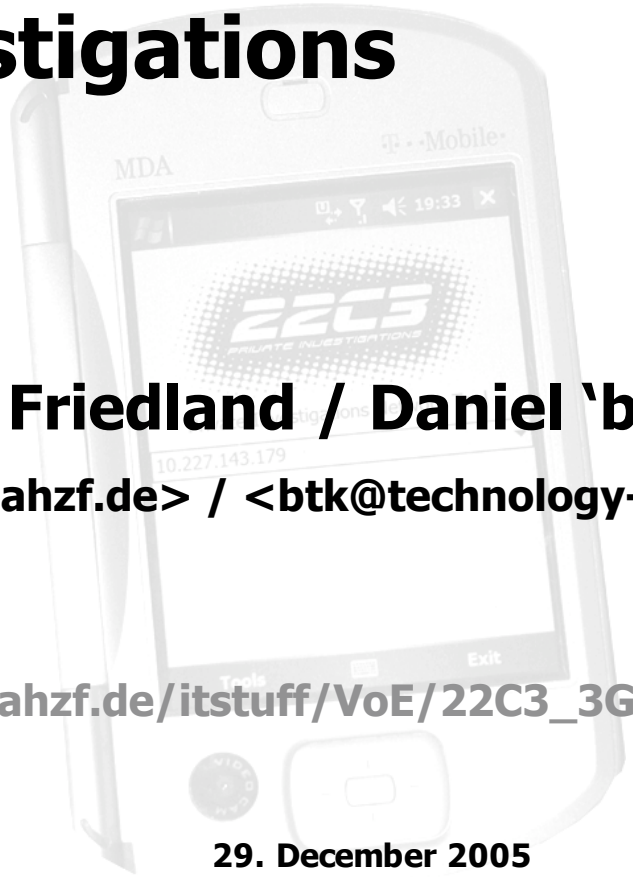
3G Investigations



Achim 'ahzf' Friedland / Daniel 'btk' Kirstenpfad

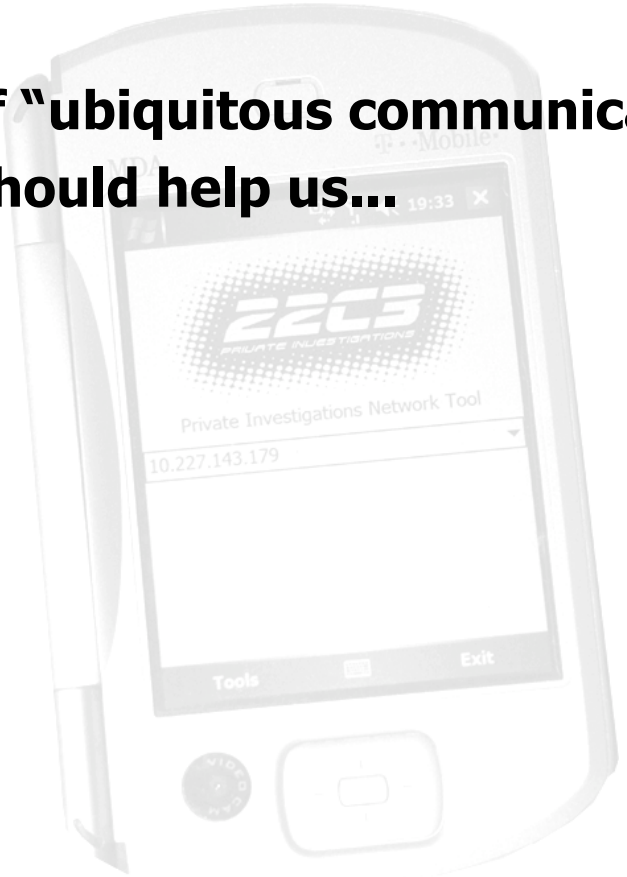
<22C3@ahzf.de> / <btk@technology-ninja.com>

http://www.ahzf.de/itstuff/VoE/22C3_3GInvestigations.pdf

29. December 2005

Sometime in the past...

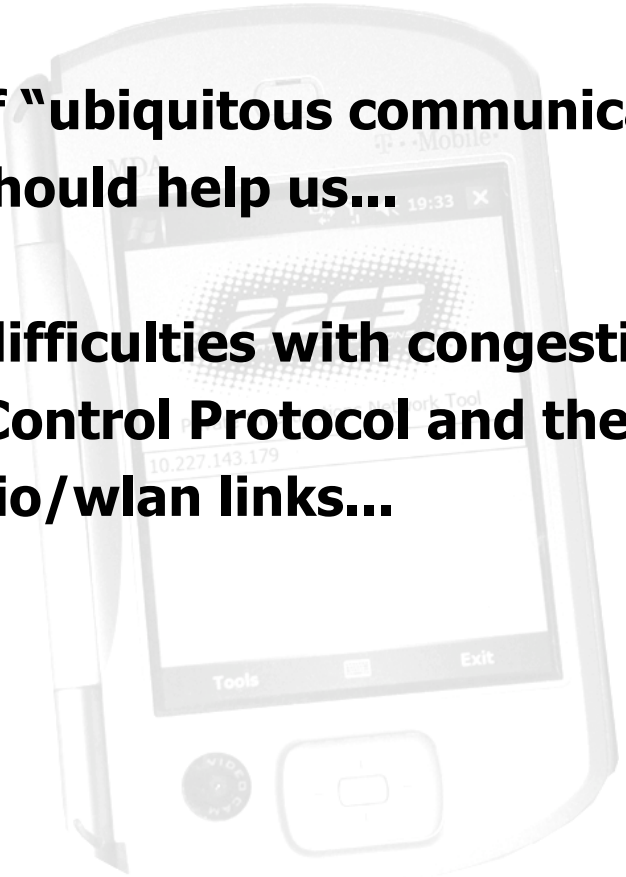
... we dreamed of "ubiquitous communication" and radio technologies should help us...



Sometime in the past...

... we dreamed of "ubiquitous communication" and radio technologies should help us...

but we had some difficulties with congestion control of the Transmission Control Protocol and the bursty nature of failures on radio/wlan links...



Sometime in the past...

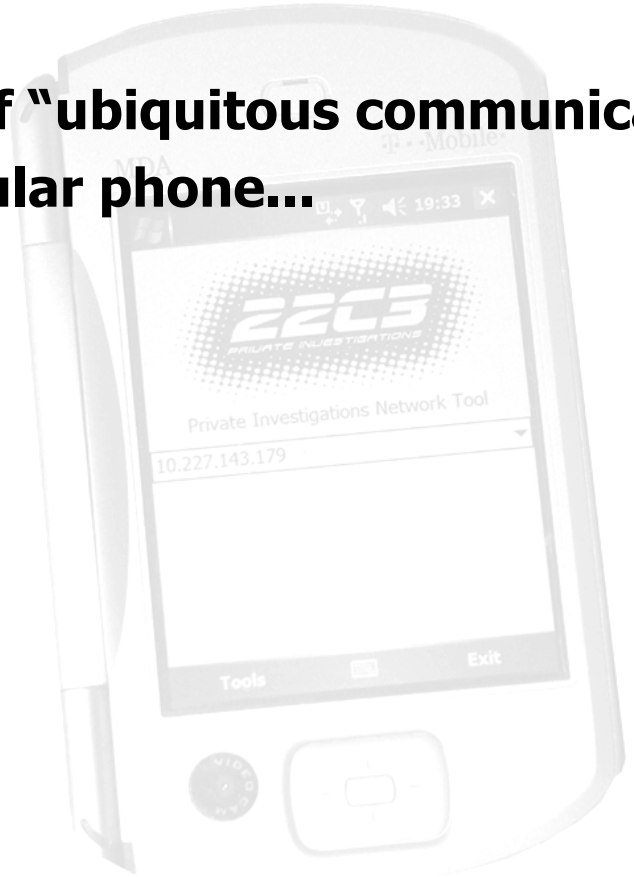
... we dreamed of "ubiquitous communication" and radio technologies should help us...

but we had some difficulties with congestion control of the Transmission Control Protocol and the bursty nature of failures on radio/wlan links...

After some thinking people of earth found a solution: TCP with Selective Acknowledgments (TCP SACK, rfc 2018)

Some billion euros later...

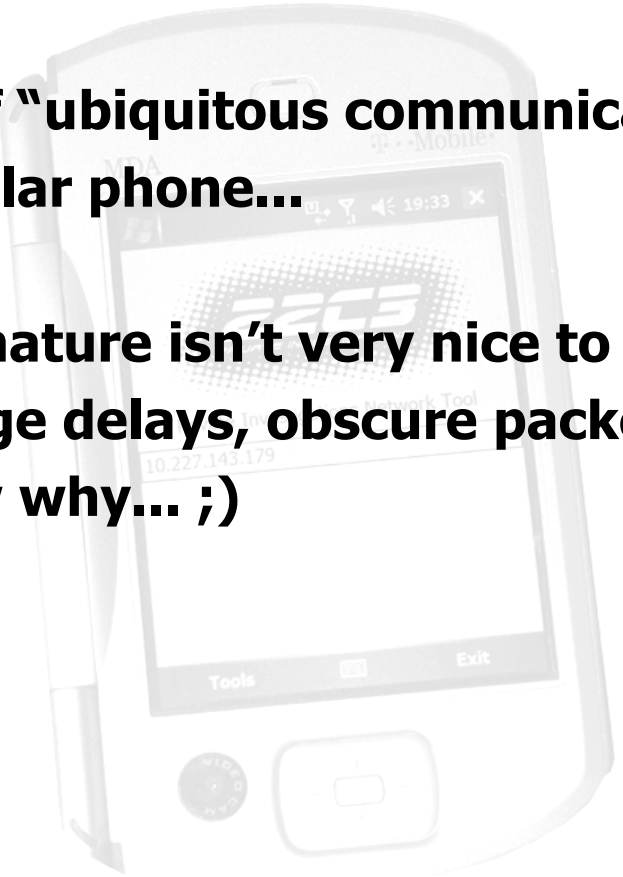
... we dreamed of “ubiquitous communication” with our new 3G/UMTS cellular phone...



Some billion euros later...

... we dreamed of "ubiquitous communication" with our new 3G/UMTS cellular phone...

but again mother nature isn't very nice to us. We have to suffer of strange delays, obscure packet losses and nobody seems to know why... ;)



Some billion euros later...

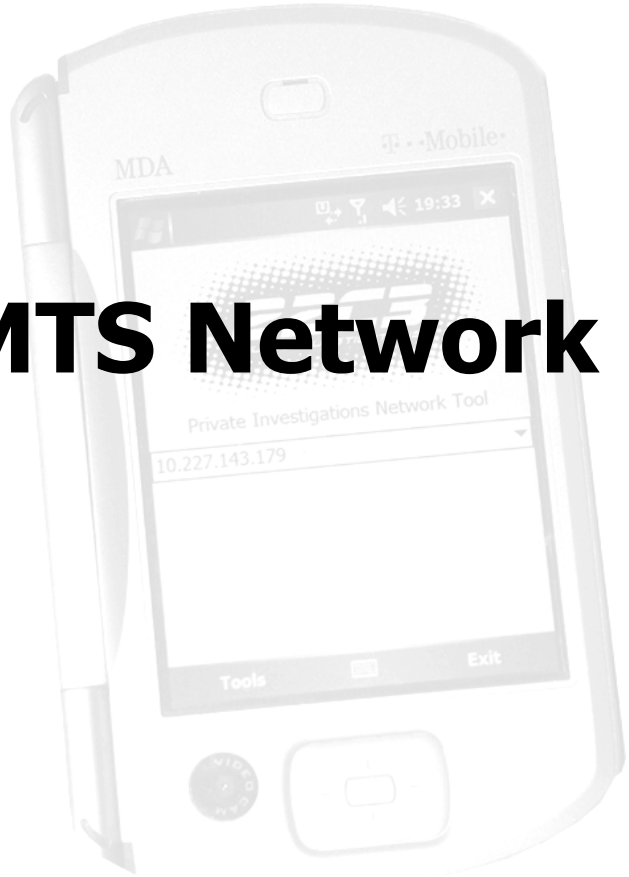
... we dreamed of "ubiquitous communication" with our new 3G/UMTS cellular phone...

but again mother nature isn't very nice to us. We have to suffer of strange delays, obscure packet losses and nobody seems to know why... ;)

We want to investigate this phenomena a bit closer...

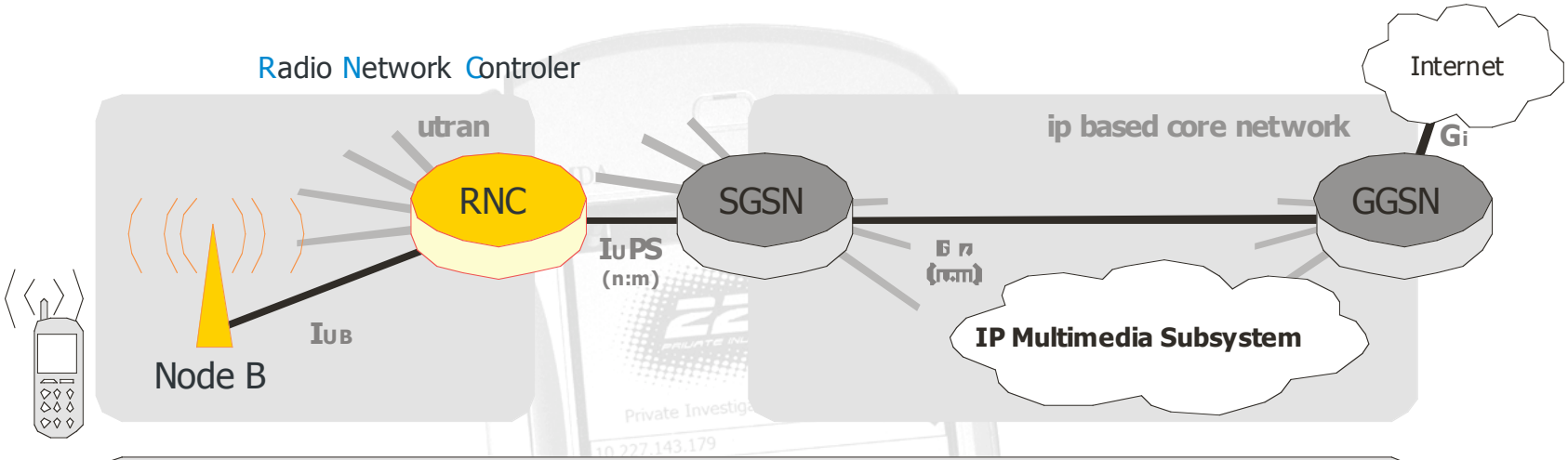
Contents

<22c3@ahzf.de, btk@technology-ninja.com>



1. UMTS Network Details

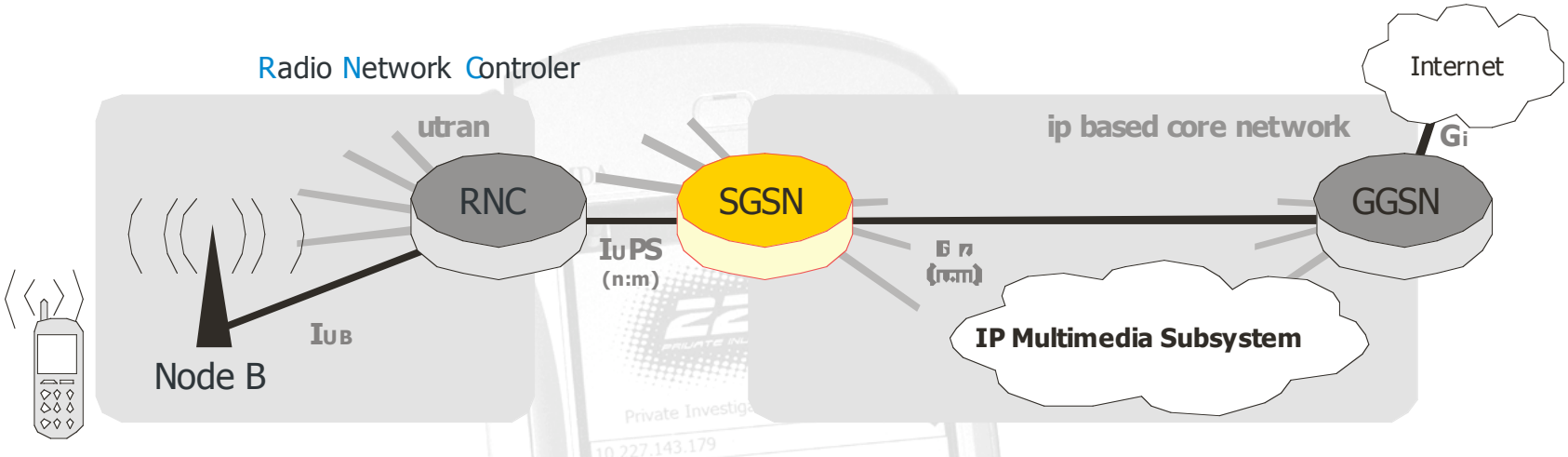
1.1 UMTS network topology - UTRAN TS 23.002 Network Architecture



UMTS - UMTS Terrestrial Radio Access Network

- More or less unimportant for us, but...
- UMTS: ACKs on packets are generated in RNC
- HSDPA: ACKs on packets are generated in Node B
- UMTS (~80ms) vs. HSDPA (~2ms)

1.1 UMTS network topology - SGSN

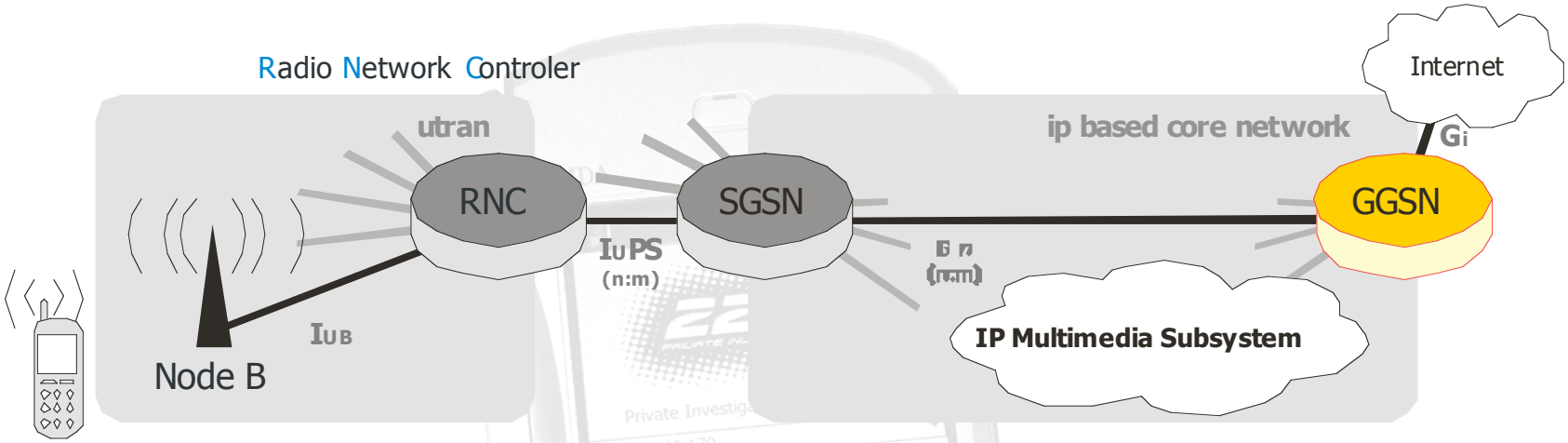


Serving GPRS Support Node

- **Session setup/management**
- **Mobility management**
- **Subscriber database management (->HLR)**
- **Charging data for radio network usage**

1.1 UMTS network topology - GGSN

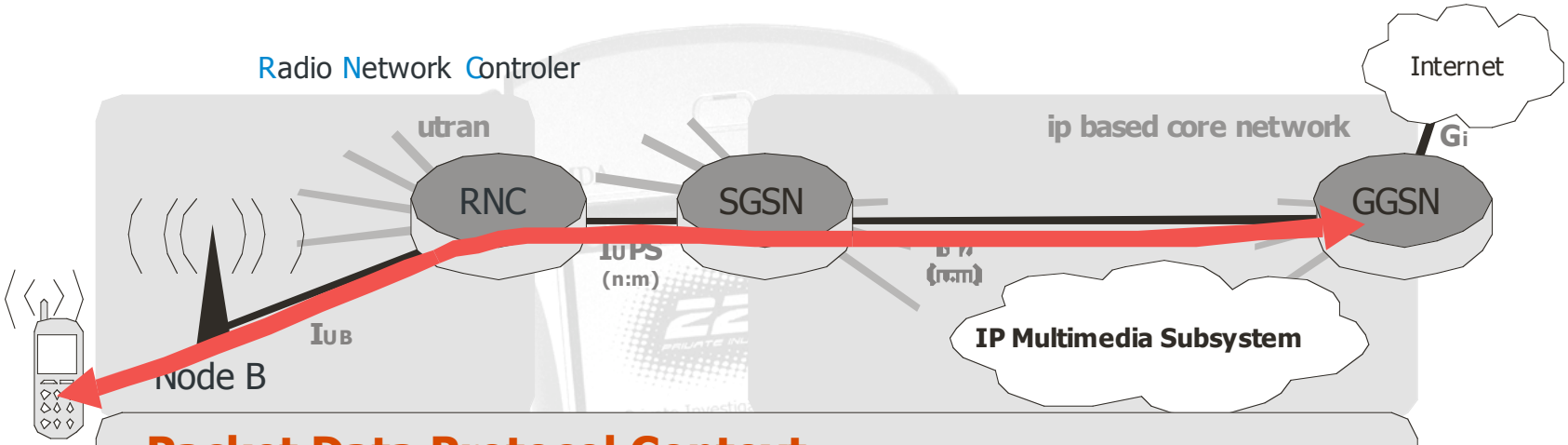
TS 23.060 General Packet Radio Service



Gateway GPRS Support Node

- Gateway for UMTS packet service to ext. networks
- PDP/IP Address Configuration, NAT, FA for MIP
- Performs user data screening and security
- Charging data for external network usage

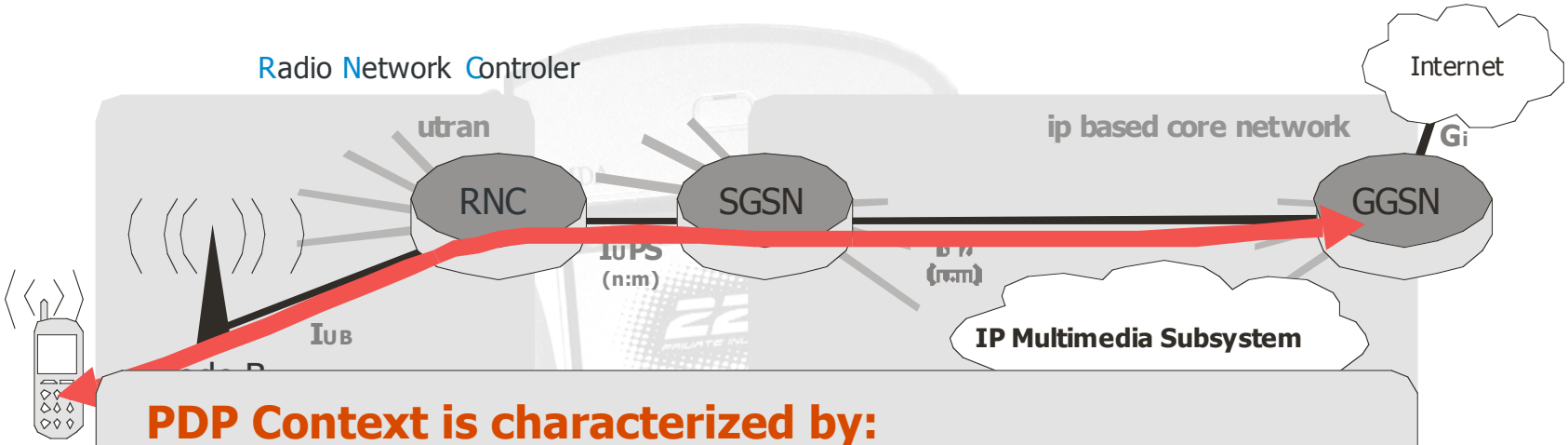
1.2 PDP context for mobility and QoS (packet switched domain, user plane)



Packet Data Protocol Context

- PDP context is a QoS- and mobility-aware tunnel
- Between mobile device and GGSN
- More than one PDP context can be used, but...
- Mobile device is not allowed to initiate a context modification
- Above PDP PPP is used for another session context

1.2 PDP context for mobility and QoS (packet switched domain, user plane)



PDP Context is characterized by:

- **Network Address of mobile device**
  - e.g. IPv4/6 address or a ::/64 IPv6 Prefix
  - CISCO supports a "network-behind-mobile" feature
- **Access Point Name of a terminating GGSN**
  - e.g. web.vodafone.de, internet.t-d1.de
- **QoS-Level**

UMTS Quality-of-Service Classes

- **Conversational Class**
  - e.g. voice, video conference
  - guaranteed bit rate and delay (80ms++), sender statistics (e.g. speech)
- **Streaming Class**
  - e.g. unidirectional video streaming
  - guaranteed bit rate and delay (250ms++) , sender statistics (e.g. speech)
- **Interactive Class**
  - e.g. www, internet games, ssh, news
  - no guaranties but lower bit-error-rate than classes 1&2, no statistics
- **Background Class**
  - e.g. background-services like FTP, e-mail
  - no guaranties but lower bit-error-rate than classes 1&2, no statistics

1.3 Quality of service within UMTS (R6, 3GPP TS 23.107 V6.1.0, 2004-03)

Uplink userdata will be...

- classified according to the PDP context
- conditionalized, e.g. dropped, delayed, ...
- GGSN can translate 'PDP context QoS' to DiffServ
- DiffServe-Tags set by the UE are ignored

Downlink userdata will be...

- reclassified according to the PDP context
- conditionalized, e.g. dropped, delayed, ...
- DiffServ-Tags within IP packets are ignored

Round-Trip-Time

- UTRAN (UE-SGSN): ~120ms, Core Network: ~20ms
- Very long slow-start phase with TCP
- slow reaction on packet losses

1.4 Charging user data within UMTS

LUCENT Technologies:

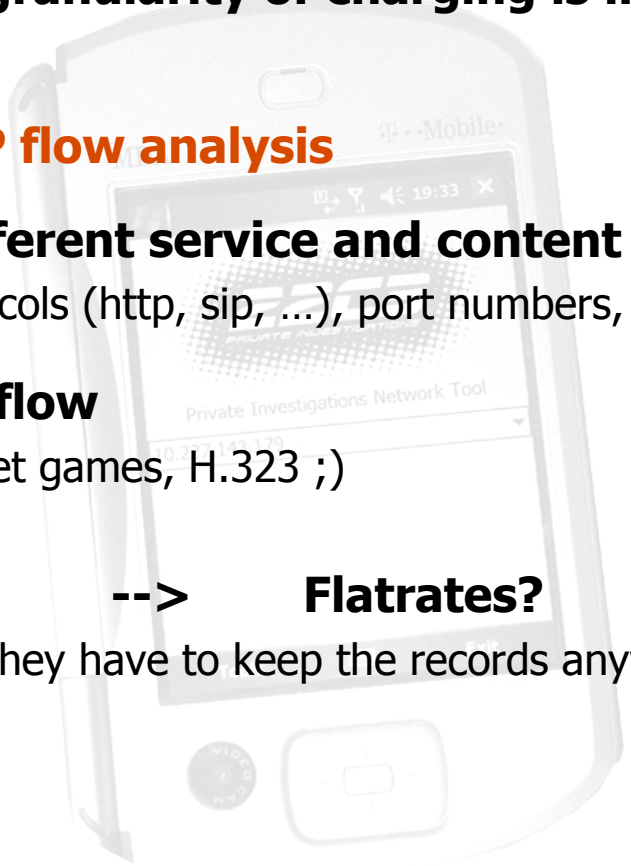
"It is widely accepted that billing will be a major issue for UMTS network operators; traditional telephony charging strategies, based on the duration and distance of a call, are no longer sufficient for 3G systems. Instead, sophisticated billing systems are required, that enable network operators to charge their customers according to complex criteria such as:

- **type of data/service**
- **transaction duration**
- **radio interface usage**
- **destination & source address**
- **location specific services**
- **bandwidth usage**
- **Quality of Service (QoS)"**

http://www.lucent.com/products/solution/0,,CTID+2019-STID+10488-SOID+1277-LOCL+1,00.html

1.4 Charging user data within UMTS

- **At the moment granularity of charging is limited to the pdp context**
- **New concept: IP flow analysis**
  - **Based on different service and content type**
    e.g. URLs, protocols (http, sip, …), port numbers, …
  - **Based on IP flow**
    e.g. P2P, Internet games, H.323 ;)
- **Too much data? --> Flatrates?**
  (But in the EU they have to keep the records anyway…)



Conclusions Part 1...

- **The UMTS packet switched domain is far more complex than technologies like WLAN or WiMAX.**
- **Most interesting part is the GGSN where IP packets are filtered charged and perhaps delayed.**
- **With the upcoming IP Multimedia Subsystem charging of individual IP flows will become of interest.**
- **We should try to have some fun with IP flow analysis. Probably a lot of hack value is waiting for us ;)**

2. Get to know your network

2.1 Different UMTS Network Designs – Vodafone D2

- 4662-4771 emule
- 5060 sip

- **Internal private network (10.0.0.0/8), no IPv6**
- **Network Address Translation:** Ext. 80.226.218.203/n
- **Transparent Application Layer Proxies**
  - e.g. thumbnails for web images
- >**100kByte/s**, sometimes with cacheable data (see below)
- **Packet lost up to 3% even on „good links"**
  -> that's not really good for TCP
- **Very long delays** **(with 95% confidence)**
  - from **293ms +-1,3ms** up to **449ms +-9,6ms**
- **Packetfiltering/-delaying?**
  - IP Packets with „record route" enabled are dropped
  - Not packet forwarding between the user equipment
  - DNS and TCP seems to have a slightly higher priority
  - Cacheable data seems to perform best
  - IPerf(TCP) doesn't perform well -> cased by tr. proxies?
  - Filesharing, but perhaps more filters from July 2007 on...

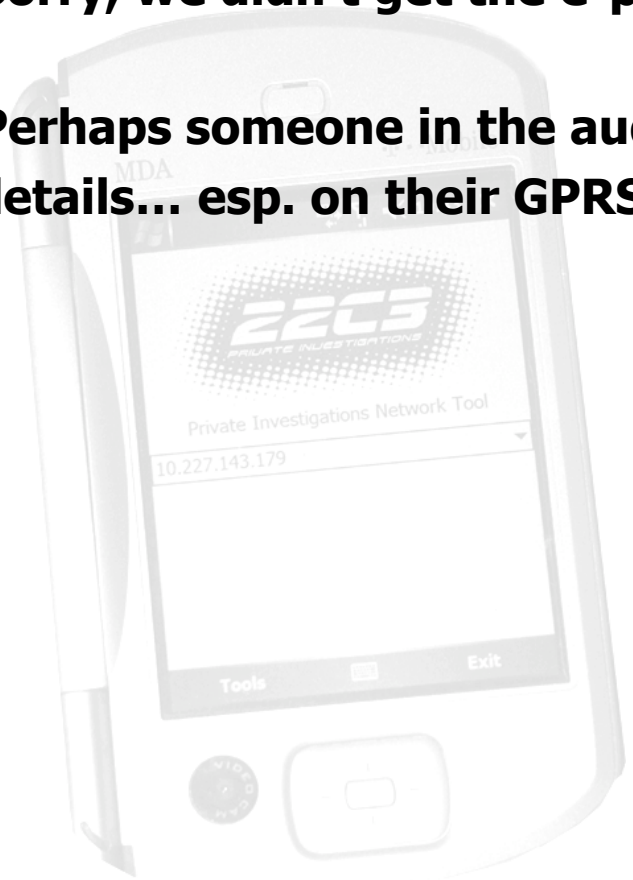
2.1 Different UMTS Network Designs – T Mobile D1

- **Very bad coverage (getting better every day)**
  - Due to constant switch between GPRS and UMTS: packet loss and huge delays (up to 2-4 sec.)
- **They use official IPv4 address space**
  - It just looks like the v4 adresses are packet filtered anyways
- **They do have a "transparent" http proxy which tries to reduce data transfers (JPEG transcoding,...)**
- **Very long delays: from 200ms up to 600ms**
- **Because of bad coverage situation the measurements are not representative**

2.1 Different UMTS Network Designs – E plus

- **Sorry, we didn`t get the e-plus account yet…**

- **Perhaps someone in the audience can give details… esp. on their GPRS/UMTS Flatrate**



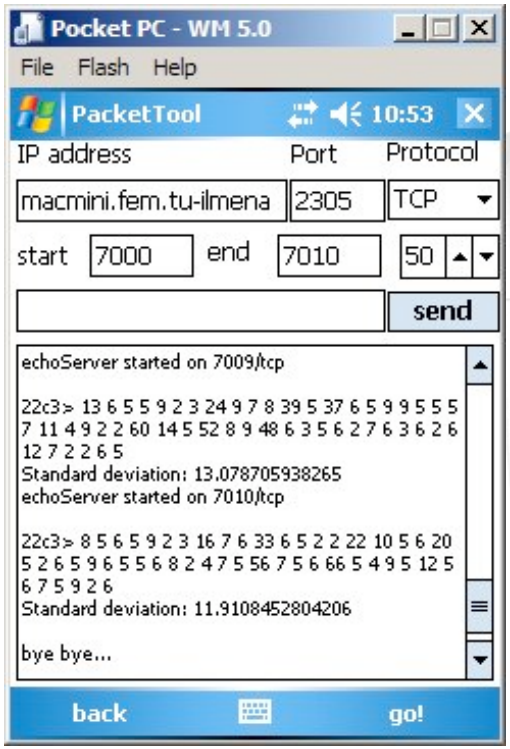
2.2 Some basic measurements

You all know the standard tools for network analysis like:

- **ping / ping –R** (Delay measurements / „record route")
- **traceroute, tcptraceroute** (Network routing)
- **iperf** (Bits/s-measurements via TCP or UDP)
- **echoping** (Delay measurements)
- **nmap** (Portscanner)

But, we are looking for a more integrated tool which could measure the latencies and the routing of packets using different ports, different protocols and „slightly modified packets"…

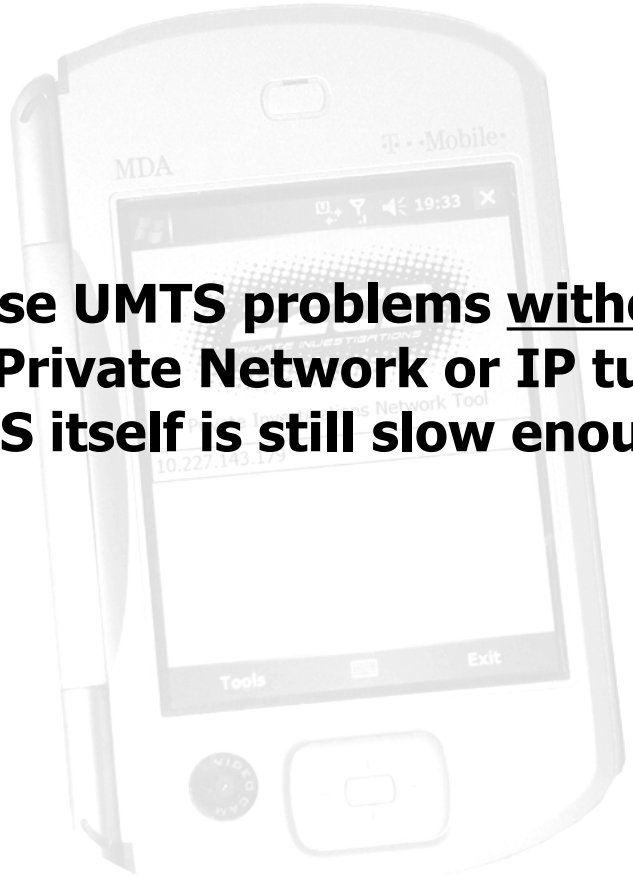
2.3 Some more advanced measurements



- **We wanted to have some easy programmable and mobile devices to play with**
  - "T-Mobile MDA pro" aka HTC Universal
  - Windows Mobile 5 UMTS/GPRS/WiFi/Bluetooth
  - We made a client application in C# (.NET compact framework) running on the mobile device and a server in JAVA (will be replaced by a C# version in the future) running "in the internet"
  - Server provides echo functionalities for udp and tcp packets
  - .NET CF 1.1/2.0 issue: no RAW IP support – we have to pInvoke
  - Sourcecode will be available for download at technology-ninja.com / ahzf.de

3. Adapt your traffic patterns

3. Adapt your traffic patterns

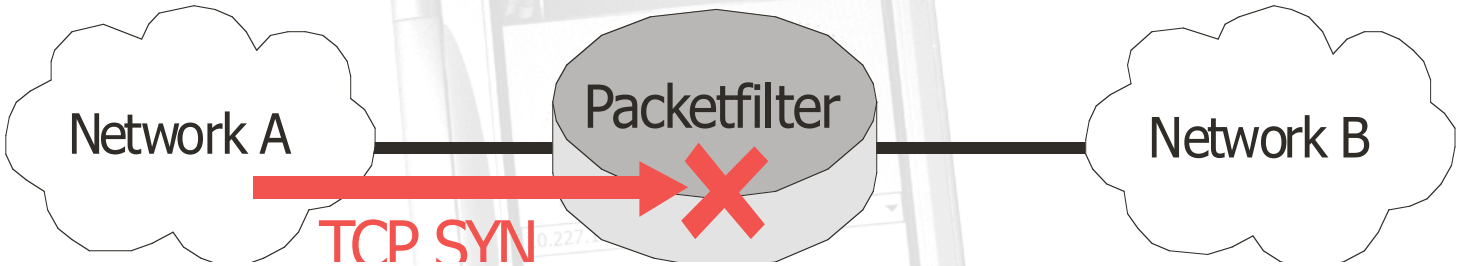


Design rule:

Get around these UMTS problems without using any kind of Virtual Private Network or IP tunnel for the main data flow. UMTS itself is still slow enough… ;)

3.2 OpenSYN/the problem

An easy example:



stupid packet filter:

No TCP connection setup from A to B allowed.

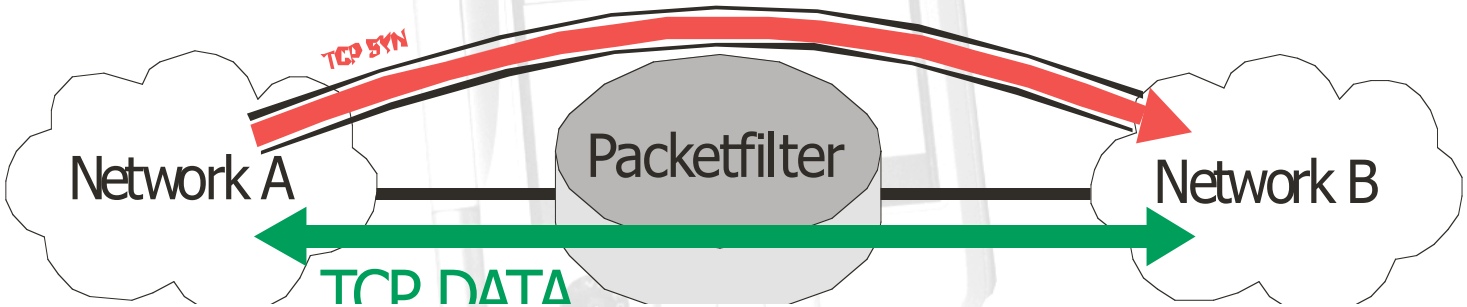
No limitations the other way round.

In both networks is at least one computer under your control…

3.2 OpenSYN / minimal tunneling

The "minimal tunneling" approach:

- **Most packet filters rely on filtering the TCP SYN packets**
- **So… adapt the routing of these packets, and only of these**
- **This is done to keep the overhead as small as possible**



3.2 OpenSYN / minimal tunneling

A very easy example…

using Linux netfilter, iproute2 and OpenVPN

1. **Setup your IP tunnel the same way you normally would do it**
2. **Register an new routing table**
   > echo 201 OpenSYN >> /etc/iproute2/rt_tables
3. **Create a rule using firewall marks and add routing entries**
   > ip rule add fwmark 2 table OpenSYN
   > ip route add default via TUNNELENDPOINT table OpenSYN
4. **Tell netfilter which packets to mark**
   > iptables –t mangle -A OUTPUT -d NET_B -p tcp --syn -j MARK --set-mark 2

(The solution is suboptimal if you have to deal with return path filtering!)

3.2 OpenSYN / minimal modification

The "minimal modification" approach:

- **Most packet filters rely on filtering the TCP SYN packets**
- **So… let it look like a response by setting the ACK bit**
- **Use netfilter connection tracking to indicate session setup**

1. **On the externale node**

   **> iptables -t mangle -d DESTINATION -A POSTROUTING -p tcp --tcp-flags SYN SYN **

   > -m state --state NEW –j QUEUE

2. **On the internal node**

   **> iptables -t mangle -s SOURCE -A PREROUTING -p tcp --tcp-flags SYN,ACK SYN,ACK **

   > -m state --state NEW –j QUEUE

3. **Use perl NetPacket::* and perl IPTables::***

   > $msg = $queue->get_message();

   > $queue->set_verdict($msg->packet_id, NF_ACCEPT, …);

(idea by ambanus)

Conclusions…

- **UMTS is a great technology for accounting, charging and with HSDPA and HSUPA perhaps even for using it ;)**
- **If there are any IP filter, delays or special charging in the future we want to find ways to deal with them… today.**
- **This could lead to an automatic creation of traffic-delay maps and adapting our packet flow by using "minimal modified packets".**
- **"Big hammer"-solutions like VPNs are not always the best approach… give smarter techniques a chance.**

Thank you… Any questions?

http://www.ahzf.de/itstuff/VoE/
http://www.technology-ninja.com

Mailinglist: projekt-voe@fem.tu-ilmenau.de
Subscribe: majordomo@fem.tu-ilmenau.de?subject=subscribe%20projekt-voe

Appendix A: Design your own delayed network

Using Linux netfilter, traffic control and netem…

1. **Setup prio qdisc for normal traffic and delayed one with netem**
   > tc qdisc add dev eth0 root handle 1: prio
   > tc qdisc add dev eth0 parent 1:3 handle 30: netem delay 300ms

2. **Filter all traffic tagged with fw mark 6 to the delayed qdisc**
   > tc filter add dev eth0 protocol ip parent 1:0 prio 3 handle 6 fw flowid 1:3

3. **Tell netfilter which packets to mark**
   > iptables -t mangle -A POSTROUTING -p tcp --dport 23 -j MARK --set-mark 6