

# Osmocom CTRL protocol

## OsmoCon 2017

Max Suraev  
msuraev@sysmocom.de

sysmocom - s. f. m. c. GmbH

April 21, 2017

# Agenda

- 1 Intro
- 2 Tech details
- 3 Usage examples
- 4 Future perspectives

## Overview

- Intention: “CTRL for programs” like “VTY for humans”

# Overview

- Intention: “CTRL for programs” like “VTY for humans”
- Python and C implementations

# Overview

- Intention: “CTRL for programs” like “VTY for humans”
- Python and C implementations
- SNMP-like protocol

# Overview

- Intention: “CTRL for programs” like “VTY for humans”
- Python and C implementations
- SNMP-like protocol
- SET/GET/TRAP

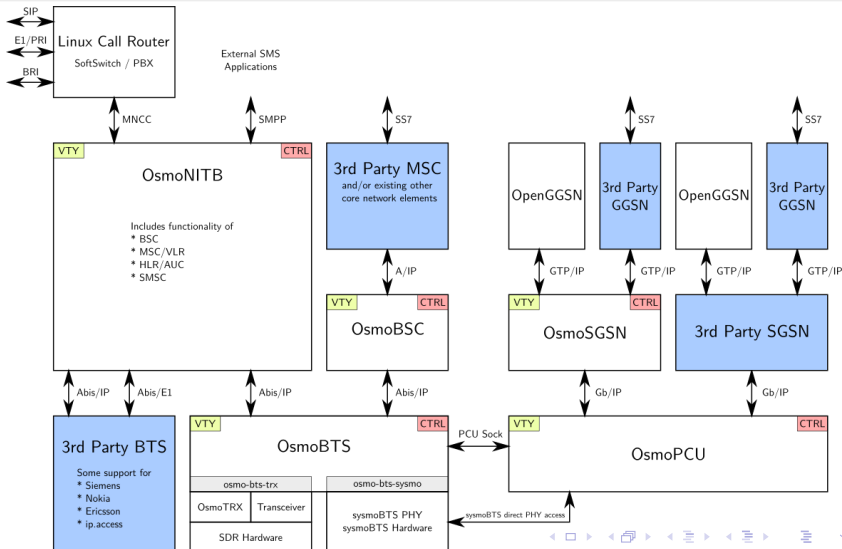
## Overview

- Intention: “CTRL for programs” like “VTY for humans”
- Python and C implementations
- SNMP-like protocol
- SET/GET/TRAP

### Security

None: it's the task of other layers.

# The bigger picture





## Random facts

- Binary framing + text based payload
- IPA multiplex based (0xEE 0x00)
- Total 6 messages (both client and server)
- RO, RW, WO variables

```
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 4250, Dst Port: 56330, Seq: 960, Ack: 1, Len: 121
▼ IPA protocol ip.access, type: OSMO EXT
  DataLen: 118
  Protocol: OSMO EXT (0xee)
  Osmo ext protocol: CTRL (0x00)
  CTRL data: TRAP 0 net.0.bsc.188.bts.0.location-state 1480410363,invalid,0.000000,0.000000,0.000000,operational,unlocked,
```

## Informal spec

| Client | Server            |
|--------|-------------------|
| SET    | SET_REPLY / ERROR |
| GET    | GET_REPLY / ERROR |
|        | TRAP              |

Table: Request/Response messages

More details in Sec. "Osmocom Control Interface":  
[http://ftp.osmocom.org/docs/latest/\\*-usermanual.pdf](http://ftp.osmocom.org/docs/latest/*-usermanual.pdf)

## Command example

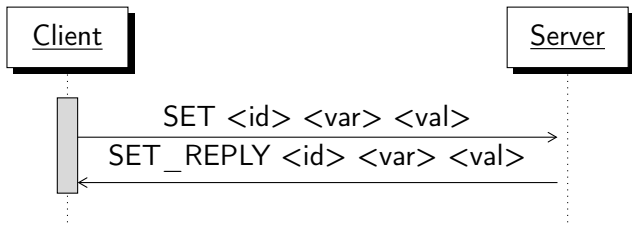


Figure: Set **<var>** variable to **<val>** value (**<id>**  $\neq$  0 is unique operation identifier).

## Send command in Python

---

```
def do_set_get(sck, var, value = None):  
    (r, c) = Ctrl().cmd(var, value)  
    sck.send(c)  
    answer = Ctrl().rem_header(sck.recv(4096))  
    return (answer,) +  
           Ctrl().verify(answer, r, var, value)
```

---

# Trap command in C

```
void osmo_bsc_send_trap(struct ctrl_cmd *cmd,
                       struct bsc_msc_connection
                           *msc_con)
{
    struct ctrl_cmd *trap;
    struct ctrl_handle *ctrl;
    struct bsc_msc_data *msc_data;

    msc_data = (struct bsc_msc_data *)
                msc_con->write_queue.bfd.data;
    ctrl = msc_data->network->ctrl;
    trap = ctrl_cmd_trap(cmd);
    ctrl_cmd_send_to_all(ctrl, trap);
    ctrl_cmd_send(&msc_con->write_queue, trap);

    talloc_free(trap);
}
```

## Useful CTRL variables

- `rate_ctr.*`
  - Rate counter value, **RO**
- `fsm.FSM_NAME.id.INSTANCE_ID.state/timer/...`
  - FSM introspection
- `enable-ps/disable-ps`
  - Control GPRS services per-IMSI

## Single source of truth

Ideally we'd like to have same “knobs” exposed via both *vtty* and *ctrl* using single definition.

- RFC 6241 NETCONF
- RFC 6020 YAML
- Your idea?

## Final slide

### Demo time

Let's actually use it for good.

### Q & A

If your question have not made it into this part - grab me afterwards or ask at [openbsc@lists.osmocom.org](mailto:openbsc@lists.osmocom.org).



osmocom