Fault Tolerant GSM Core Network

<keith@rhizomatica.org>

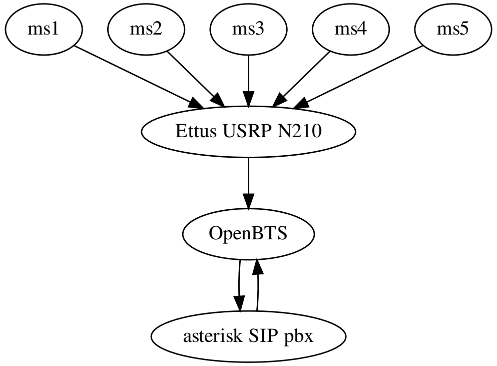
A Fault tolerant Core network - Distributed GSM

How do you run a GSM network with unreliable back-haul, maintaining as many local services as possible available during periods of back-haul outage?

1. How Rhizomatica originally setup the autonomous GSM cell
2. Failure: multi-master databases.
3. Proposal / Status

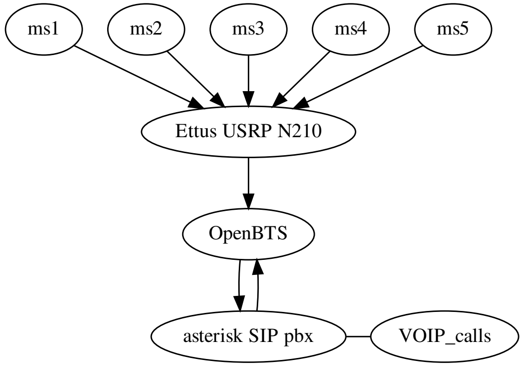
Original Requirements TIC a.c. Networks in Oaxaca.

- Self-contained Network.
  - Original experiment was with OpenBTS and asterisk to simply loop back calls (2011)
  - Inaccurate maybe to call it a "network"
    - Concept was to have one tower, to support local calling/SMS using existing UE
  - Full autonomy in each local site.
    - politically as much as technically!



Original Requirements TIC a.c. Networks in Oaxaca.

- Self-contained Network.
  - Original experiment was with OpenBTS and asterisk to simply loop back calls (2011)
  - Inaccurate maybe to call it a "network"
    - Concept was to have one tower, to support local calling/SMS using existing UE
  - Full autonomy in each local site.
    - politically as much as technically!



Moving to Osmo

- Change to Osmo Stack
    - Replace Asterisk with FreeSwitch
    - A python based dialplan for freeswitch with a postgreSQL backend
        - PostgreSQL database contains fields we need that are not in Osmo-HLR (balance, for ex.) Also the entire HLR, plus last known location.
        - This then handles most of the functions of the "core network"
            - Commissioning
            - Authorisation
            - Routing
            - Billing
    - RAPI - A REST API
    - Kannel (as a bridge between SMPP and RAPI)
    - LCR (as the MNCC <-> SIP bridge)
    - All controlled by a autonomous web based front end. (in the village)
        - Called "RAI" Rhizomatica Administration Interface

Osmocom based RCCN

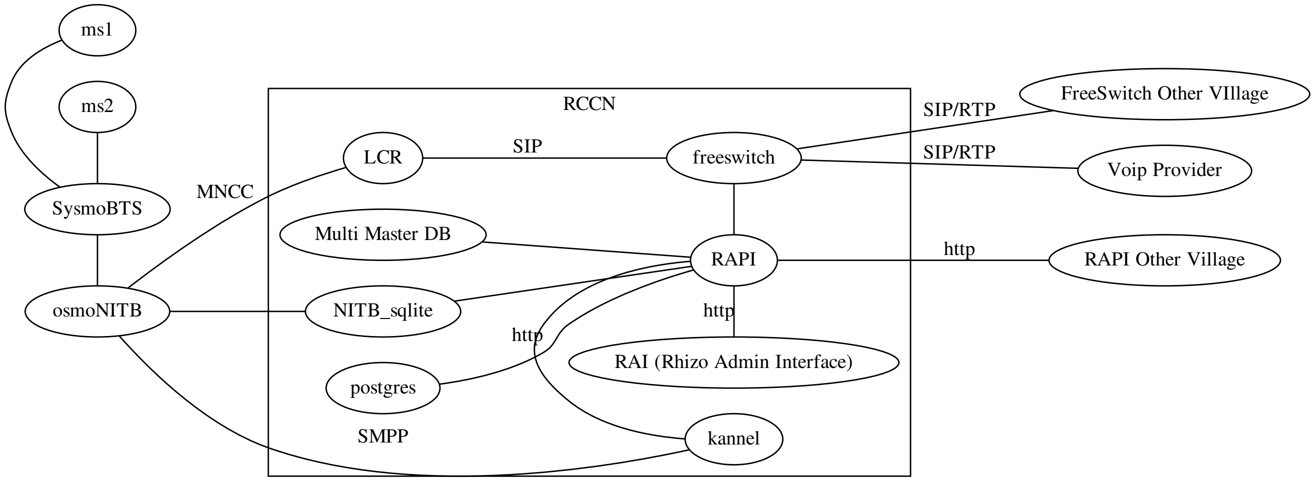
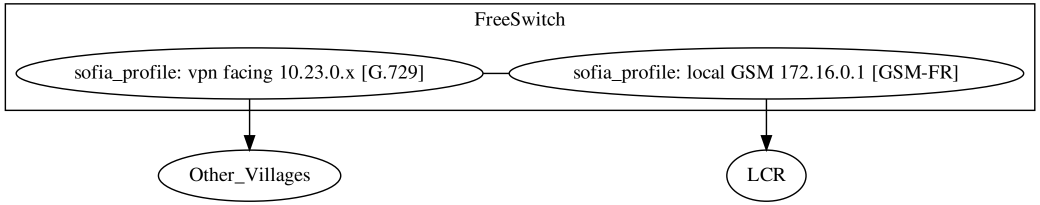


Figure 1. Rhizomatica Community Cellular Network.

Implement Inter Village Communication

- Rhizomatica Numbering: [POSTCODE]-[PBX]-[EXTENSION]
  - example: 68068-1-12345
- Post code to IP address mapping **sip:[code][pbx][exten]@[ip]:[port]**
  - Mapping implemented in a multi-master distributed database
  - Latest known user location also in same multi-master database.


FreeSwitch
sofia_profile: vpn facing 10.23.0.x [G.729]
sofia_profile: local GSM 172.16.0.1 [GSM-FR]
Other_Villages
LCR

Inter Village VPN

A variety of IP "internet" connections possibilities in villages

- VSAT
- Local WISP
- Direct ISP (ADSL, Cable)

.

- We use the tinc VPN in switch mode in all villages. This is over the "internet"
- TINC can get through n+ levels of NAT. (I think the most we have seen is 3)
- We could (maybe) work with the local WISP provider to punch holes in their NATS for direct tinc-tinc data between villages. (or they could use IPv6)
  - but mostly, traffic ends up going at best through the providers office, at worst, to Texas, USA and back.
- TINC in switch mode is like an ethernet switch; broadcast works, ARP works. tradeoff - it can be noisy.
- TINC in router mode also supports multicast.

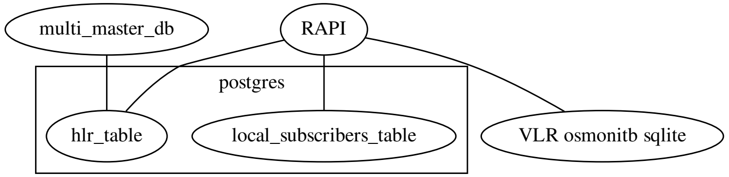
Roaming

"roaming"

Not really "roaming" - All our networks are using the same MCC-MNC.

Originally is was not possible for a user to physically move from one village to another faster than the LUR timeout. As the towers became more numerous, we have places (mountain ridge for example) where phones would ping-pong from one system to another.

- Maintain details of UE location in a multi-master database, running a node on each BSS
  - Cronjobs run to sync content. Originally by polling osmo-nitb sqlite, which introduced errors, Later by reacting to LUR triggered SMPP alert-notification messages.
- Cronjob searches the local osmo HLR for subcriber-create(d)-on-demand entries.
  - If the IMSI is known in the master HLR we authorise it
  - Obviously only works if the unauthorised IMSI can attach in the 1st place (accept-all mode) :((
  - Local OsmoNITB HLR is then a kind of VLR



Handling Billling for "Roaming"

- If the User is "roaming", for billable calls the local freeswitch routes the call to the user’s home freeswitch.
- Home freeswitch (and python dialplan/RAPI) handles auth check + billing.
- If the home system is down/unavailable obviously the roaming user has no service to PSTN
  - Not exactly fault tolerant. :((
- Possible Solution: Implement a centralised billing cache that syncs later with local system.

Multi Master Databases.

Tried various products, all appear problematic with unstable links. They are designed to run in data centres.

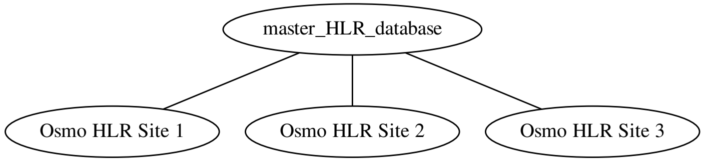
- All seemed unworkable.
- Even if they are are not, will they scale? to 100, 1000’s of nodes?
  - Spend some effort on this, maybe not sufficient, but enough.
  - We don’t need them, so let’s not bother to find out.

Multicast Request for UE location.

- A proposal from Harald.
- Flood the network with a request for IMSI location.
- Any local HLR that has seen that IMSI recently will respond with the timestamp of the most recent LUR
- Route the call to the winner (the most recent)
- Indeed, one can also have the SIP UA branch the call to all responders, thereby paging on all BTSs. If one branch answers the SIP UA will cancel the other branches.

A centralised HLR

- So, what about our distributed database of known IMSIs?
- Not so time sensitive
  - When subscribers are (de)commissioned, some delay while this propagates to all local HLR is acceptable.
- Maybe do something like what is existing (cronjob sync)

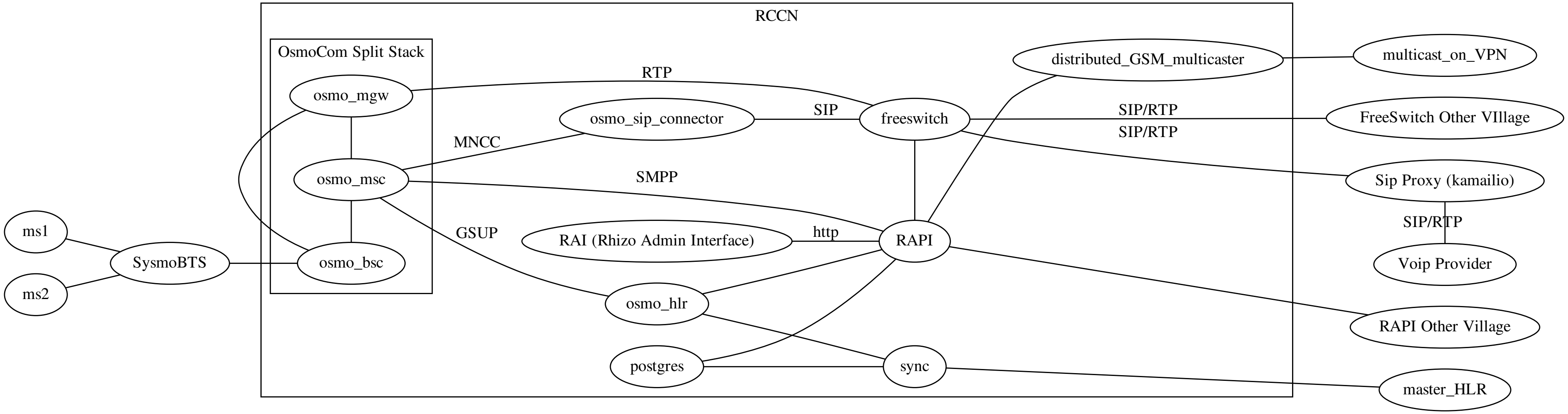


- Keep all local osmo-hlr databases populated with all IMSIs
  - No more accept-all mode :))

2018 Status

RCCN (II) work in progress.

- Replace Kannel by implementing an ESME inside RAPI (using python-smpplib)
    - RAPI could also use a full rewrite.
    - Osmo SIP connector not production ready.

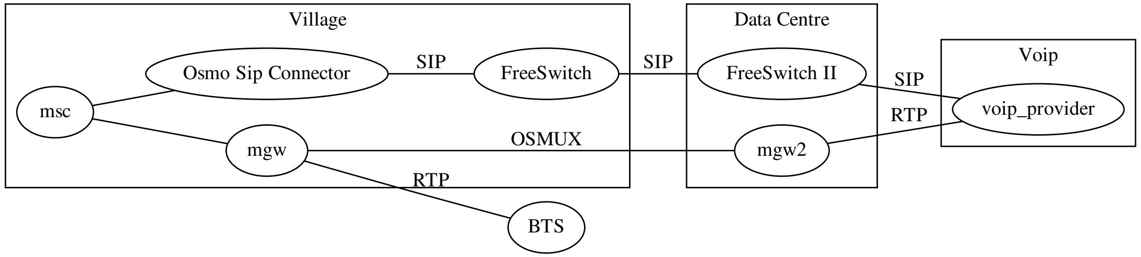


Future

- We have:
    - Load Balancing HO (to optimise SysmoBTS 2050)
    - USSD
    - Everything else. (direct BTS to BTS LCLS)
    - Thank you!
- If towers become more dense, build associated community owned backhaul.
    - Star wifi network or mesh network.
        - In this case we might use locally centralised BSC to benefit from active call HO
- Essentially we are still "emulating" the NITB in every village + SIP based core network.
- For VSAT links we need OsMux for RTP, but how to implement this remains unclear in a "fault tolerant core network.."

Osmux

- Implement OsMux signalling



- add some kind of transparent mode in MGW some kind of ip:port mangling.
- rfc8108
- No space for signalling in the stream itself.
- drop freeswitch and the rccn altogether and rethink everything
- fallback to a local MSC if uplink is down?

Questions / Suggestions / Discussion

A Fault tolerant Core network - Distributed GSM