    "full" : {"level" : Number, "quality" : Number},  
    "sub" : {"level" : Number, "quality" : Number}  
  }, "dl" : {  
    "full" : {"level" : Number, "quality" : Number},  
    "sub" : {"level" : Number, "quality" : Number}  
  },  
  "bs_power" : NR (optional),  
  "ms_timing_offset": NR (optional),  
  "ms_power" : NR (in dbM),  
  "ms_ta" : TimingAdvance,  
  "neighbors" : [  
    {"rxlevel" : Nr, "bisc" : Nr, "index" : Nr,  
     "arfcn" : ARFCN, "flags" : flags},  
    ...  
  ]  
}
```

3.3 Registering for measurement trap and generating it

The OpenBSC code has the concept of signals and signals handlers. A signal is associated with a subsystem and has a number. A signal handler can register a callback to be called for

a specific subsystem. The `src/libbsc/abis_rsl.c:rsl_rx_meas_res` is sending a signal for every measurement report that is received within the system.

NITB needs to be modified to:

1. Create a new file for the registration handling.
2. Register to the `SS_LCHAN` subsystem.
3. Keep track of which control connections have requested which notifications.
4. Convert and forward the signal to the registered connections.
5. Make sure the registration is cleared when the connection is closed.

The new file should be called `src/osmo-nitb/ctrl_trap_handling.c`. The code should register for the to be monitored subsystems. The registration should be done from within an `init` function that is called from the main method of the `osmo-nitb` application. Inside this file there could be a global pointer that is initialized by the `init` function. It would hold a list anchor for the list of registered control connections. Whenever a command is sent to enable or disable the notification an entry in the list will be searched and either updated or created and inserted. The control connection closed callback functionality can be used to make sure that the entry is removed when the control connection is gone.