The Iuh protocol stack and osmo-iuh

Implementing HNBAP, RUA and RANAP in Free Software

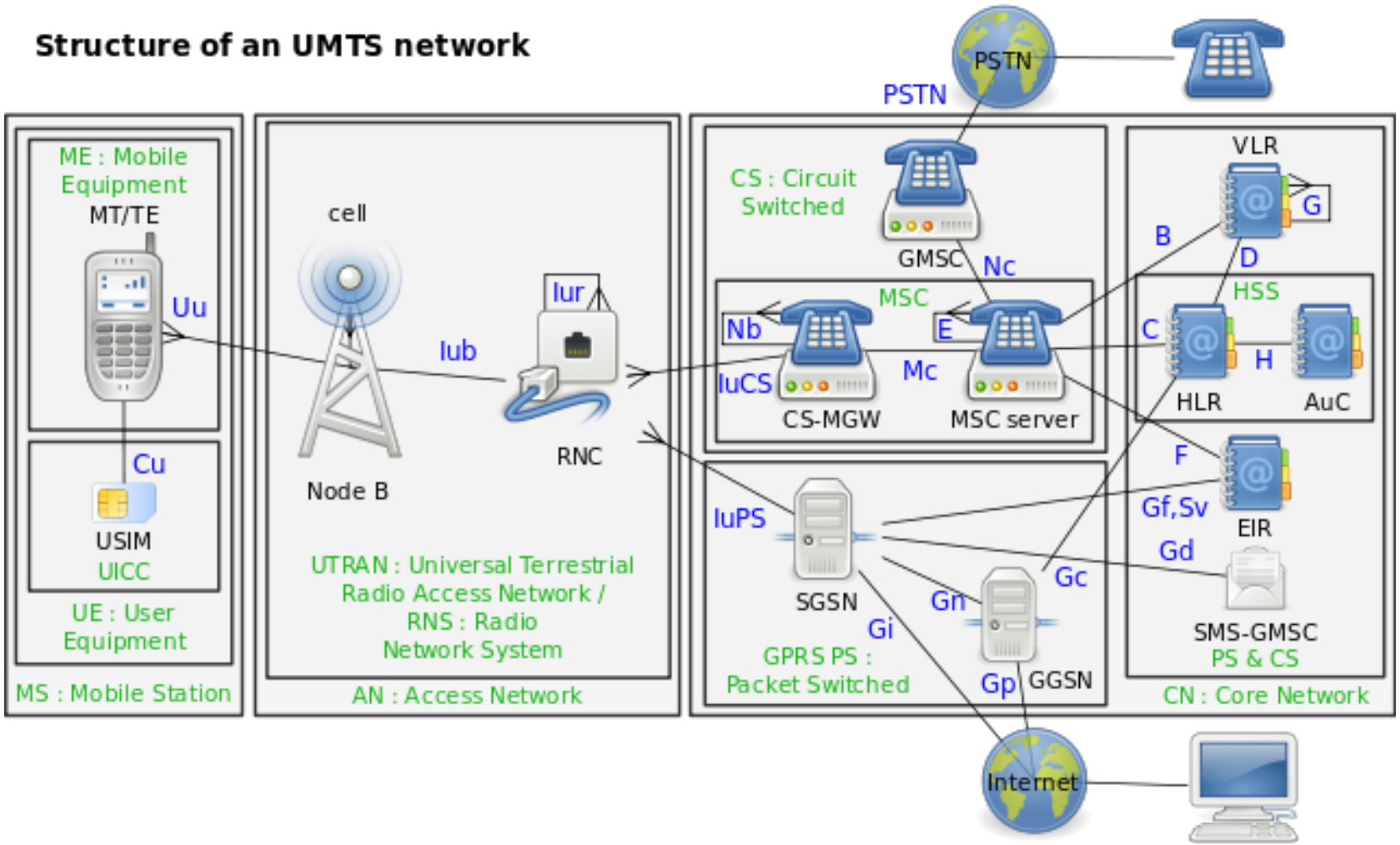
Harald Welte

Osmocom / sysmocom GmbH

About

- Linux Kernel / bootloader / driver / firmware developer since 1999
- Former core developer of Linux packet filter netfilter/iptables
- Comms / Network Security beyond TCP/IP
  - OpenPCD, librfid, libmtrd, OpenBeacon
  - deDECTed.org project
  - Openmoko - FOSS smartphone with focus on security + owner device control
  - OpenBSC as network-side FOSS GSM Stack
  - OsmocomBB - device-side GSM protocol stack + baseband firmware
- practical security research / testing on baseband side and telecom infrastructure side
- running a small team at sysmocom GmbH in Berlin, building custom tailored mobile communications technology

UMTS Architecture

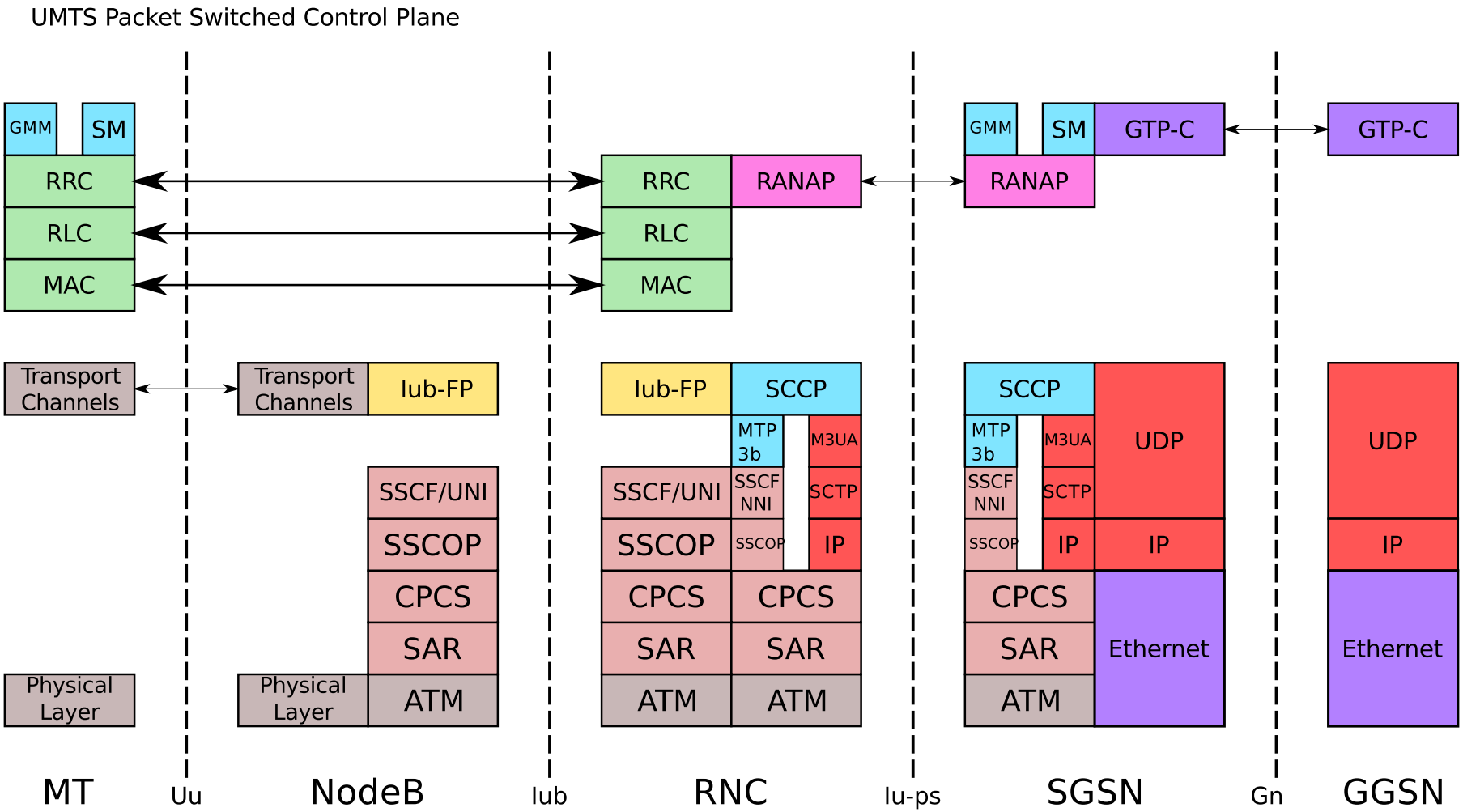


UMTS Structure by Tsaitgaist - icons from Gnome

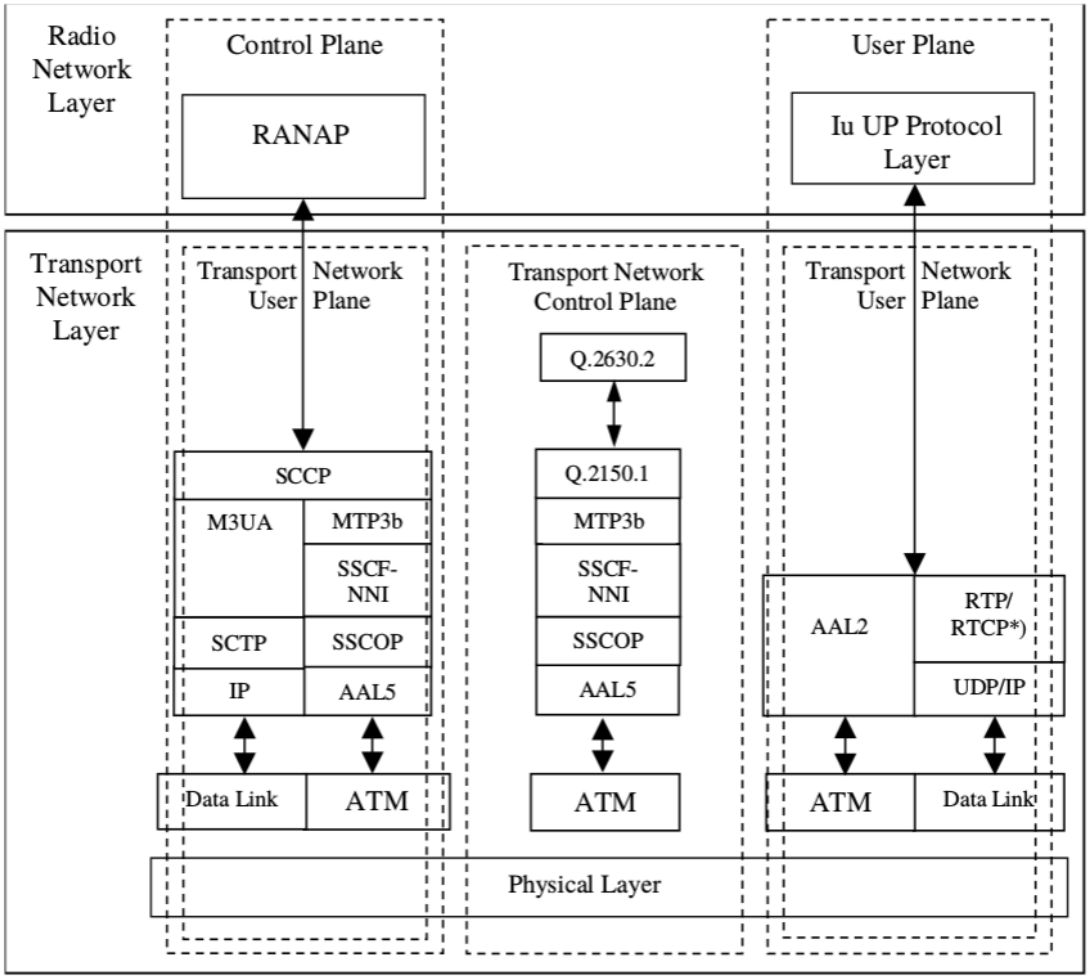
UMTS Protocol stacking

- Iu is split in Iu-CS (MSC) and Iu-PS (SGSN)
- Next slides show protocol stacking of Iu-CS and Iu-PS
- Notice all the ATM legacy that's way obsolete by now
- IP based transport does away with a lot of it
- however, M3UA and SCCP remain even on IP based Iu

UMTS protocol stacking



Iu-CS protocol stacking



*) RTCP is optional.

Figure 6.1: I_u –Interface Protocol Structure towards CS Domain

Iu-PS protocol stacking

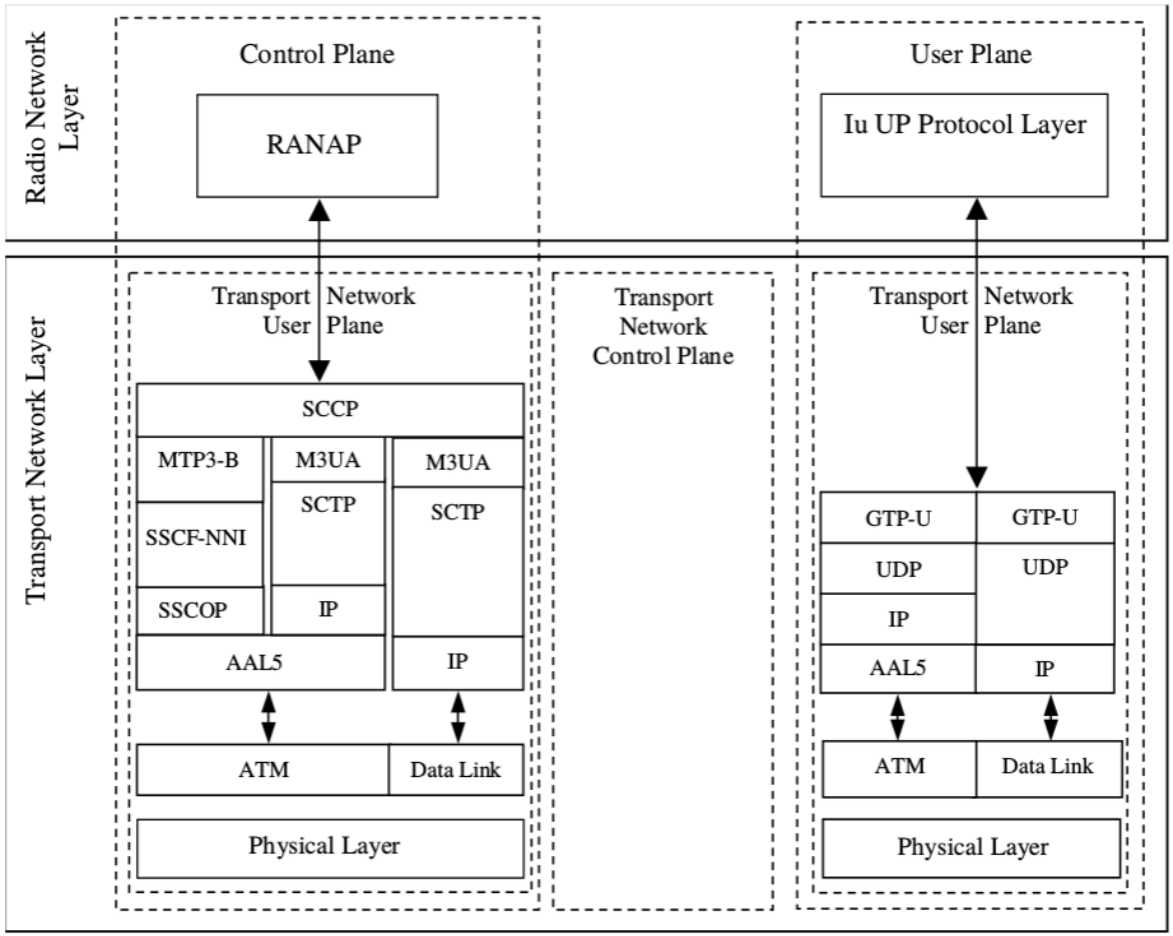
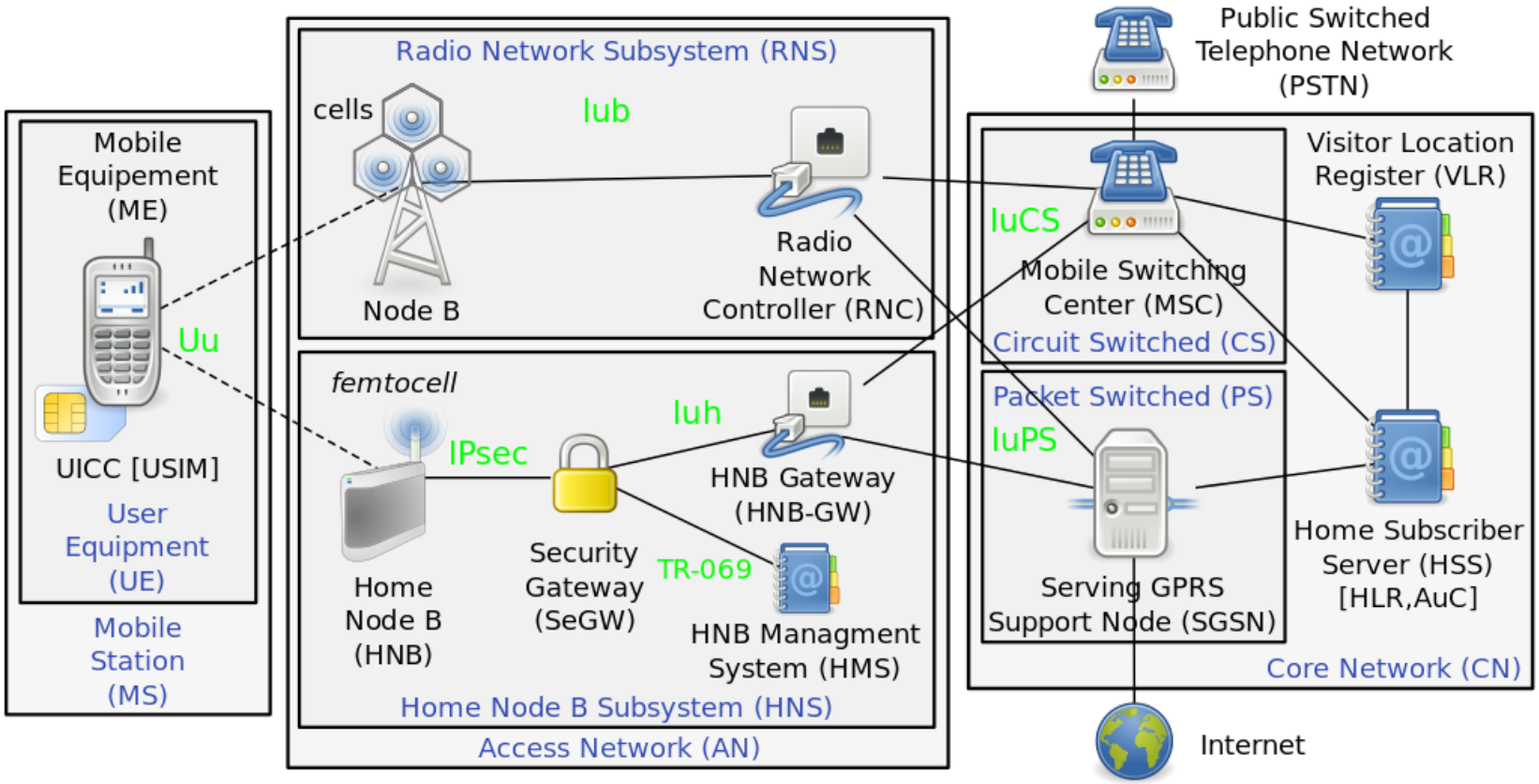


Figure 6.3: I_u Interface Protocol Structure towards PS Domain

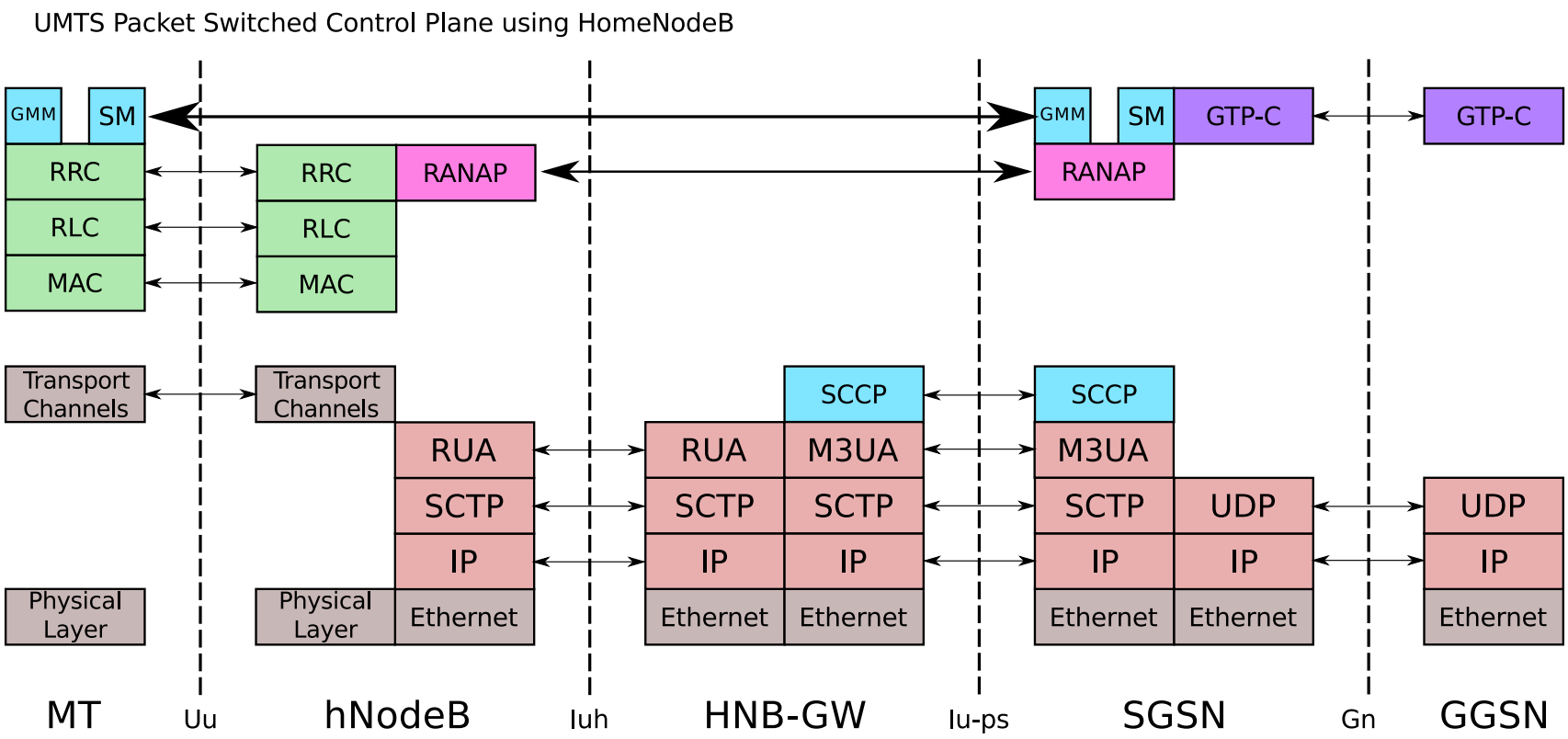
from 3GPP TS 25.410

UMTS Architecture for hNodeB



nodeB and Home nodeB by Tsaitgaist - icons from Gnome

UMTS protocol stacking with HomeNodeB



Differences NodeB to hNodeB

- hNodeB is basically a NodeB with a RNC built-in
- all lower-level protocols are implemented in the RNC
- only RANAP is exposed
- Iuh interface is similar to Iu-CS/Iu-PS
- Iu interface is at much lower level.
- Compared with GSM: Iu = Abis, Iuh = A

Why work with hNodeB instead of NodeB

- UMTS is not a single telephony system but a set of re-configurable building blocks to create any type of telephony system.
- complexity at every level, particularly the lower levels
- using hNodeB interface / stack (Iuh), we can avoid having to worry about RLC/MAC, RRC, HNBAP, etc.
- many femtocells implement Iuh
- quite some small cells also implement Iuh

UMTS channel mapping

speaking of UMTS access stratum complexity...

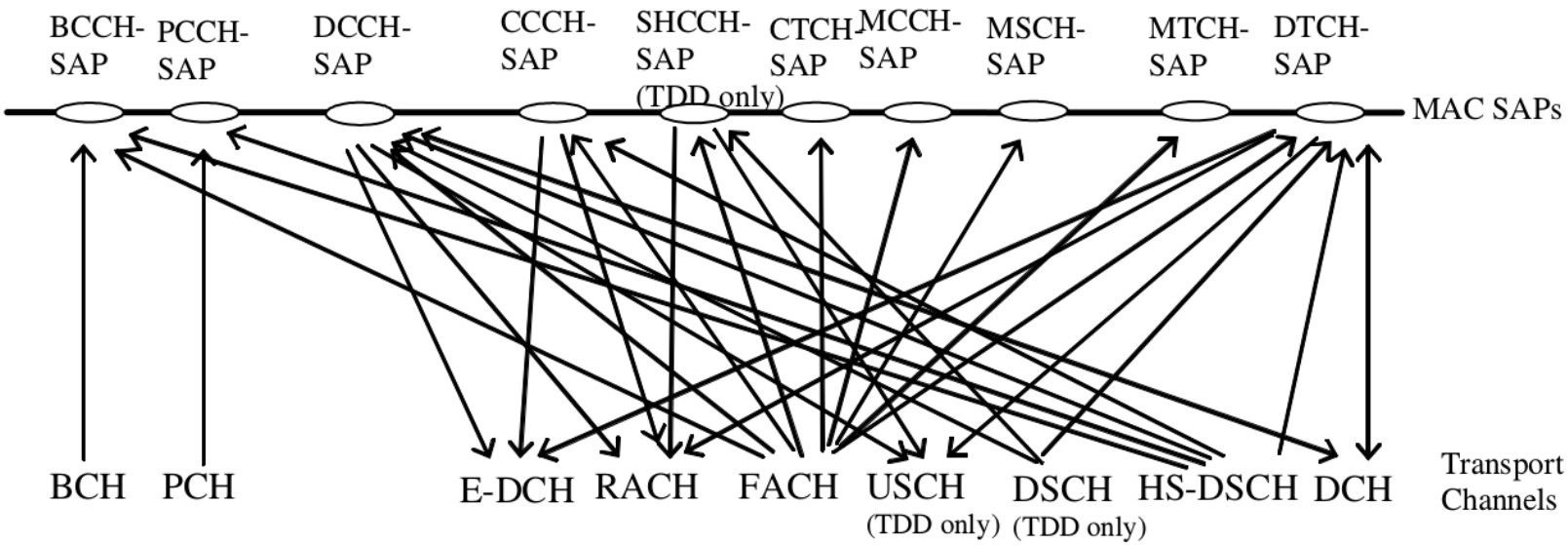


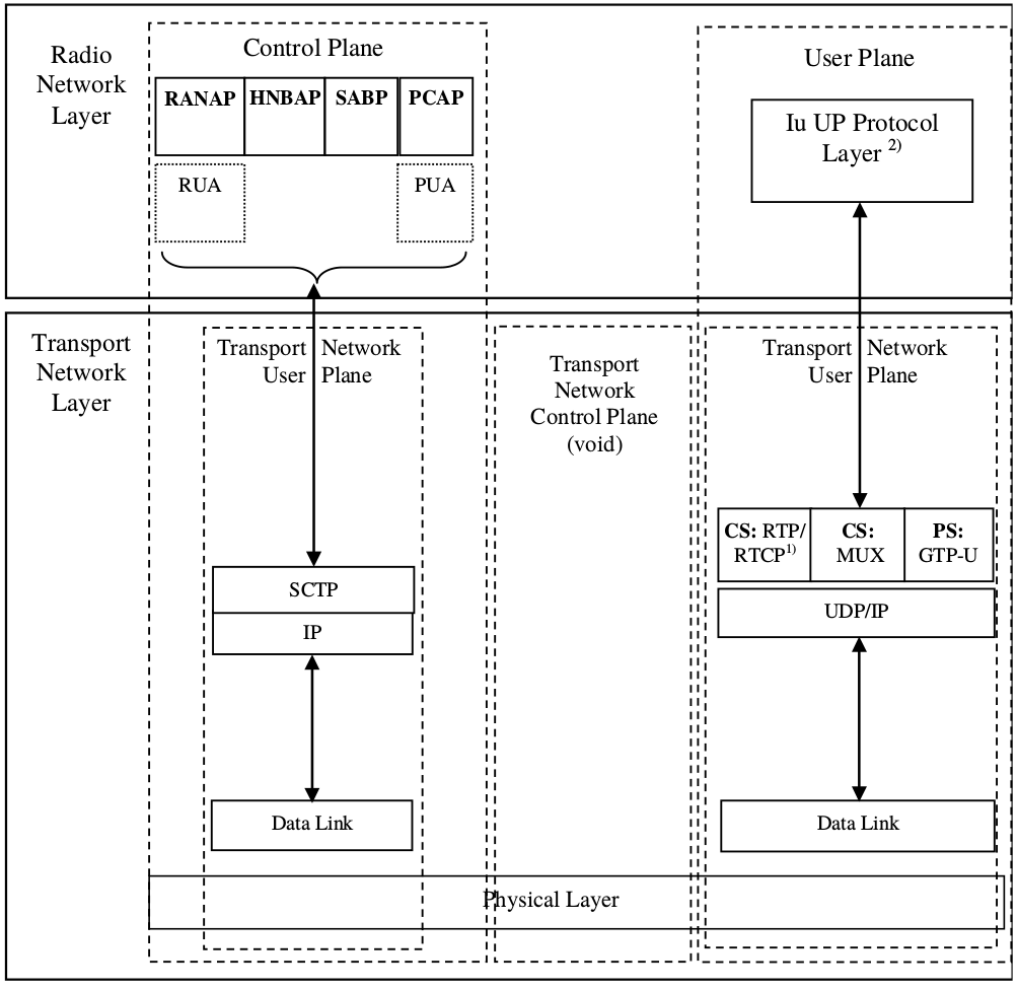
Figure 4: Logical channels mapped onto transport channels, seen from the UE side

from 3GPP TS 25.301

A closer look at Iuh

- Iuh is *basically* just RANAP encapsulated in something less complex over SCTP/IP
- In addition to RANAP, there is
  - RUA (RANAP User Adaption) to replace SCCP
  - HNBAP to register hNodeB and UE
- RANAP for both CS and PS is sent together, but on RUA level there is a *Domain Indicator* that helps separating both.

UMTS protocol stacking for Iuh



Note 1) RTCP is optional.
Note 2) Iu UP is terminated in CN and HNB only (i.e. not in the HNB GW)

Figure 7.2-1: I_{uh}–Interface Protocol Stack.

RUA Protocol (3GPP TS 25.468)

- Very simple connection-oriented layer
  - `CONNECT`
  - `DIRECT TRANSFER`
  - `DISCONNECT`
  - `CONNECTIONLESS TRANSFER`
  - `ERROR INDICATION`
- 24-bit Context ID differentiates multiple parallel RUA connections

HNBAP Protocol (3GPP TS 25.469)

- HNBAP protocol has only very few messages/transactions
  - HNB REGISTER (REQUEST, ACCEPT, REJECT)
  - HNB DE-REGISTER
  - UE REGISTER (REQUEST, ACCEPT, REJECT)
  - UE DE-REGISTER
  - TNL UPDATE (REQUEST, RESPONSE, FAILURE)
  - HNB CONFIG TRANSFER (REQUEST, RESPONSE)
  - ERROR INDICATION
  - CSG MEMBERSHIP UPDATE
  - RELOCATION COMPLETE
- most important is HNB and UE registration

RANAP Protocol (3GPP TS 25.413)

- Lots of transactions, some key transactions here:
  - RESET / RESET ACKNOWLEDGE
  - INITIAL UE MESSAGE
  - DIRECT TRANSFER
  - IU RELEASE (COMMAND, COMPLETE)
  - SECURITY MODE (COMMAND, COMPLETE, REJECT)
  - PAGING
  - RAB ASSIGNMENT (REQUEST, RESPONSE)

SCCP in Free Software

- comes in connection-less and connection-oriented flavor
- is used a lot in SS7 core network protocols
- connection-oriented SCCP is only used on classic GSM A interface (over E1) and in UMTS Iu interface
- no finished free software implementation of connection-oriented SCCP exists
  - libosmo-sccp, Yate, Mobicents only implement connection-less
  - osmo_sccp Erlang code has partial but never completed/tested code for connection-oriented mode

How to support UMTS from OsmoNITB, OsmoSGSN

- Separation of MSC-part from NITB, generating Osmo-MSS
  - OsmoBSC already implements BSC-side A interface, we need to add MSC-side A interface
- UMTS AKA support as library, link into OsmoMSS and OsmoSGSN
- RANAP protocol support in a library, also linked into OsmoMSS and OsmoSGSN
- NITB: support `subscriber_connection` over A (BSSMAP/BSSAP) and over RANAP
- SGSN: support `mm_context` over Gb (LLC/BSSGP/NS) or over RANAP

How to encapsulate RANAP towards the RAN

- we could either
  - Try to convert from Iuh to A interface, make (h)NodeB look like GSM BTS+BSC.
  - Implement classic Iu-CS and Iu-PS over SCCP/M3Ua and have a classic HNB-GW to convert to Iuh
  - Implement Iuh directly, avoiding SCCP and M3UA
- Iu-CS/PS requires connection-oriented SCCP
- when implementing Iuh directly, we still need to somehow split CS and PS plane
- Idea: Simple proxy that speaks Iuh to hNodeB, MSS and SGSN
- Iu-CS/PS over SCCP/M3UA could be added later, if required

RANAP, RUA and HNBAP Encoding

- Use ASN.1 syntax for defining protocol messages
- Use APER (Aligned Packed Encoding Rules)
  - unlike BER: No Tag/Length values
  - unlike UPER: all fields start at octet boundary
- ASN.1 syntax uses Information Object Classes heavily
- ASN.1 is not abstract enough for them, so they use ASN.1 to define containers, i.e. they build something like a TLV structure inside ASN.1
  - Every IE is its own ASN.1 SEQUENCE, and it gets wrapped into an IE container indicating an IEI and the encoded sequence
  - The Main message then simply has an array (SEQUENCE OF) of IE containers
- Regular ASN.1 code generator will not generate very useful code for this, i.e. it will not be able to parse the entire message in one go, but it requires manual iteration code that calls the generated decoder separately for every IE Container

RANAP, RUA, HNBAP and asn1c

- Lev Walkins asn1c is a Free Software ASN.1 compiler / code generator
- it is good for basic usage, but lacks many if not most of the features required in telecom
  - No support for information object classes
  - No support for aligned PER support
  - No support for type prefixing, i.e. every type uses the same global C namespace and you have problems if RANAP, RUA and/or HNBAP all have types of the same name
- No other free software alternatives exist
- Somebody with firm knowledge on compiler theory needs to help out, I'm at a loss here.

Alternatives to asn1c

- Write all related code in Erlang
  - I tried that in the past, but nobody ever contributed to any of the Osmocom Erlang projects :(
  - At Osmocom we're mostly low-level C guys with an inherent dislike of abstract/complex languages, VMs and the like
- Use proprietary asn1 compiler
  - In theory not a problem, as the compiler has no copyright on the generated C code, we can use it from FOSS
  - Problem: Mandatory runtime code is proprietary
  - We certainly don't want proprietary blobs in Free Software, ever
  - FOSS code would have to be MIT/BSD/LGPL, incompatible with osmo-* GPL/AGPL.
- So it seems we have to stick with asn1c, after all

How to make asn1c work for Iuh

- Eurecom has a patch for adding APER support to asn1c
  - it's against an ages old version of asn1c
  - I forward-ported that to current asn1c master
  - Probably needs some clean-up before it can be merged
- Information Object Classes are hard
  - compile only the IE and PDU definitions of the ASN.1
  - skip all parts related to Information Object Classes
- Type prefixing
  - Could be done in the ASN.1 source files, but that's ugly
  - I hacked asn1c for a day until I finally had found all the locations where prefixing must be used (or not)
  - Code is at `git://git.osmocom.org/asn1c.git`

But what about the IE Containers?

- Eurecom has an `asn1tostruct.py` script
  - Another layer on top of asn1c to handle the IE containers and un-do the damage caused by the additional layer of abstraction of RANAP and related protocols
  - Developed to cope with S1-AP (RANAP equivalent for LTE)
  - Can be used for Iuh with some modifications
  - Also had to be taught type prefixing

Putting it all together

Brief history of what I did so far:

- copy+paste Asn.1 syntax from 3GPP .doc files
- use hacked asn1c to generate C code
- don't use copied runtime code but shared osmocom libasn1c
- use modified asn1tostruct.py for the obfuscation layer
- write some code to dispatch messages
- implement minimally required transactions like `HNB REGISTER`, `UE REGISTER`
- see the `INITIAL UE MESSAGE` with the `LOCATION UPDATE`

```
git clone git://git.osmocom.org/osmo-iuh.git
```

Where do we go from here?

- Implement UMTS AKA in libosmogsm, test over GSM and GPRS
- Crete small HNB-GW with RANAP-over-RUA on both sides, splitting CS and PS
- Split OsmoMSS from OsmoNITB, add RANAP interface
- Add RANAP-over-RUA to OsmoSGSN
- More Volunteers needed!

What kind of hardware can we use?

- The (undisclosed) small cell hardware I currently use is very expensive (several thousand EUR) and thus not suitable to most hackers
- Many consumer-grade femtocells in the market, most modern ones should use Iuh
  - they are typically quite locked down and provide no local console / JTAG
  - they establish an IPsec tunnel to the SEGW (Security Gateway) and then only talk Iuh inside the tunnel
  - Several groups of people have looked at them in the past (including Kevin, Nico and myself)
  - maybe we can find a model that's easily convinced to talk to a different HNB-GW?

Summary

- Iuh is actually not difficult conceptually
- Lack of good FOSS asn1 tools is biggest factor
- Obfuscation by IE Containers must be overcome
- In the end you spend 90% of the time on tooling, before you can spend the remaining 10% on actual code
- Core Iuh protocol code exists now as `osmo-iuh`
- Work on OsmoMSS and OsmoSGSN has not even started yet
- Volunteers needed. Now!

Thanks

Thanks for your attention. I hope we have time for Q&A.