

VERIFICATION OF THE CORE SOFTWARE OF MOBILE NETWORKS

Gábor Soós

soos@tmit.bme.hu

BME-TMIT

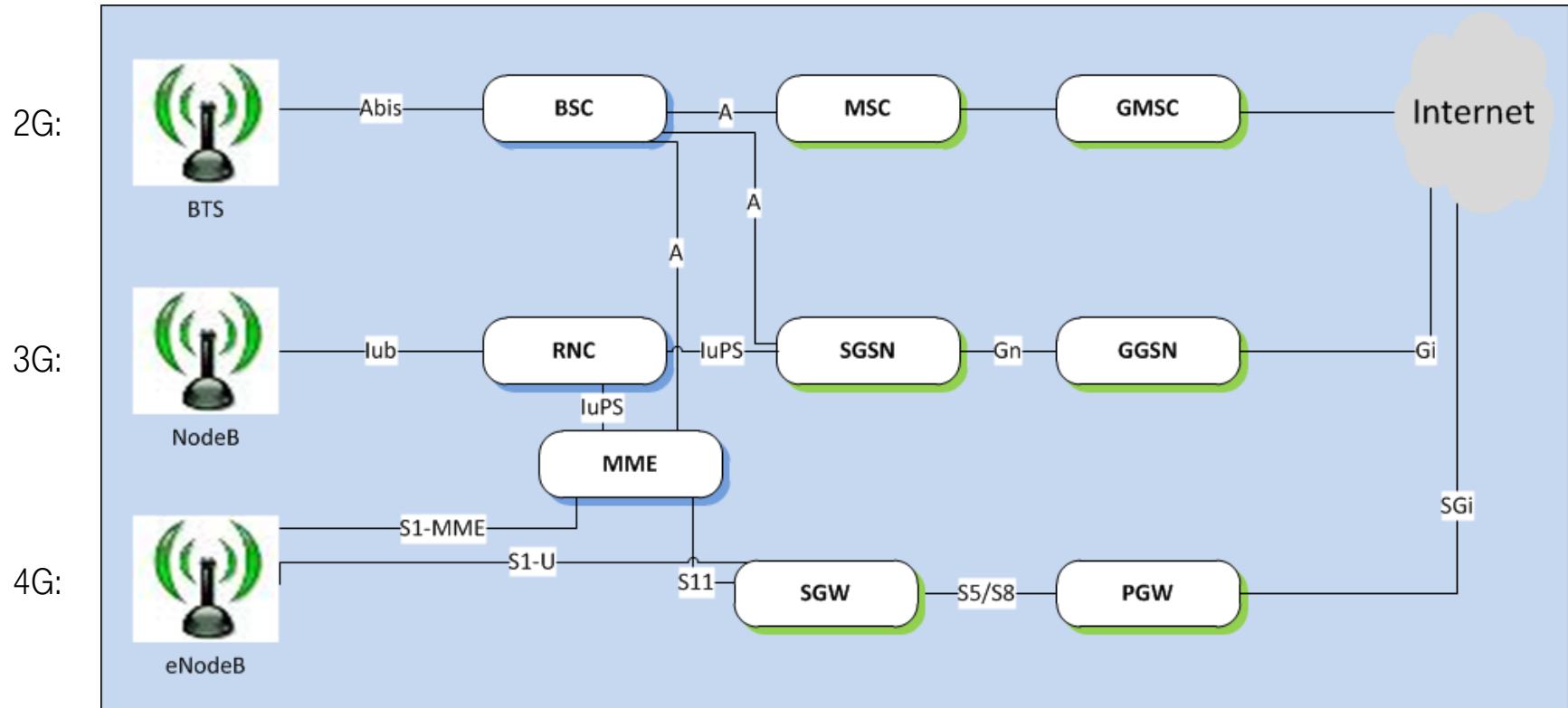
Budapest University of Technology and Economics

Software Verification and Validation

7. Dec 2017

- Overview of cellular network
- Motivation
- Definitions
- Current development cycle
- Verification methods and recommendations
- Results

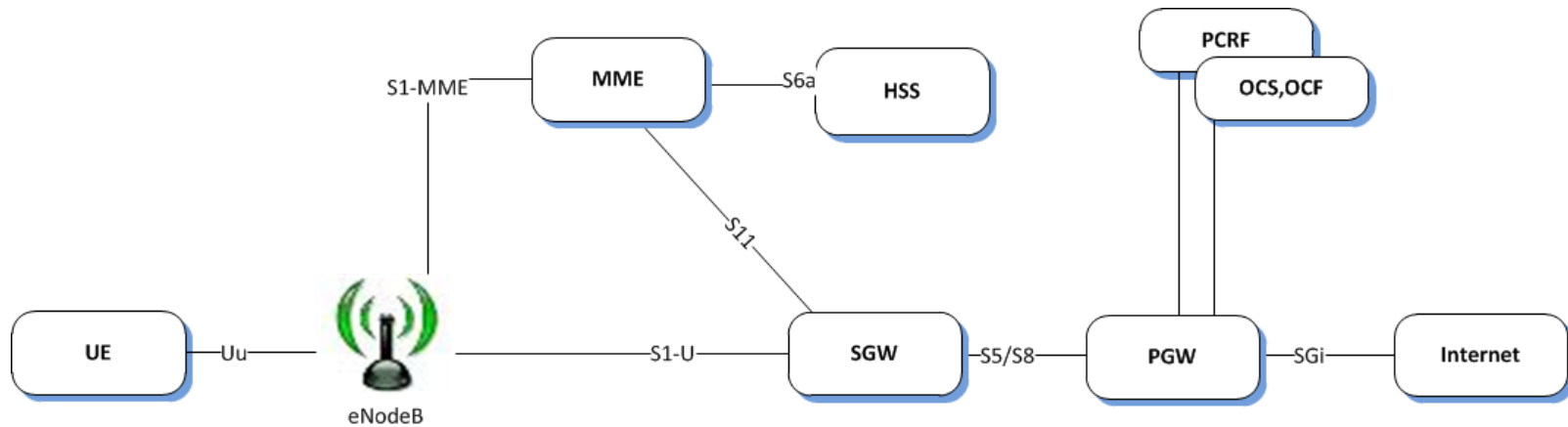
THE MOBILE NETWORK- OVERVIEW



Radio related services

IP related services

SOFTWARE FUNCTIONS IN A 4G MOBILE NETWORK



MME

Controlling the user
User Authentication
Encryption
Device Authentication

SGW

Handling user data
Handover between:
eNB
Access technology

PGW

Ending user data to IP network
Check: user traffic
available balance
QoS

MOTIVATION

Customers:

changing cellular phones, devices, software
changing security requirements, type of behavior
changing applications, way of usage

Services:

2G, 3G, 4G, cellular technology
+WiFi, Bluetooth, Zigbee

New features:

Throughput (speed)

Latency

New standards:

applications, low power, high latency, better coverage

Competitors:

Customers experience is a key role here

**All SOFTWARE based
functions!**

DEVELOPMENT CYCLE

Customer(external/internal)

- feature to add
- failure to correct, parameter to be improve

Designer

- Get current architecture
- Specification and current technology for change(HW/SW)
- HighLevelDesign, LowLevel Design
- Testing specification for current task and overall architecture

Implementation:

- Design: functions, features, parameter alignment
- Developer: integration and implementation

Testing:

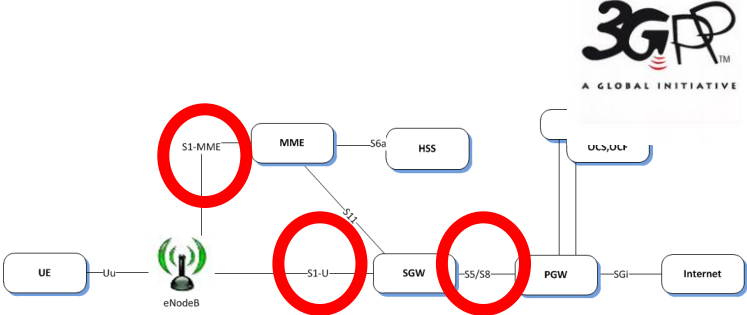
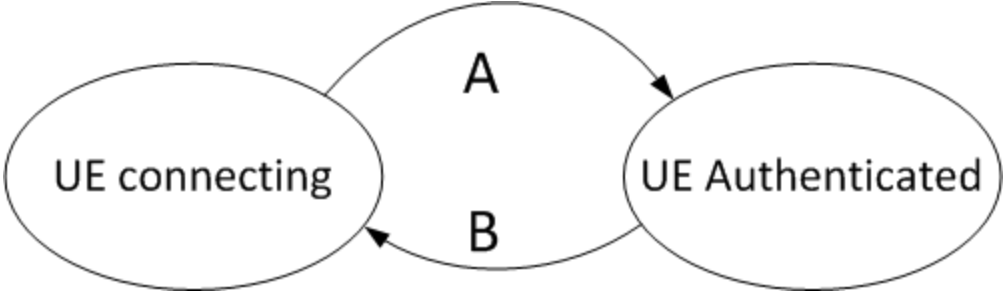
- Regression
- Integration
- Input
- Stability
- Long time testing (1week, 1month)

Fall back

- In case of service degradation, fall-back process shall be executed

All manual phases and tasks!

COMPLICATE CONNECTION FLOWS- MOBILE NETWORK



UE connecting → UE Authenticated: A

MME L7: MME service shall be running
 GTP port shall be open
 eNB shall be attached
 HSS service shall be started
 HSS port shall be open

L3: firewall, routing shall be established
 Packet rate shall match the expectation
 buffers shall be ok

Firewall shall allow app. Packet rate
 Settings: APN, IP-type, user/passwd

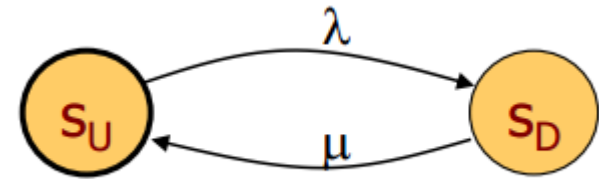
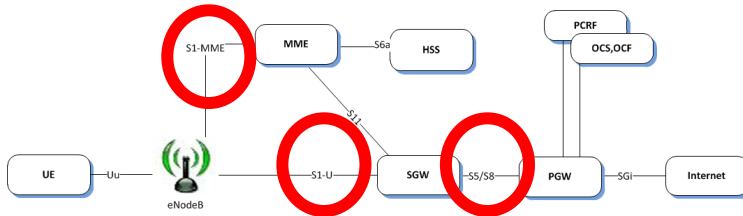
UE connecting → UE Authenticated: A

User L7: Connecting button shall be pressed
 correct APN setting, IP-type
 correct Username/password
 MCC id must be connect

User L1: stable battery
 stable internal software
 stable User!

How to test it automatically, regularly, and detect failures?

THE NEW WAY OF VERIFICATION



Instead of defining and testing very properly each states and values one by one, describe the possibility of transition with a probability distribution. Based on this the different software and states, they can be compared well with previous states.

Advantages:

- Testing the current software version very quickly

- Get an overall benchmark

- Benchmark about the possibilities of the current network set

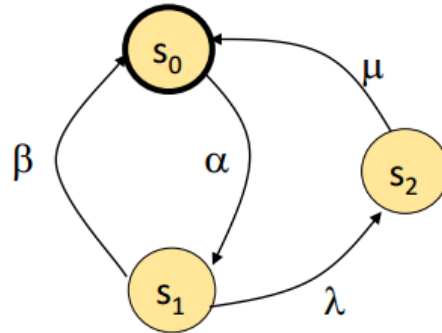
- Great help for the operators to show them where to look for errors on a dashboard

Disadvantages:

- Can hide „not high impact” failures

Continuous Time Markov Chain model

- MOBILE NETWORK



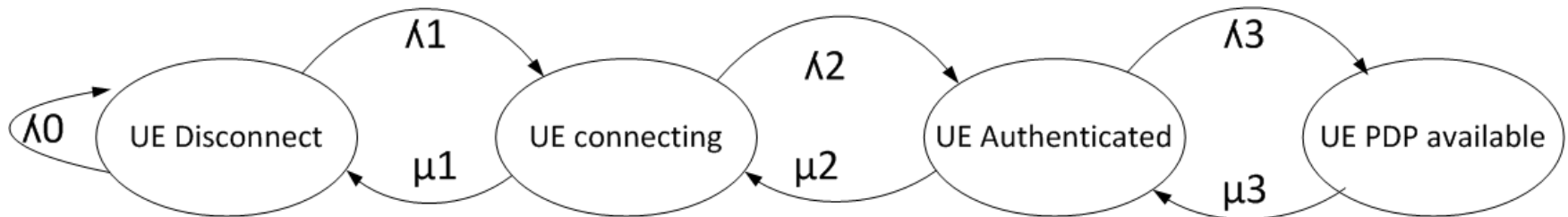
Continuous Time Markov Chain :

Definition: CTMC = (S, R)

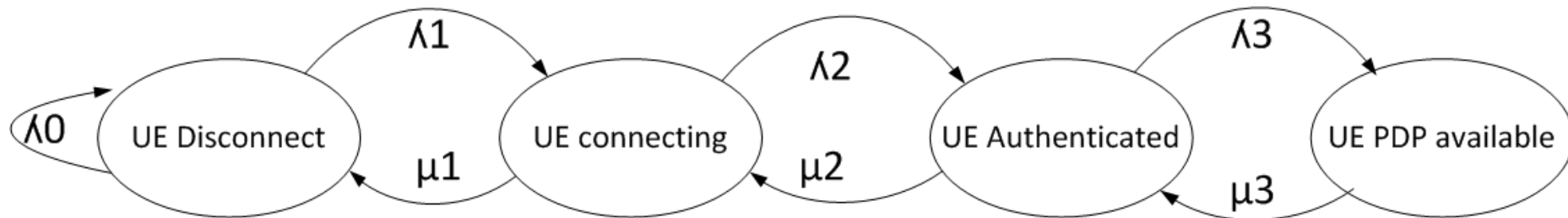
S: set of discrete states: s_0, s_1, \dots, s_n

R: state transition rates

Details can be improved, if required. Single model with transition rates:



MME SOFTWARE VERIFICATION



Transition rates:

UE begin connecting after UE Disconnect state:

UE PDP will be available after UE Authenticated:

$$\lambda_3 = 1 - \mu_2$$

| SW version | λ_1 Average | λ_1 dispersion | μ_1 Average | λ_2 Average | λ_2 dispersion | μ_2 Average |
|------------|---------------------|------------------------|-----------------|---------------------|------------------------|-----------------|
| 1.0 | 0,8 | 0,1 | 0,2 | 0,6 | 0,2 | 0,4 |
| 2.0 | 0,7 | 0,1 | 0,3 | 0,5 | 0,2 | 0,5 |

TESTING THE SOFTWARE

Automaton based

Interface and functional test:

- Services and ports
- Links reliability

Simulated single user Behavior:

- Attach
- Connecting
- Disconnecting
- Roaming
- User traffic

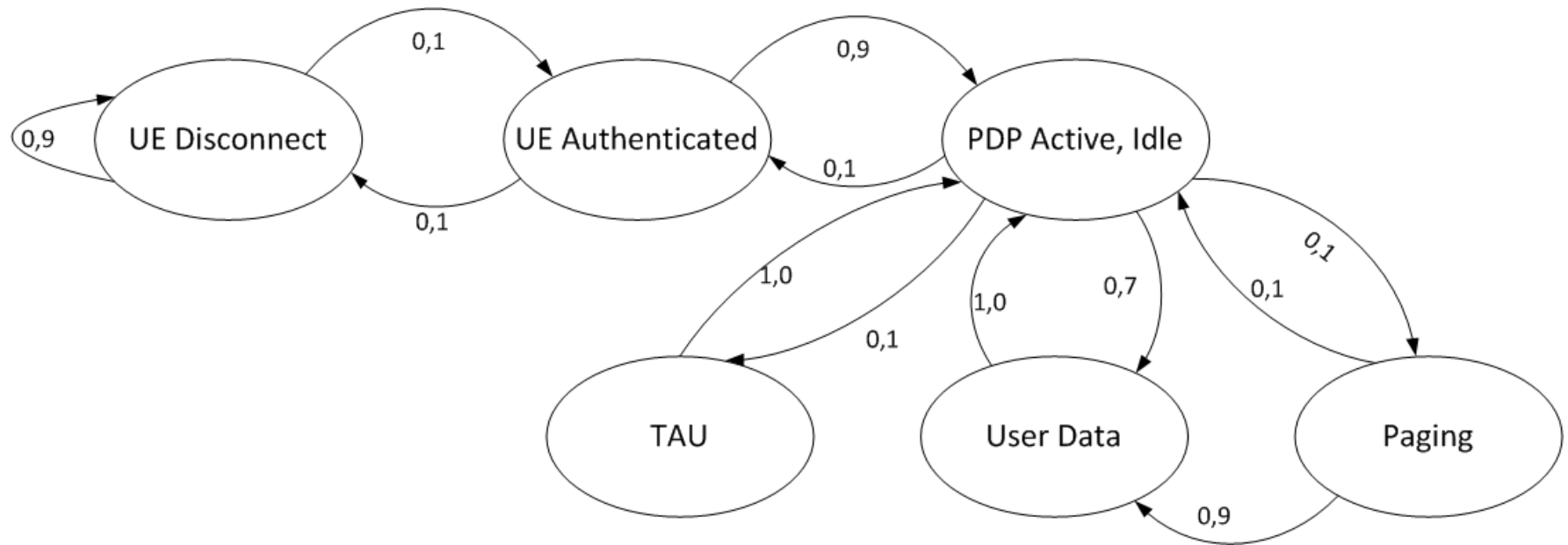
Simulated multi user:

- Sending 1000X same attach from different users
- Roaming between eNBs
- Max throughput tests

Still manual

- Earlier detected failures
- New services and functions
- New expected rates
- New applications

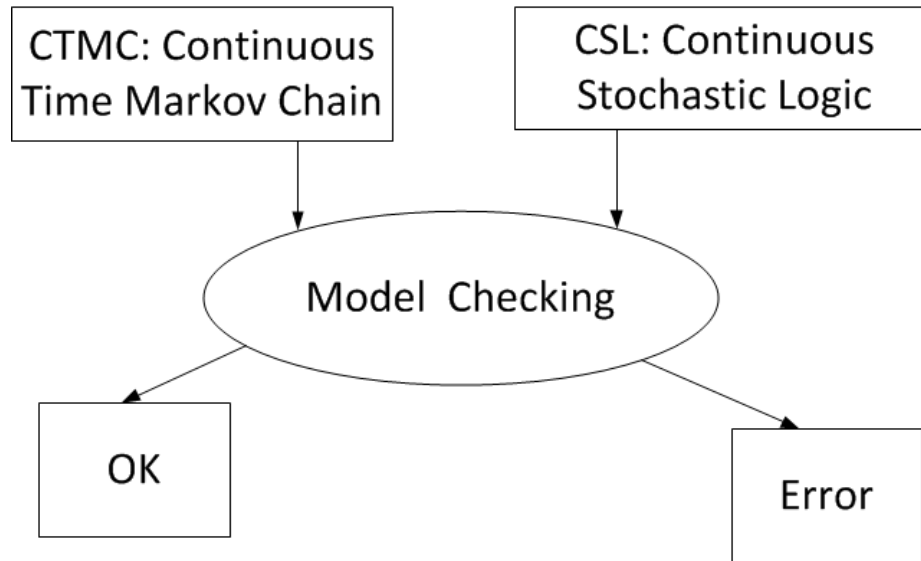
PROBABILITIES- USER ATTACH SCENARIO (X VENDOR R14. VIRTUAL SOFTWARE)



RESULTS – FINGERPRINT OF THE CELLULAR NETWORK SET (HW & SW BONDED)

| SW: R14.5, HW: x86, c46, CPU: type A, MEM: type: B, Architect: v4.1, Feature: R13, with: 1 user | | | | |
|---|------------------|------------|--------------------------|-----------------------------|
| | Avarage [probab] | dispersion | answer time average [ms] | answer time dispertion [ms] |
| Λ1 | 0,9 | 0,001 | 350 | 45,0354158 |
| Λ2 | 0,8 | 0,01 | 200 | 27,01710532 |
| Λ3 | 0,9 | 0,05 | 182 | 28,12867827 |
| Λ4 | 0,9 | 0 | 120 | 28,66531438 |
| Λ5 | 0,8 | 0,2 | 175 | 38,18460792 |
| Λ6 | 0,9 | 0,05 | 650 | 24,58812048 |
| Λ7 | 0,7 | 0,06 | 890 | 18,22165846 |
| Λ8 | 0,6 | 0,3 | 400 | 28,80892808 |
| Λ9 | 0,9 | 0,1 | 810 | 32,18747156 |
| Λ10 | 1 | 0 | 2500 | 8790,273155 |

AUTOMATED MODEL CHECKING



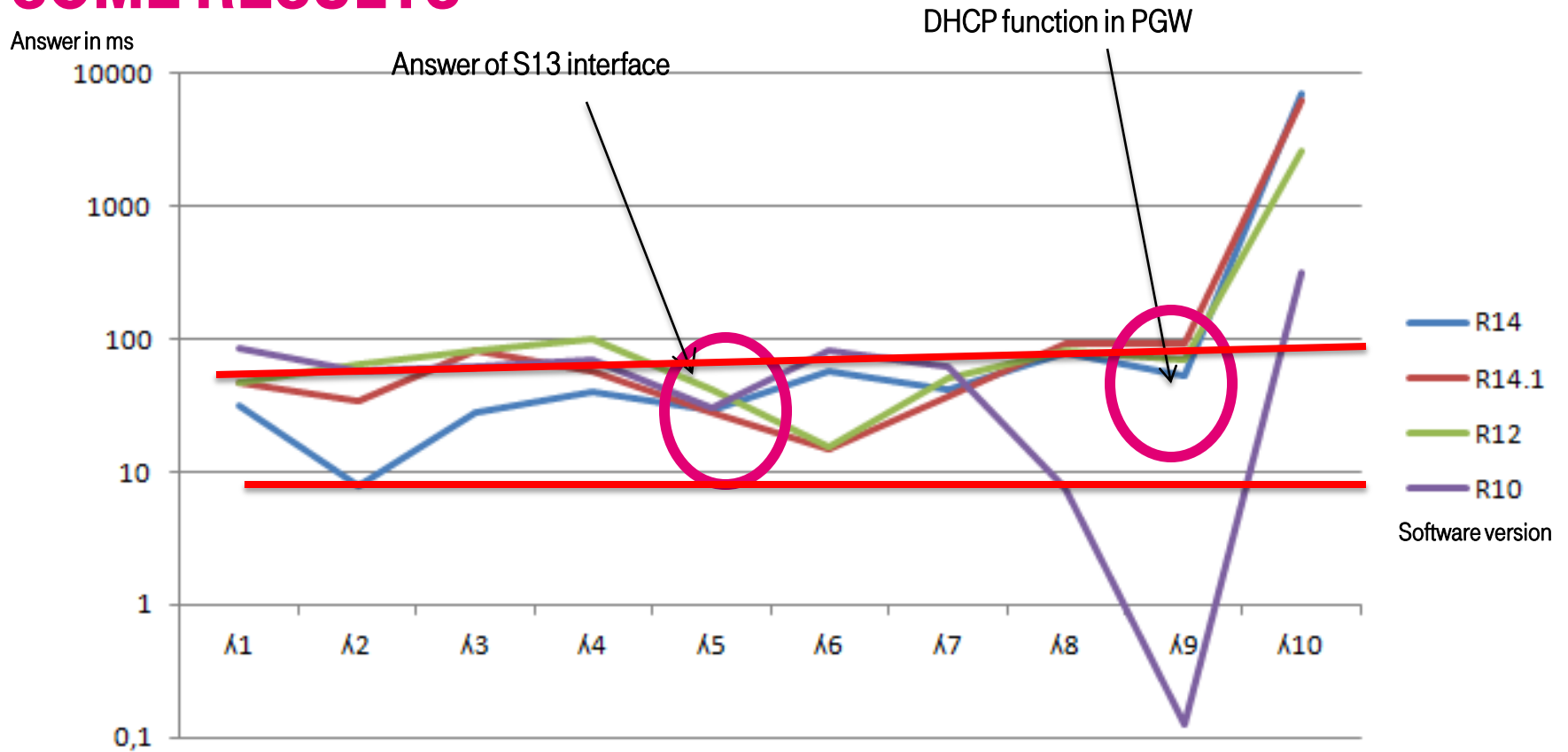
Parameters shall be in „some” close range of the expectation. If they are not, then an Error mark shall be showed.

Verification means monitoring, checking and giving feedback:

Time-slots, communications, synchronization.

Memory and CPU-usage

SOME RESULTS



Automaton, interpreting data from live network and comparing with expected results.

Divergence from the stripe is not acceptable.

THANK YOU!