Traffic Monitoring for 3G Network Diagnostic: a Doctor's View

Dr. Fabio Ricciato

This talk

Goal

share lessons learned about the role of
traffic measurements data → *signals*
for the diagnosis
of network problems
in a 3G mobile network → *system*

Outline

- Background
- Introduction to 3G cellular networks
- Traffic monitoring and Network Signals for Diagnostics
- Example: detection of congestion bottleneck

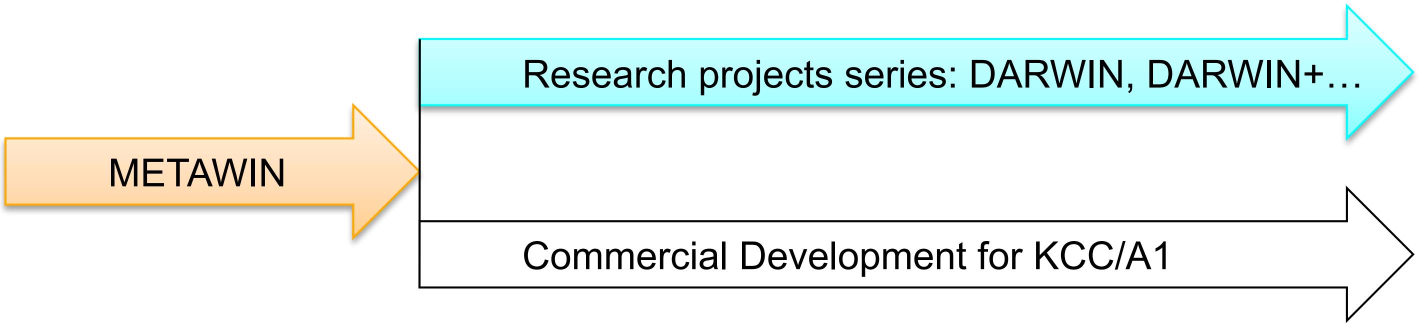
Research on 3G traffic monitoring @FTW

- Research on 3G Traffic Monitoring @FTW began in 2004
- METAWIN project
  - Measurement and Traffic Analysis in Wireless Network
  - partners: network operator, system integrator, university
- Goals
  - Sniff packets in the 3G core Network (GPRS and UMTS)
  - Analyze traffic traces to support network planning
  - Understand "what's going on"

Research on 3G traffic monitoring @FTW

- Results from 1st project
  - Prototype of advanced **monitoring system**
    - Developed from scratch on Linux
    - Deployed in the live network of A1 for dual use (production + research)
  - Access to real network and monitoring data for research

- Follow-up projects
  - research: analysis of traffic data
  - commercial: development/extension of monitoring system



Introduction to 3G mobile networks

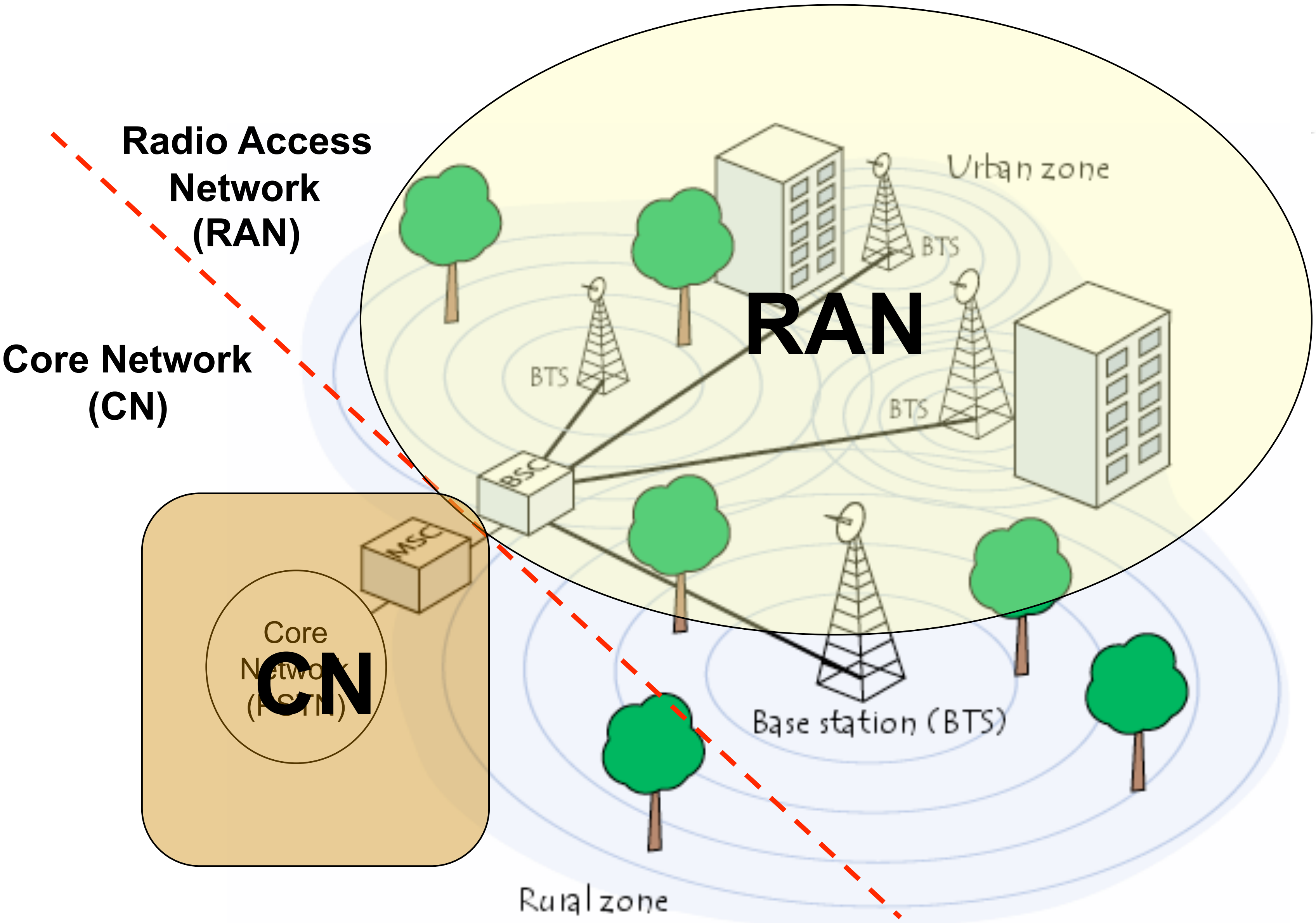
Circuits and Packets

- Two main design approaches for Communication Networks

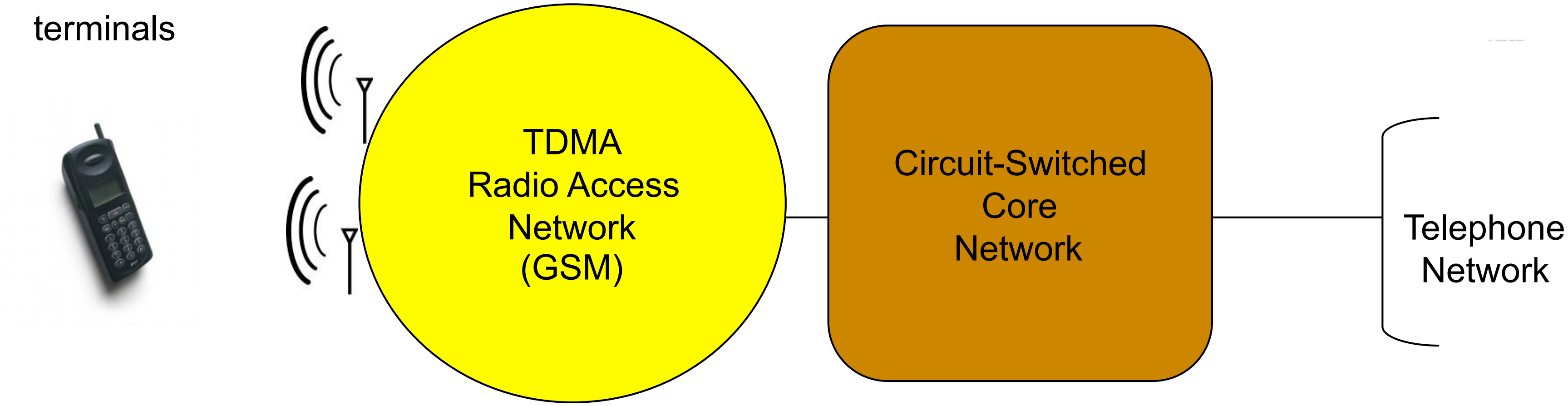
- Circuit-switched (CS)
  - nodes build a long pipe (→ "circuit") from source to destination
  - data (e.g. voice samples) travel into the pipe

- Packet-switched (PS)
  - data travel in independent chunks → "packets"
  - packets received, processed and forwarded independently by intermediate nodes

Architecture of 2G mobile network (GSM)

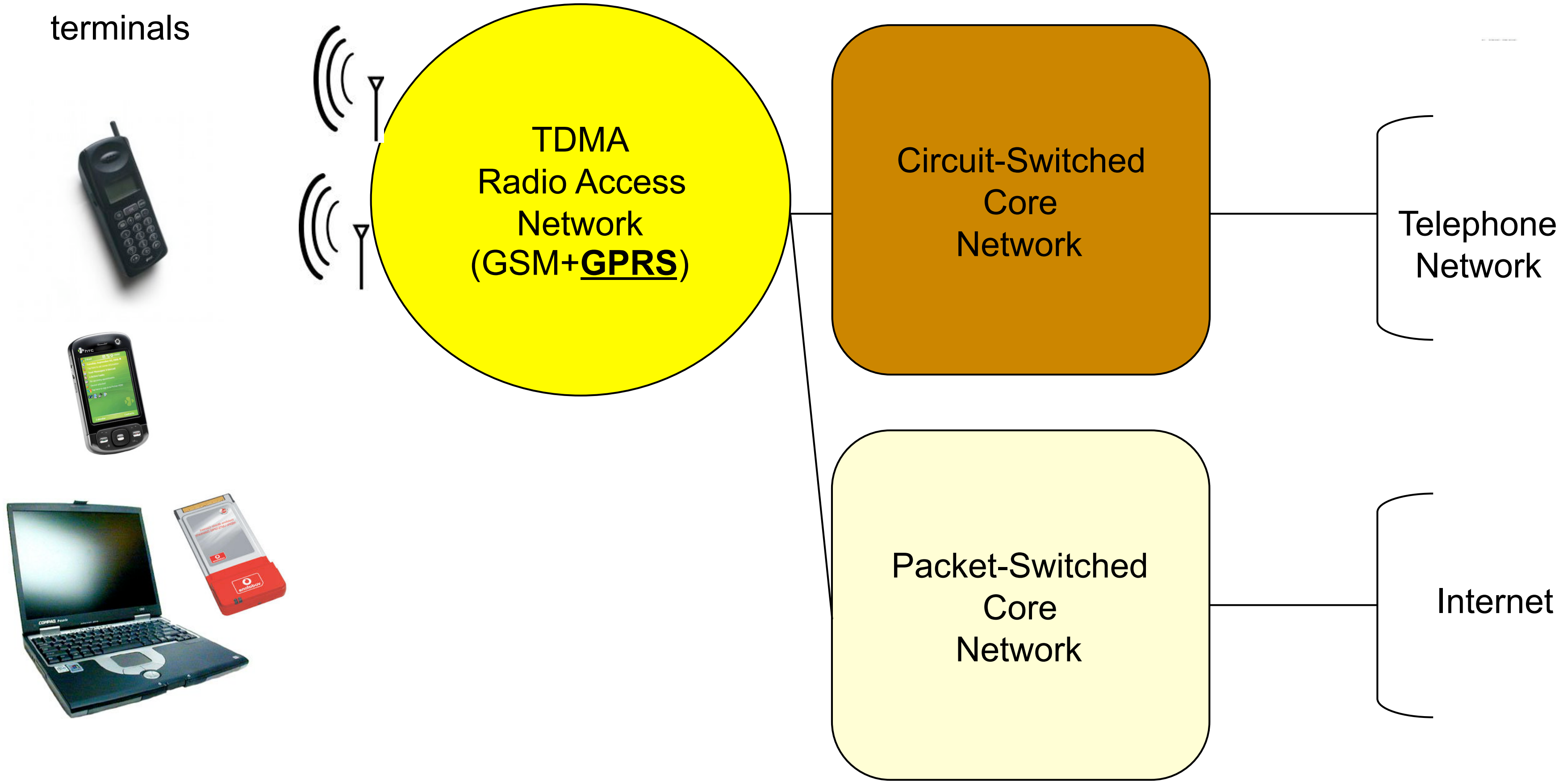


Evolution of cellular networks 2G (GSM) ...



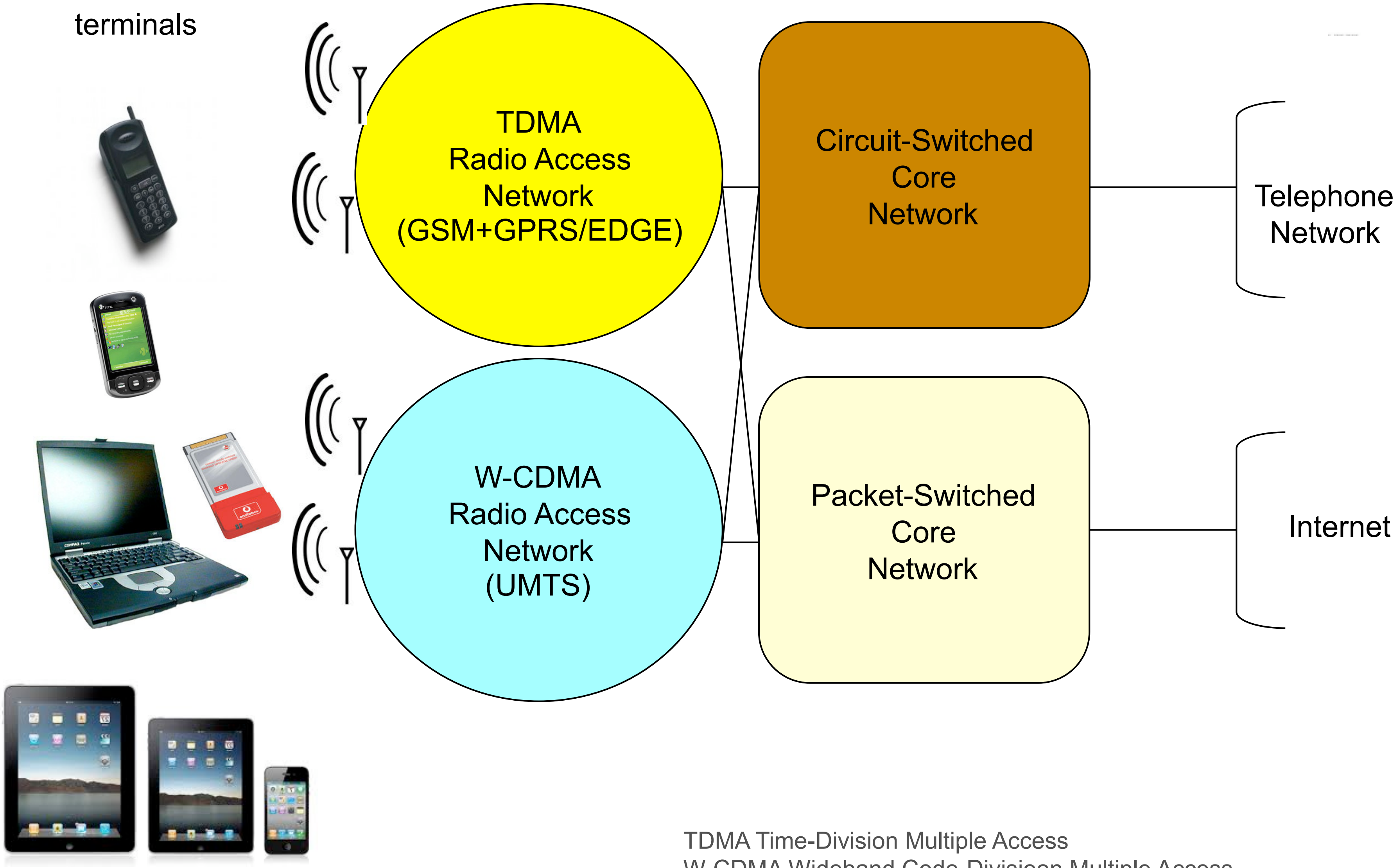
TDMA Time-Division Multiple Access

... 2.5G (GPRS) ...



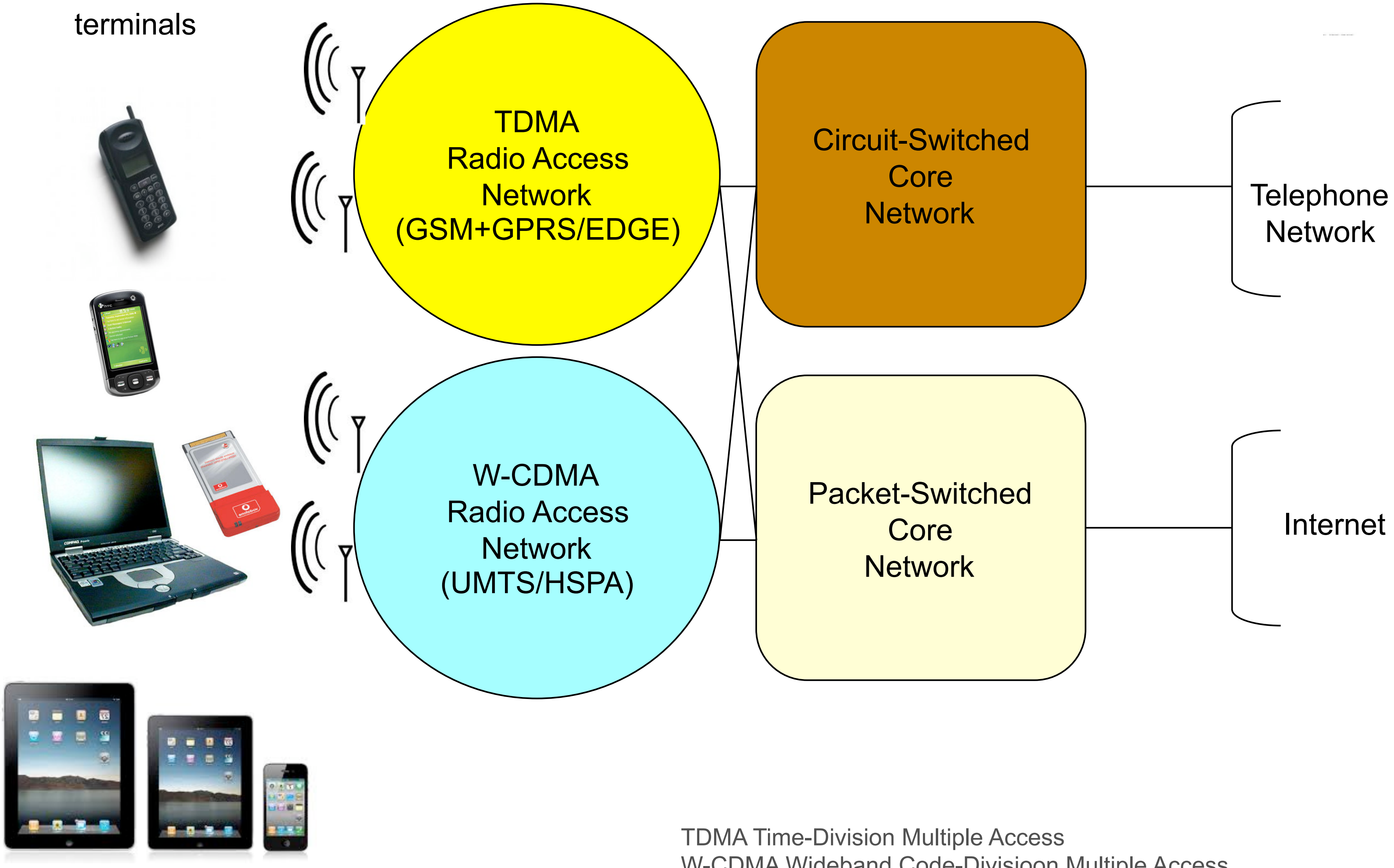
TDMA Time-Division Multiple Access
W-CDMA Wideband Code-Divisioon Multiple Access

... 3G (UMTS) ...



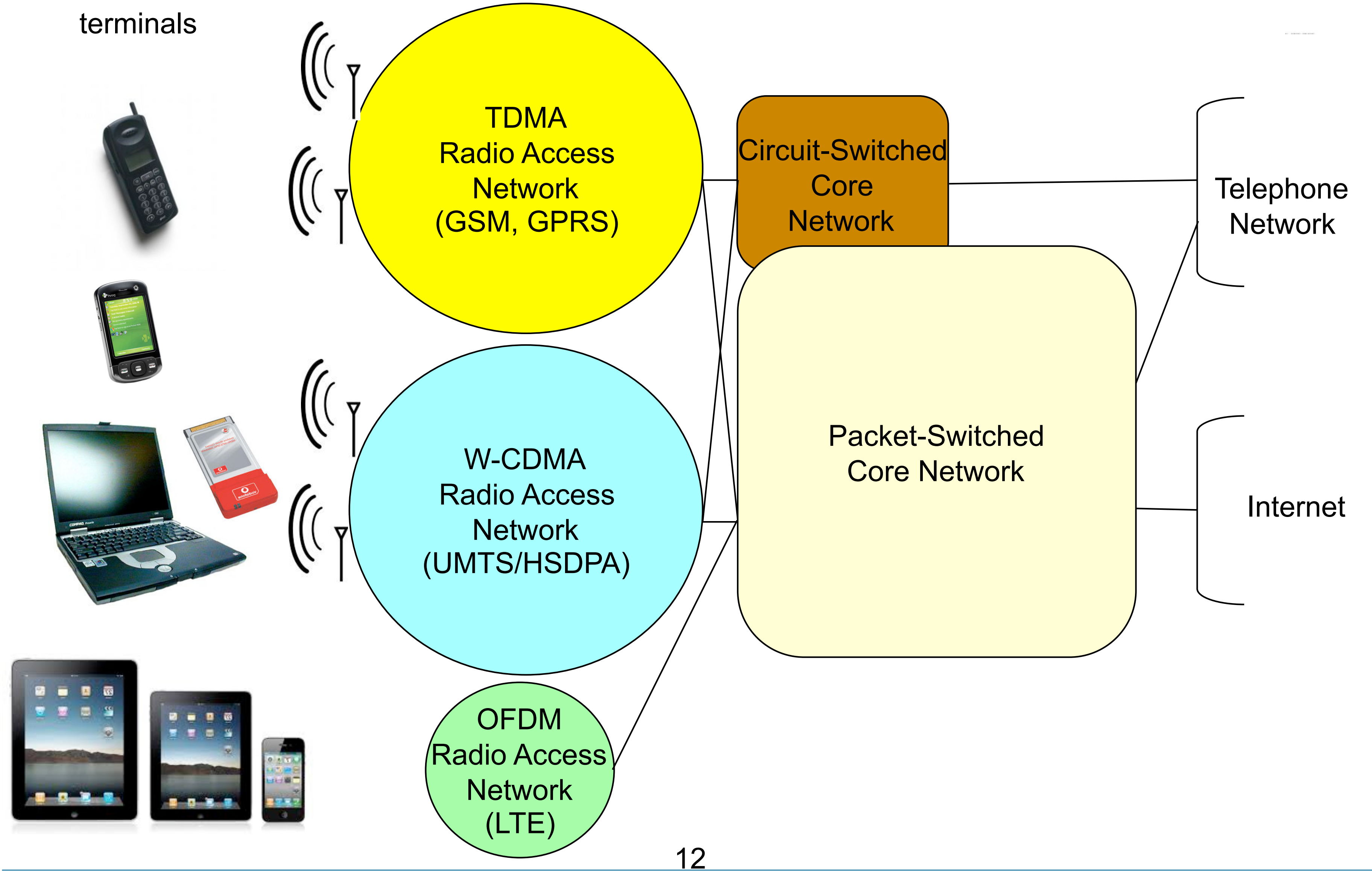
TDMA Time-Division Multiple Access
W-CDMA Wideband Code-Divisioon Multiple Access

...3.5G (HSPA)...



TDMA Time-Division Multiple Access
W-CDMA Wideband Code-Divisioon Multiple Access

… 4G (LTE/SAE) …

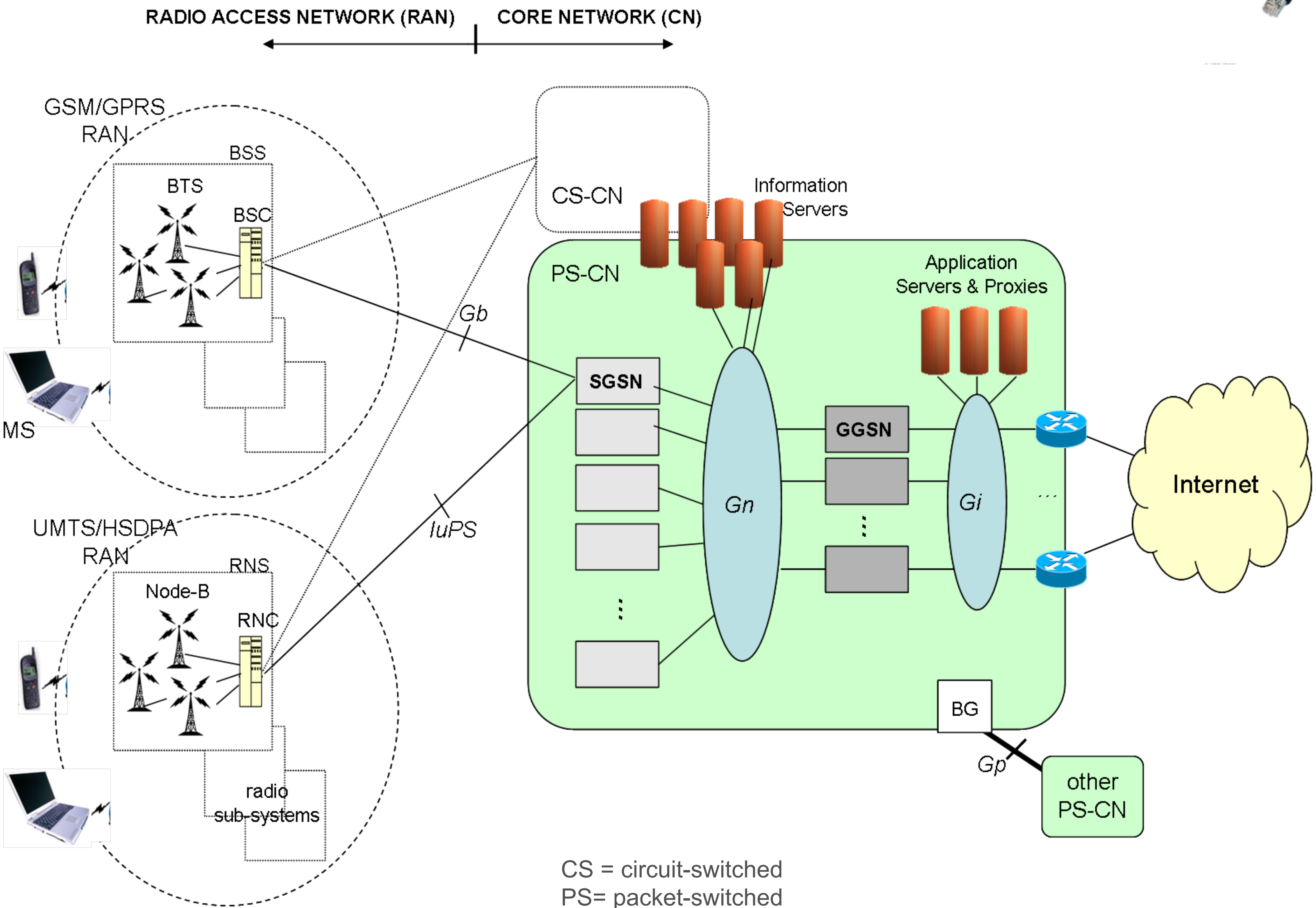

terminals
TDMA
Radio Access
Network
(GSM, GPRS)
W-CDMA
Radio Access
Network
(UMTS/HSDPA)
OFDM
Radio Access
Network
(LTE)
Circuit-Switched
Core
Network
Packet-Switched
Core Network
Telephone
Network
Internet

It keeps changing …

- Cellular Network is continuously evolving system…
  - architecture evolves: GSM, GPRS, EDGE, UMTS, HSPA, LTE/SAE
  - upgrade/replacement of network equipments
  - new SW releases, new network features
  - capacity upgrades: more radio bandwidth, higher link speed

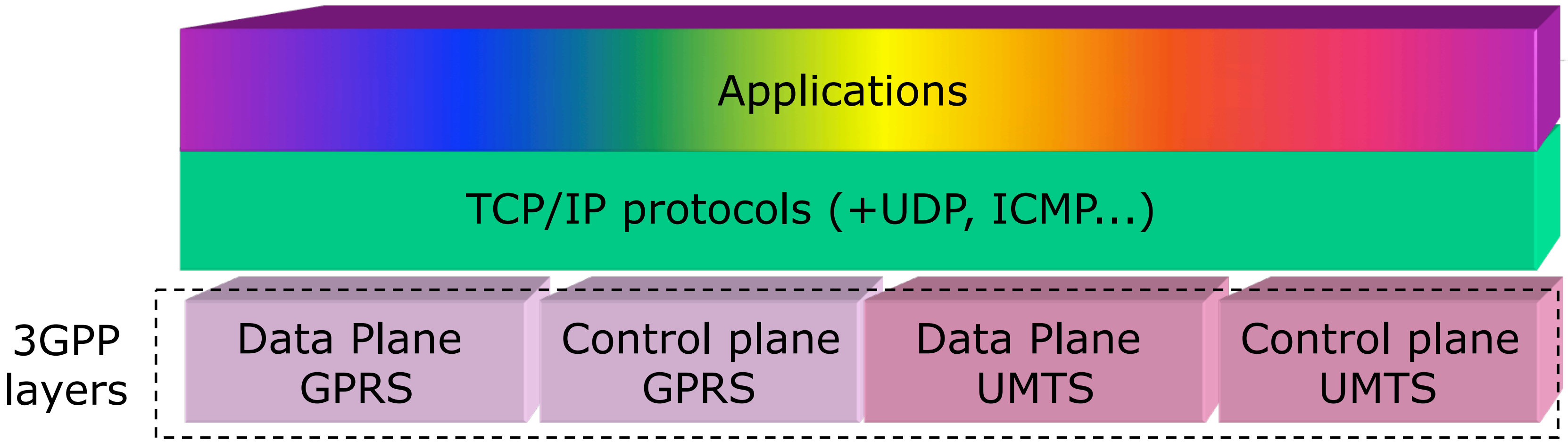
- … embedded in a continuosly evolving usaeg environment
  - more 3G **users**, increasing bandwidth demads, changing traffic patterns
  - more **terminals**, of new classes (laptop, smartphone, tablets… Internet of Things…) and capabilities
  - evolution of **applications**, apps, services
  - new habits: mobile tethering, wifi offloading

Representation of Architecture

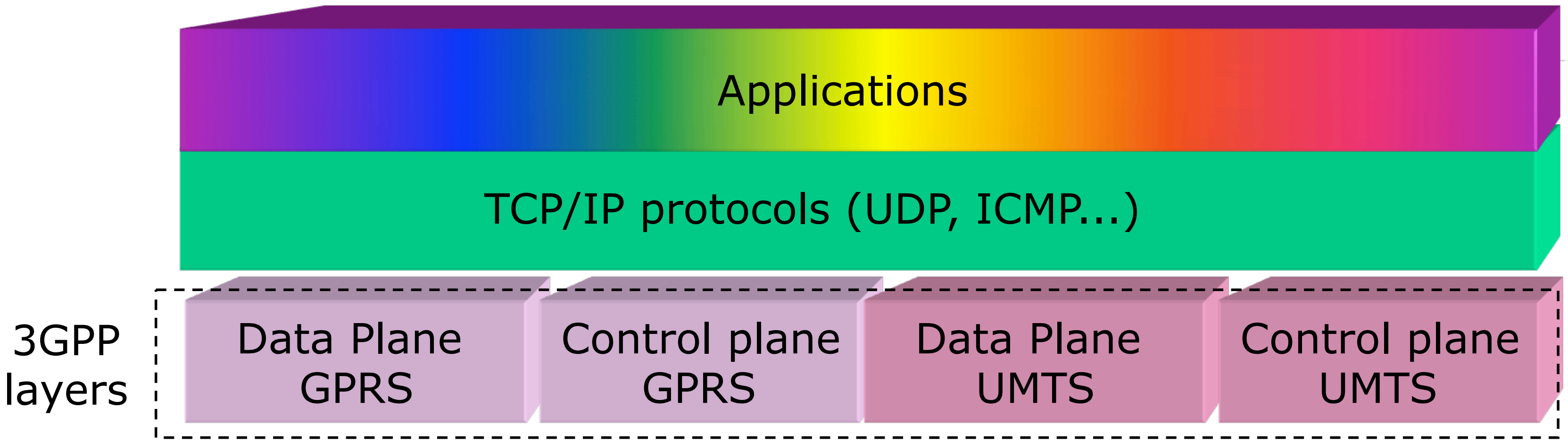


CS = circuit-switched
PS= packet-switched

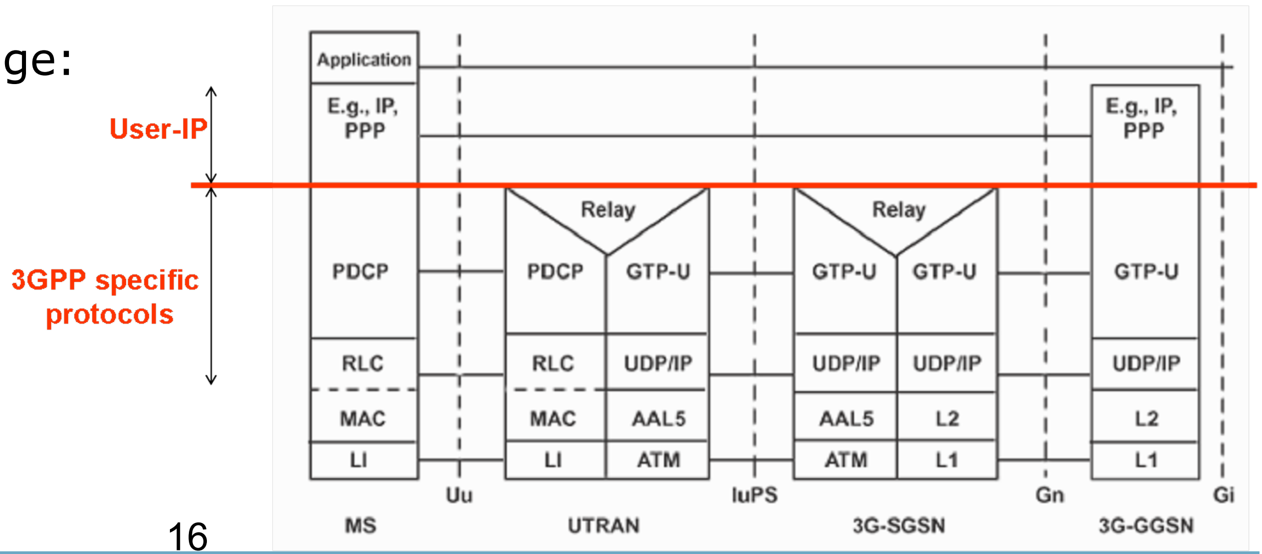
Representation of Protocol Stack



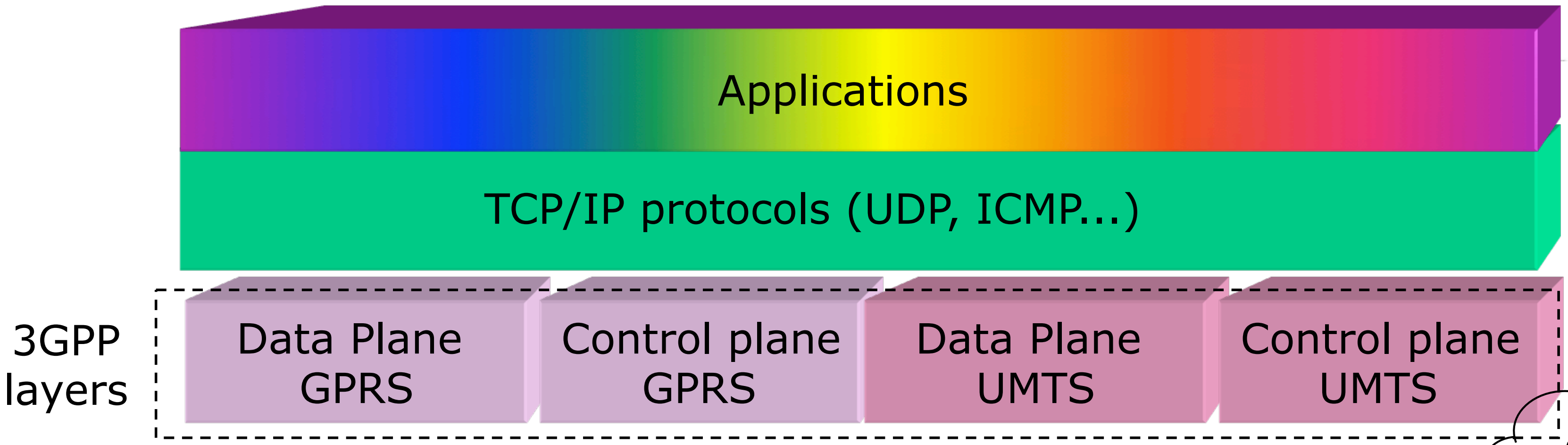
Representation of Protocol Stack

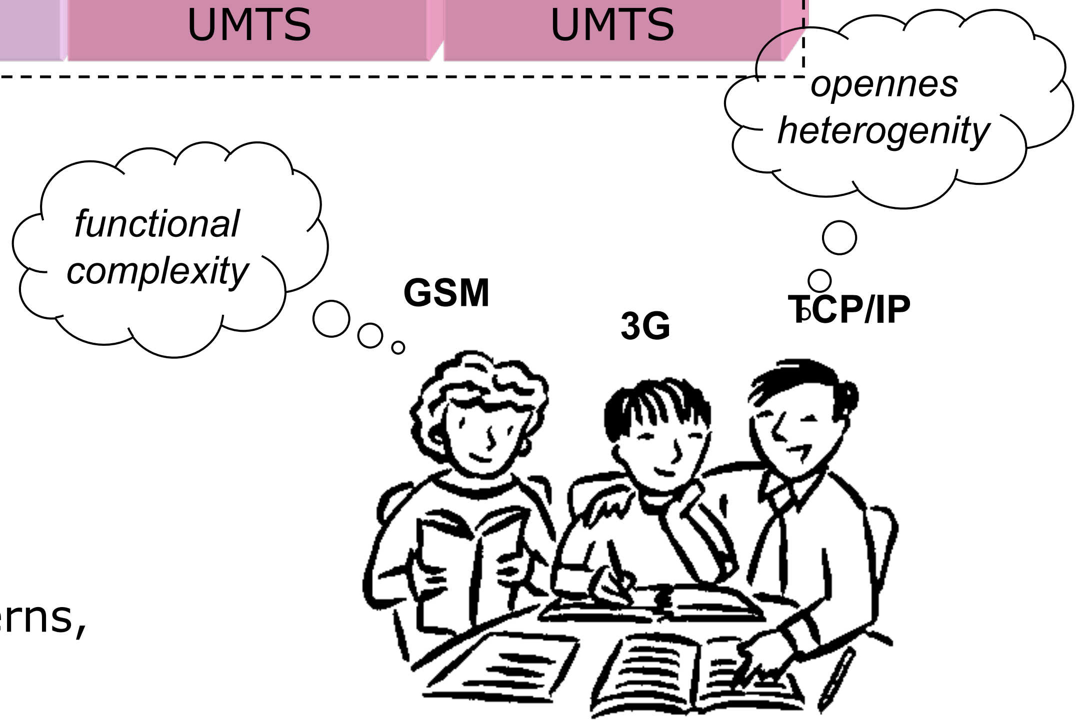


- Control Plane needs to Manage:
  - Radio Resources RRM
  - Mobility MM
  - Connections CM



Representation of Protocol Stack



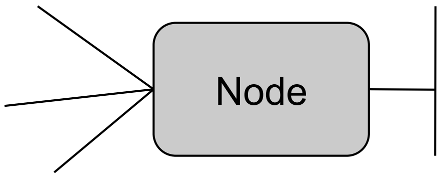


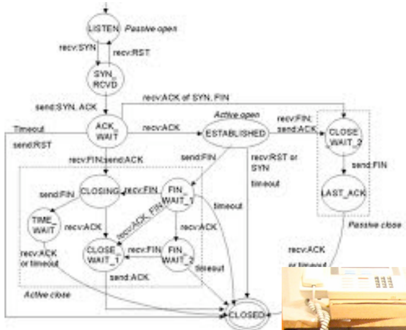
- Control Plane needs to Manage:
  - Radio Resources RRM
  - Mobility MM
  - Connections CM

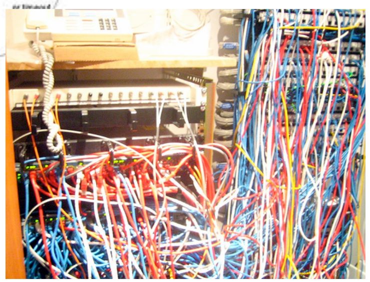
- Diversity of applications, traffic patterns, open reachability

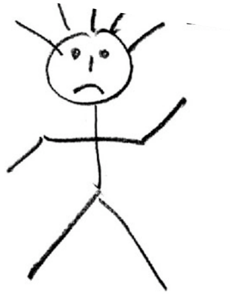
Analogy with human body

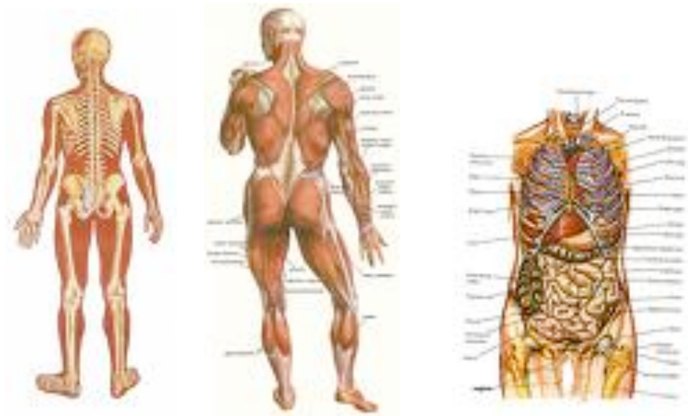
- The real **infrastructure** is much more complex than *any* of its schematic **representation**
  - Physical and logical components, dependencies, functional layers … **like a human body!**

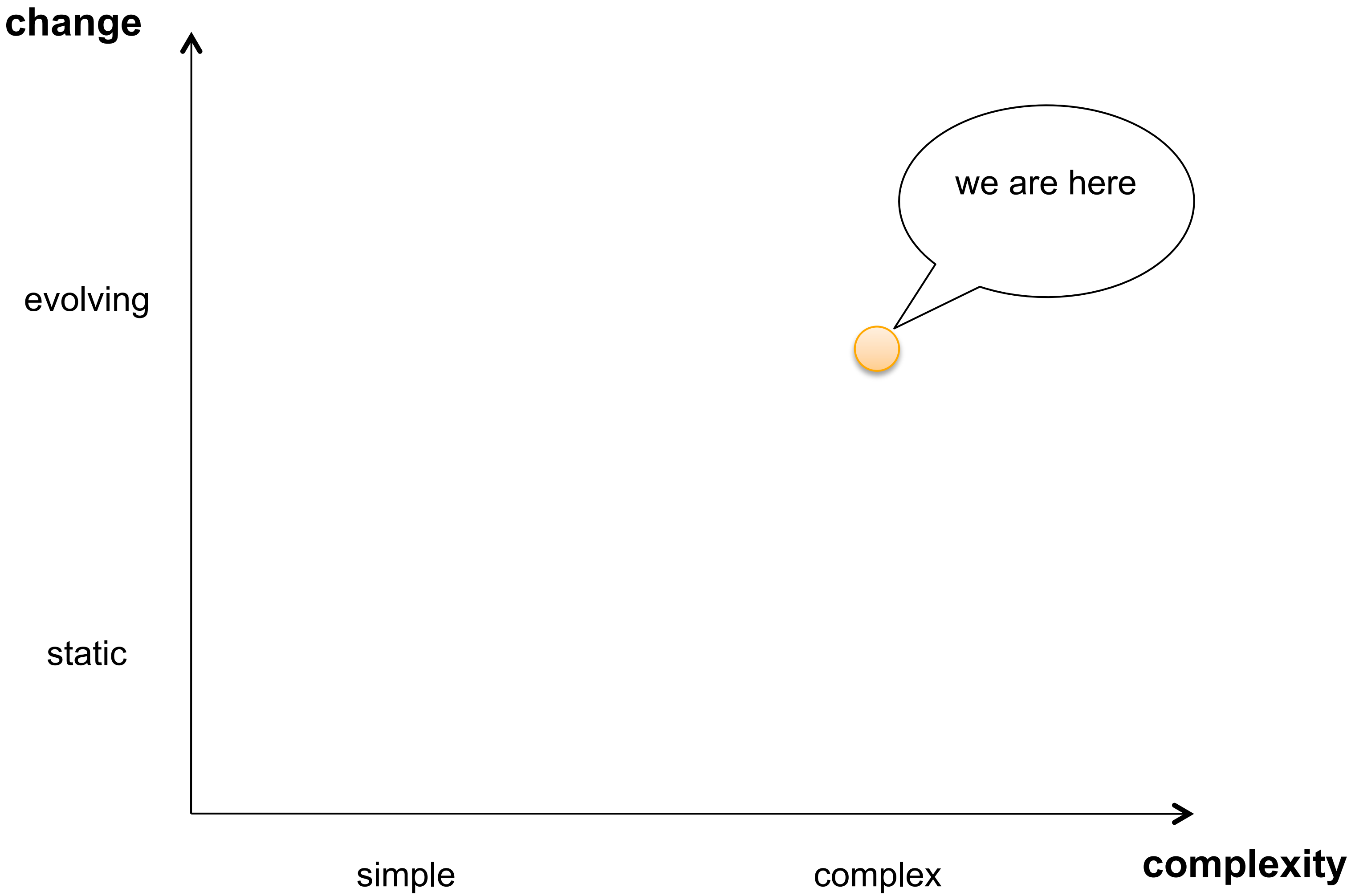











change
evolving
static
we are here
simple
complex
complexity

… keeps watching it

- 3G networks are <u>complex</u> and <u>evolving</u> systems
- → new risks, problems, anomalies arise continuously …
- **Endogenous:** congestions, misconfiguration, failures, malfunctioning
- **Exogenous:** attacks, large-scale infections

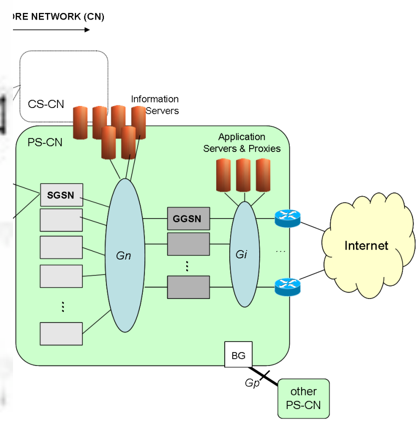
- To operate correctly the network infrastructure, its **health** status must be monitored continuously, to reveal problems/ anomalies as early as possible
- → diagnosis & troubleshooting are continuous processes (not one-shot tasks)
- *How network monitoring can help the <u>process</u> of diagnosis and troubleshooting ? What are the difficulties & challenges?*

Like a Doctor

- How ?
  - Observe many “**signals**” from the infrastructure, obtained with non-invasive methods
  - Reveal “**symptoms**” of abnormal conditions and interpret possible causes → “**diagnosis**”

- Challenges
  - define the “right” signals (cost vs. benefit)
  - detect “abnormal conditions” (symptoms)
  - interpret the root cause (diagnosis)

- Like in Medicine …
  - coping with “soft” definitions
  - some irreducible level of subjectivity



Introduction to Network Monitoring

Network Monitoring

- How the operator can monitor its network

- 1. Ask the boxes
  - Collects logs and alarms from network equipment themselves
  - Limited amount of data, coarse accuracy, sometimes unreliable

Network Monitoring

- How the operator can monitor a network

- 1. Ask the boxes
  - Collects logs and alarms from network equipment themselves
  - Limited amount of data, coarse accuracy, sometimes unreliable

- 2. Active measurements
  - Send end-to-end probe traffic
    through the network
    e.g. test downloads, pings

Network Monitoring

- How the operator can monitor a network

- 1. Ask the boxes
  - Collects logs and alarms from network equipment themselves
  - Limited amount of data, coarse accuracy, sometimes unreliable

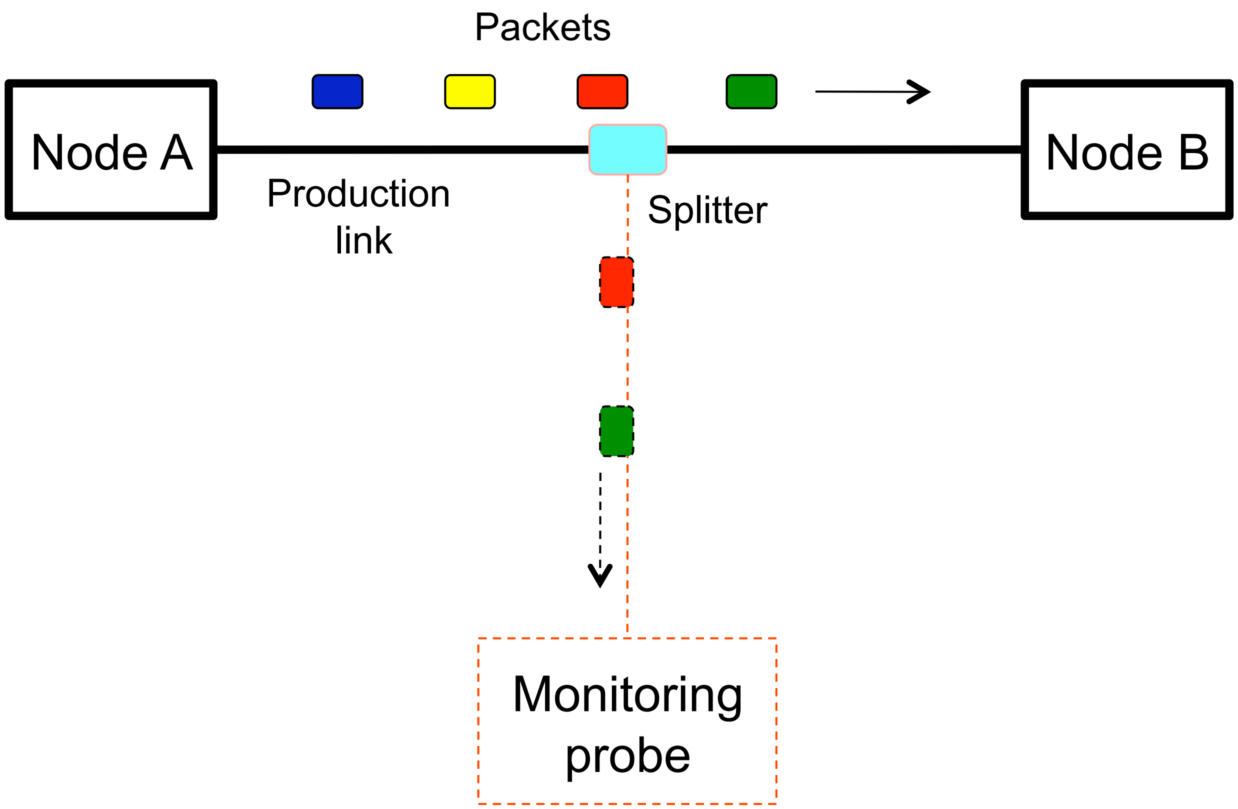
- 2. Active measurements
  - Send end-to-end probe traffic through the network
    e.g. test downloads, pings

- **3. Passive monitoring**
  - "sniff" packets on the wire
  - Non invasive, but requires monitoring HW installation

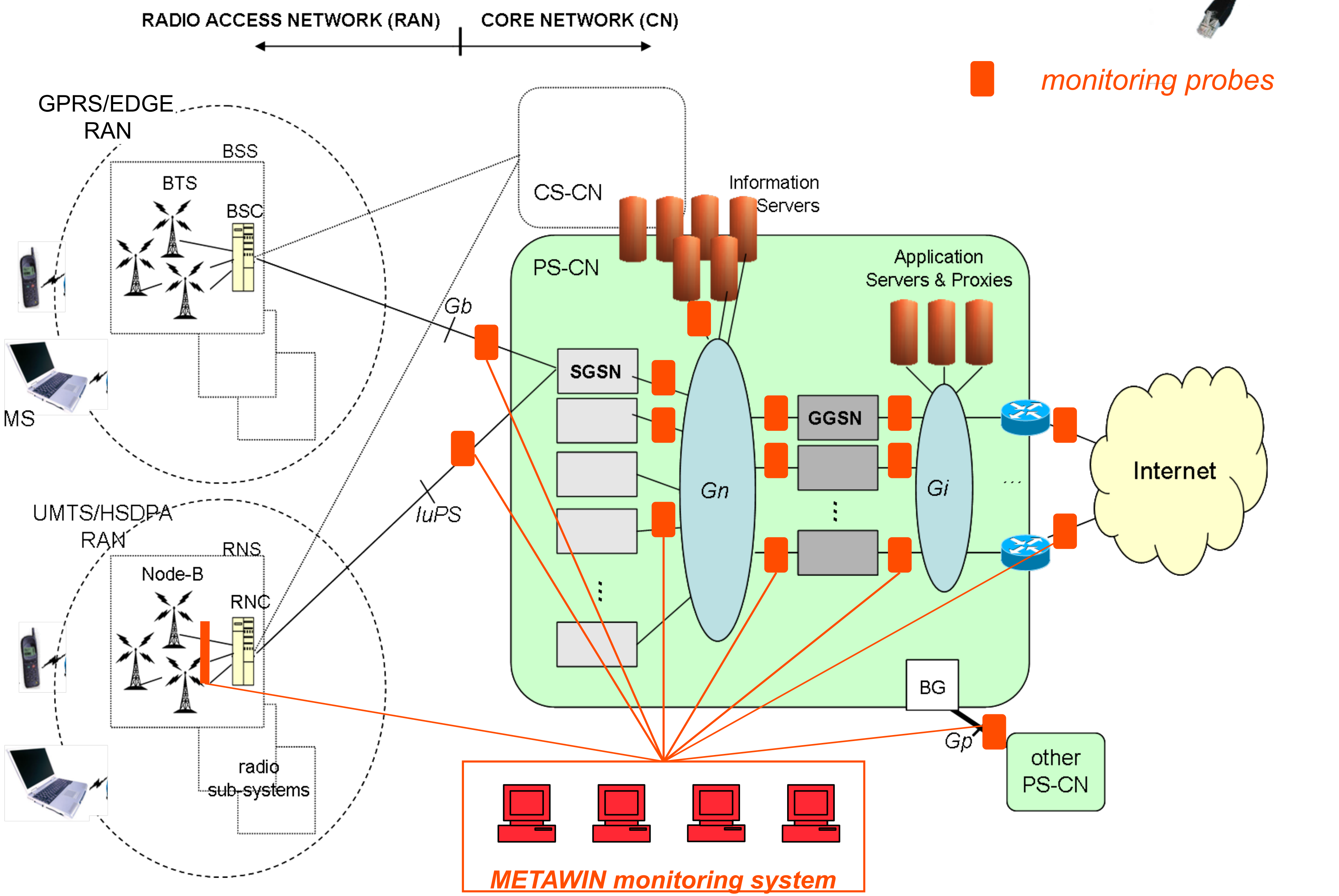
- Hybrid Measurements
  - send probe traffic and capture it passively inside the network

Passive monitoring

- Sniffing
  - pick a copy of the packet as it traverses the wire
  - Store it with additional information to each record
    - e.g. timestamps, capture interface, terminal type



Example of passive monitor deployment



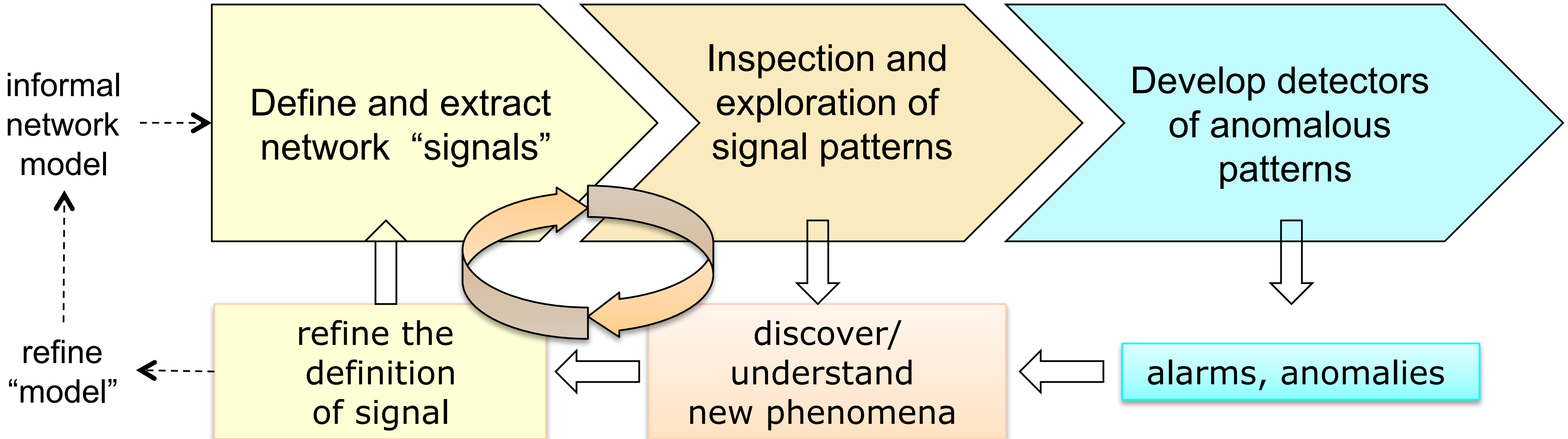
Network signals

- Examples of “network signals” related to performance metrics
  - one-point measurements on data plane
    - TCP Round-Trip-Time (RTT), Retransmissions, Timeouts
    - Throughput statistics (peak, mean,…), Download times
    - higher-layer Request/Response delay (DNS, HTTP, …)
  - two-point measurements on data plane
    - One-way delay
    - Packet loss
  - control-plane signals
    - Frequency of error messages, latency of signaling procedures

- Signals can be partitioned across several dimensions
  - Network section
    - e.g. traffic to server Y, to peering link X, to node Z …
  - User class, Terminal type
  - Application, Service … etc.

Considerations

- Basic principle: network problems will impact one or several "signals", causing abnormal patterns
  - congestion → higher loss and/or delay, lower flow throughput
  - observe the signal to infer back the problem



our mental model …

event A ⟹ **pattern on signal X**

eg. “congested link”

eg. “more flows getting low throughput”

To reveal this

We measure this

the reality …

event A ⟹ **pattern on signal X**

congested link

cause B

large-scale antivirus update

cause C

low-rate app getting popular

pattern on signal Y

a matter of definitions

- Q. How many sheeps in this field ?

- counting is easy …
- once you define clearly **what** to count
  - how is the “field” delimited ?
  - and what do you mean exactly by “sheep”?
    - are *lambs* counted as sheep, or only adults ?
    - do *pregnant sheeps* count for 1 or 2 ?
    - what about those *dead sheeps* over there ?
    - shouldn’t we count *goats* too ?
    - …

WHY do you want to count ? ⟵ WHAT to count ?

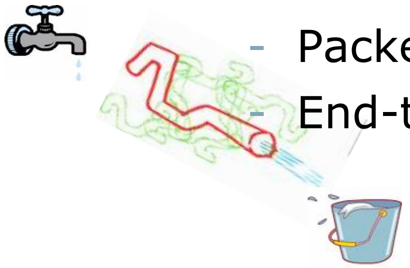
Examples of Passive Monitoring for Network Diagnostic

Longum est iter per praecepta, breve et efficax *per exempla*

Detecting congestion bottleneck

- Congestion bottleneck
  - Definition: too much traffic for too little capacity *at some point*

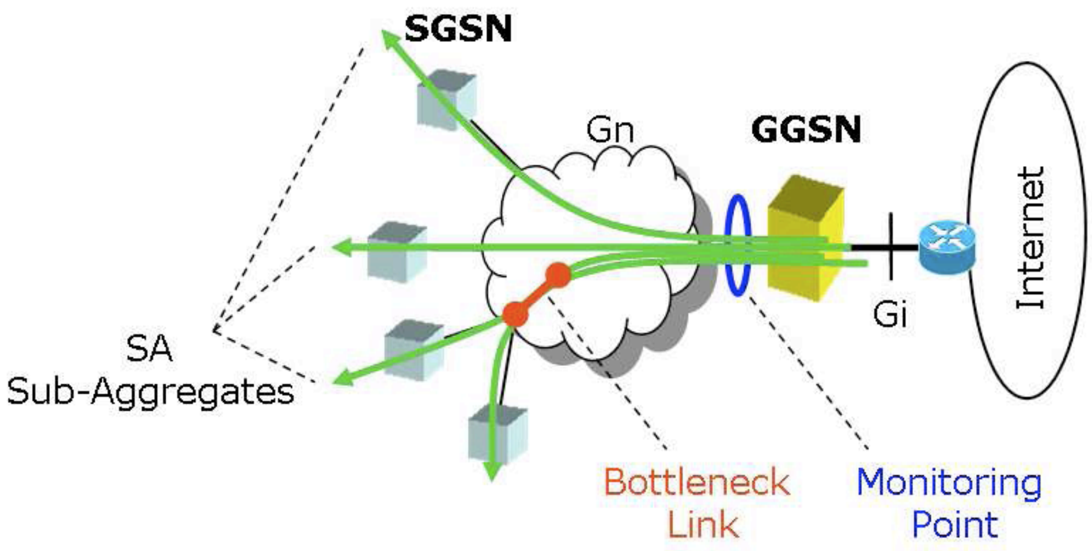
  - Effects on packets
    - Packets queued in buffers: longer transit delay
    - Packet drops: retransmissions
    - End-terminals react reducing the sending rate



  - Impact on users
    - Loooooong waiting for downloads
    - Interactive applications don't work: voip, skype, game…

Detecting congestion bottleneck

- Causes for congestion bottleneck
  - Traffic grew faster than provisioning cycle
  - gap between nominal vs. actual capacity (e.g. due to misconfigurations, misfunctioning, …)
- Problem: not *every* link/node can be monitored
- Goal: detect congestion on link/node X from the analysis of traffic at a different point M
  - without a priori information about *actual* link capacity
  - without topology information



Look at TCP

- Idea: look at TCP
  - TCP is closed-loop → protocol dynamics have end-to-end interactions → local problem on link X should be visible at any other path section
  - >90% of traffic is TCP

- Possible approaches: analysis of …
  - Distribution of Traffic Rate
  - Frequency of Retransmission Timeouts (RTOs)
  - Round-Trip Times (RTT)
  - Per-flow Throughput

Detecting congestion bottleneck

- Detecting congestion from aggregate rate analysis
  - a real case study

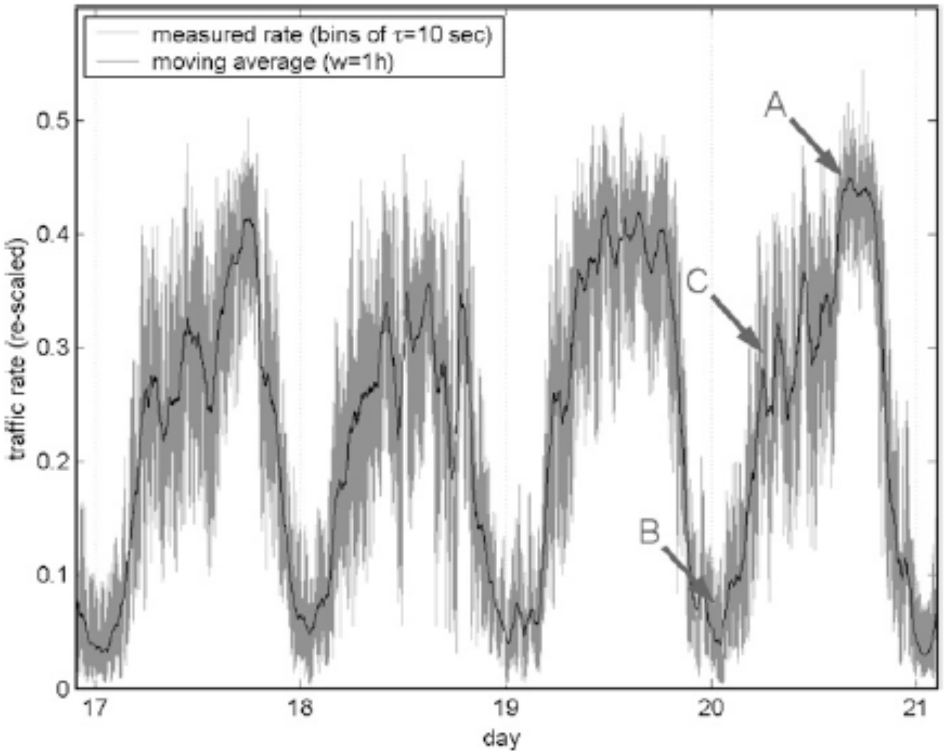


Fig. 3. SA total rate for days 17–20 (10 s bins).

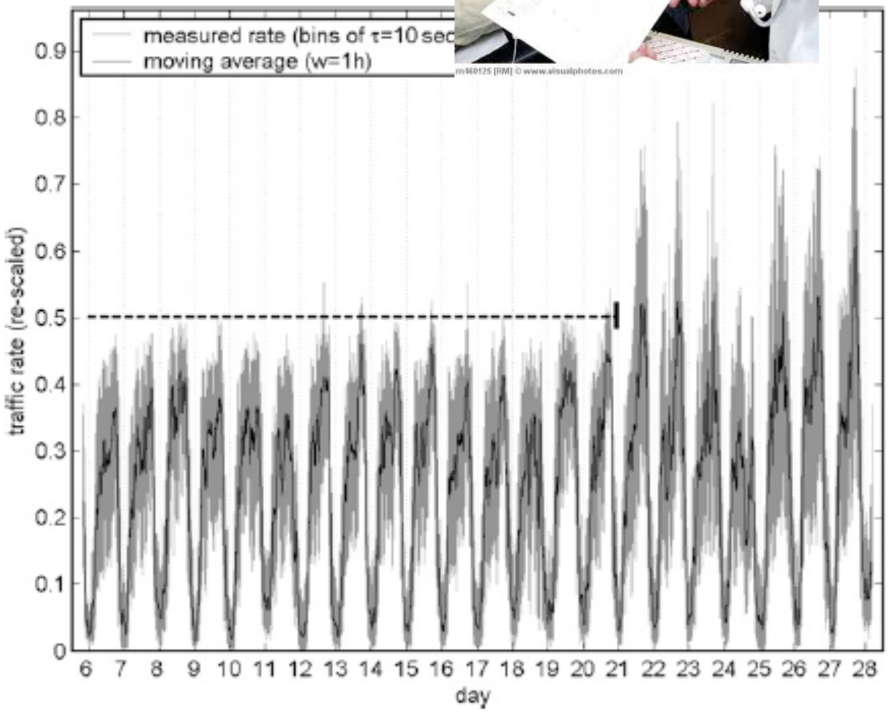


Fig. 2. SA total rate for the monitored period (10 s bins, rescaled values).

F. Ricciato, F. Vacirca, P. Svoboda, Diagnosis of Capacity Bottlenecks via Passive Monitoring in 3G Networks: an Empirical Analysis, Computer Networks, vol. 51, n.4, pp. 1205-1231, March 2007

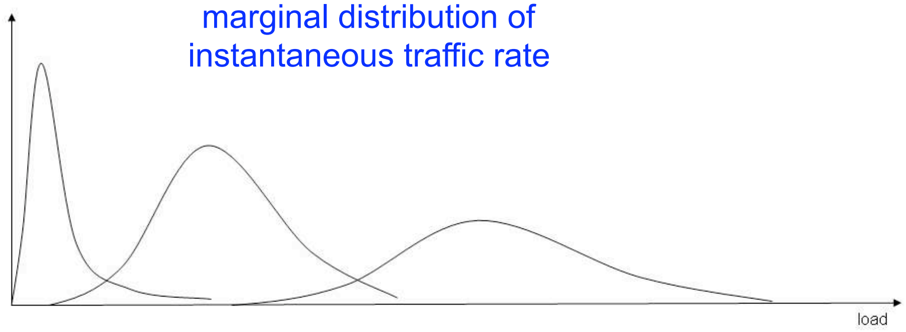
Aggregate Rate Analysis - Background

- Background
  - The simplest traffic model for infinite capacity: superposition of *independent* On/Off flows
    - Poisson arrivals of rate λ, holding time of mean ϑ
    - fixed rate r=1 (M/G/∞ queue)
  - Aggregate rate (marginal) is Poisson distributed
    - Variance = Mean
    - Skewness = Mean$^{-1/2}$

$$p_R(k) = \frac{e^{-A} A^k}{k!}$$

$$VAR(R) = E(R) = A$$

$$SKEW(R) = 1/\sqrt{A}$$

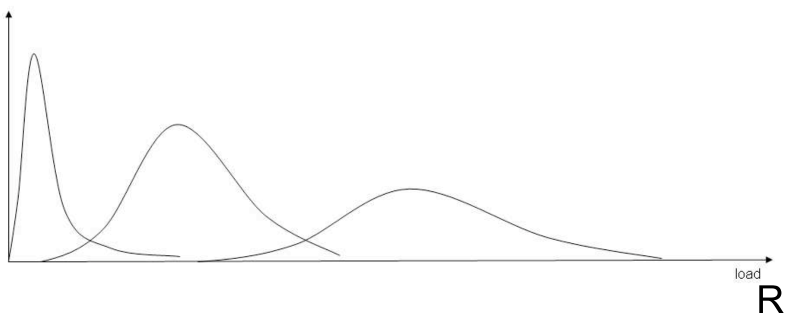


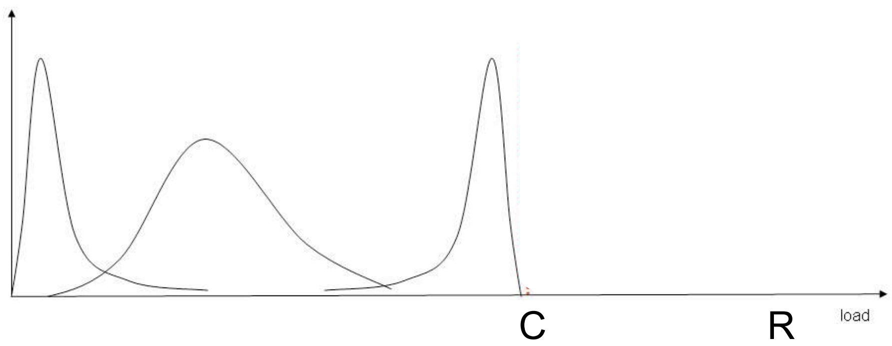
- Bottleneck-free conditions
  - Variance of marginal rate increases with mean load
  - Variance is higher at peak-hour
  - Skewness is lower at peak-hour

Aggregate Rate Analysis – the idea

- Conjecture on bottleneck-constrained traffic
  - Instantaneous rate R can not exceed path capacity C
  - Bottleneck induces *correlation* between flow rates as *R→C⁻*
    - via congestion control loop
    - reflecting barrier at R=C
  - Variance reduction and left-skewness as R→C⁻

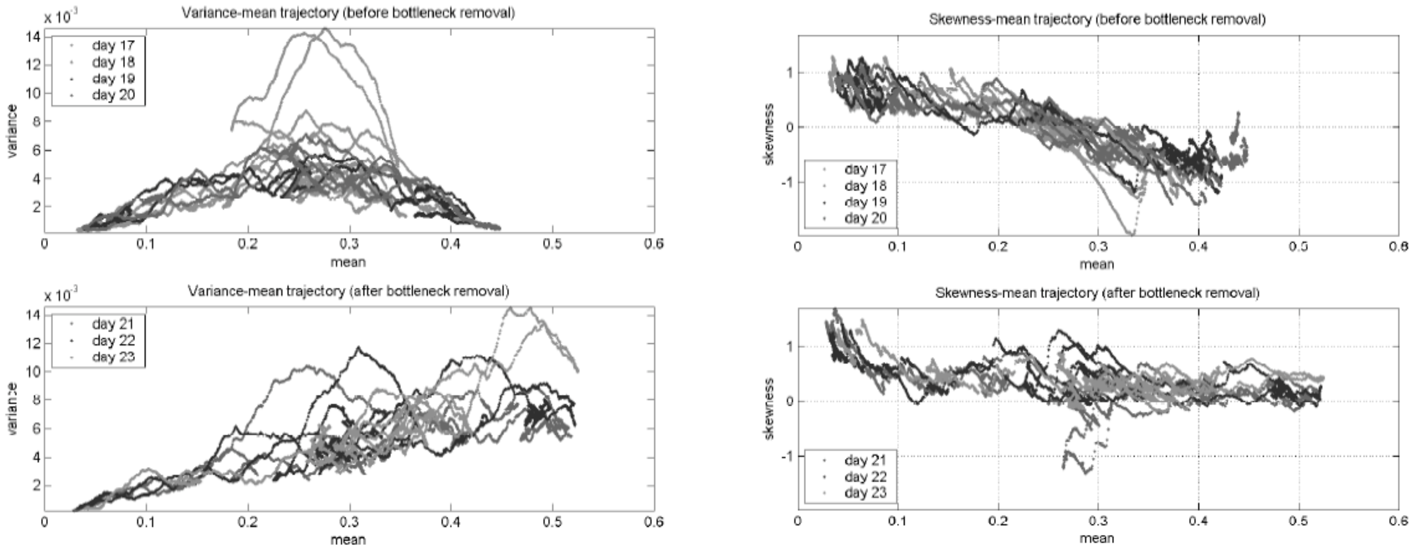
Idea: use trajectories of VAR-MEAN and SKEW-MEAN
to infer the presence of a bottleneck (and value of C)
without a priori information on C !





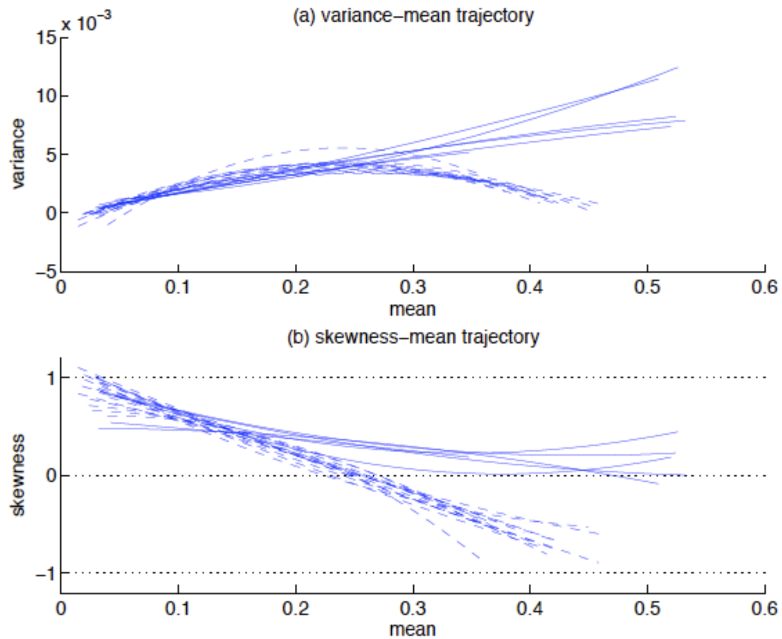
Time-scales

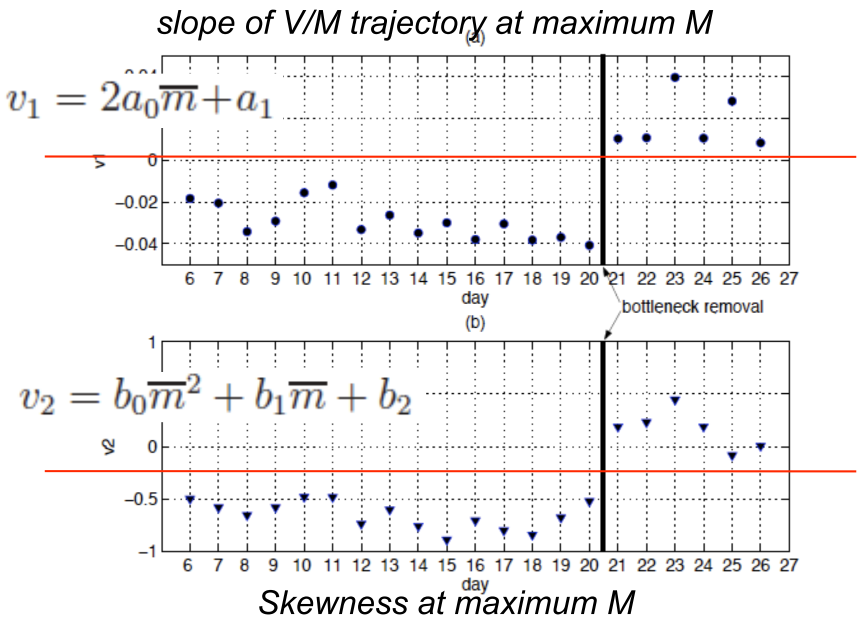
- Importance of time-scales
  - aggregate rate R(t) is measured in time-bins of length **T**
  - moments (MEAN, VAR, SKEW) are estimated in window of length **W**
  - time-scales constraints
    - T >= a few RTTs (to close the CC loop)
    - W >> T (to have enough samples)
      but small enough to ensure stationarity (time-of-day fluctuations)
    - our setting: T=10 sec, W=1 hour.



Data reduction

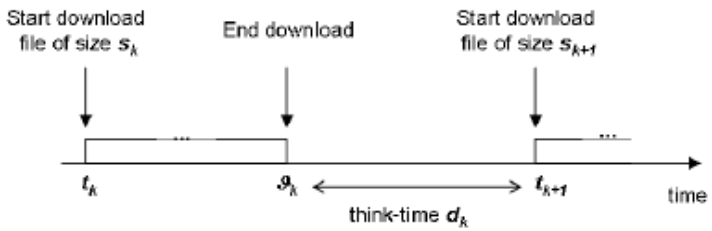
- Data reduction
  - Our goal is only to discriminate increase/decrease trends
  - apply quadratic regression on V/M, S/M data
  - benefits: reduces noise, impact of outliers → robustness
    - *don't use more data than you actually need!*
  - fitted polynomial parameters give synthetic indicators that can be thresholded to trigger alarm

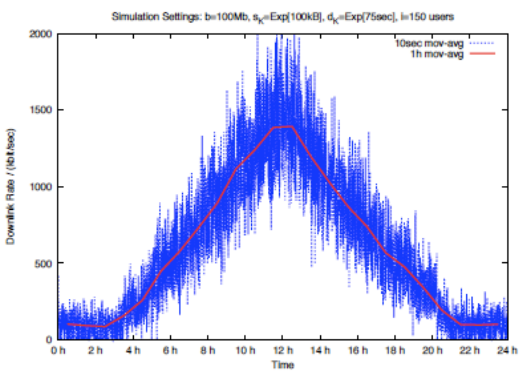


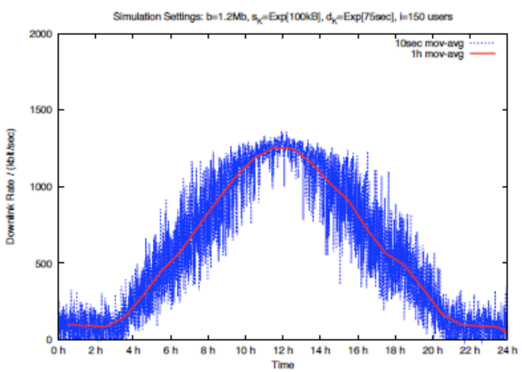


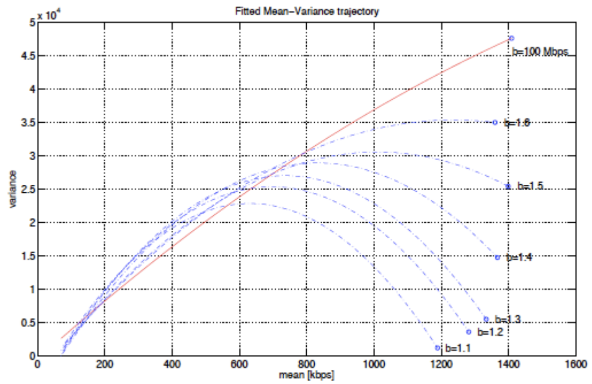
Validation by simulation

- Simulation set-up (ns-2)
  - simple closed-loop user model
  - allows testing different congestion levels, bottleneck types (buffering vs. discarding excess packets) than real bottleneck









Retransmission TimeOuts

- Bottleneck → Congestion *<u>at peak hour</u>* → more packet loss → more retransmissions due to timeout expirations
- Idea: infer congestion from Frequency of Retransmission Timeouts (RTO)s in some timebin T

$$F_{RTO}(T) = \frac{\#\text{ of RTO events in T}}{\#\text{ of DATA packets in T}}$$

- Expectation: bottleneck causes large increase of F_{RTO} in peak-hour compared to off-peak periods

- Tool to infer TCP RTOs from DATA-ACK (mis)matching
  - classifies different types of RTOs (spurious, …)
  - measure frequency of RTOs in timebins of 1 min

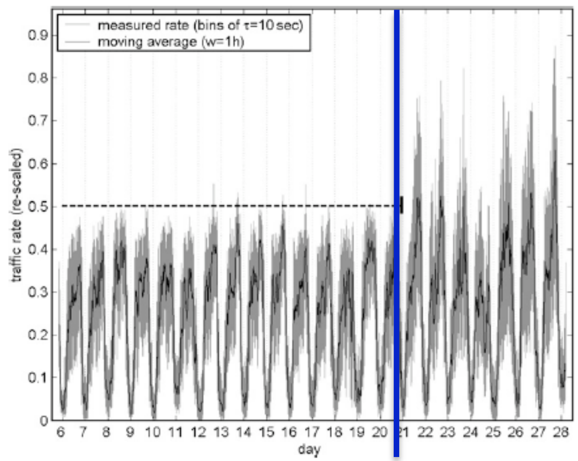
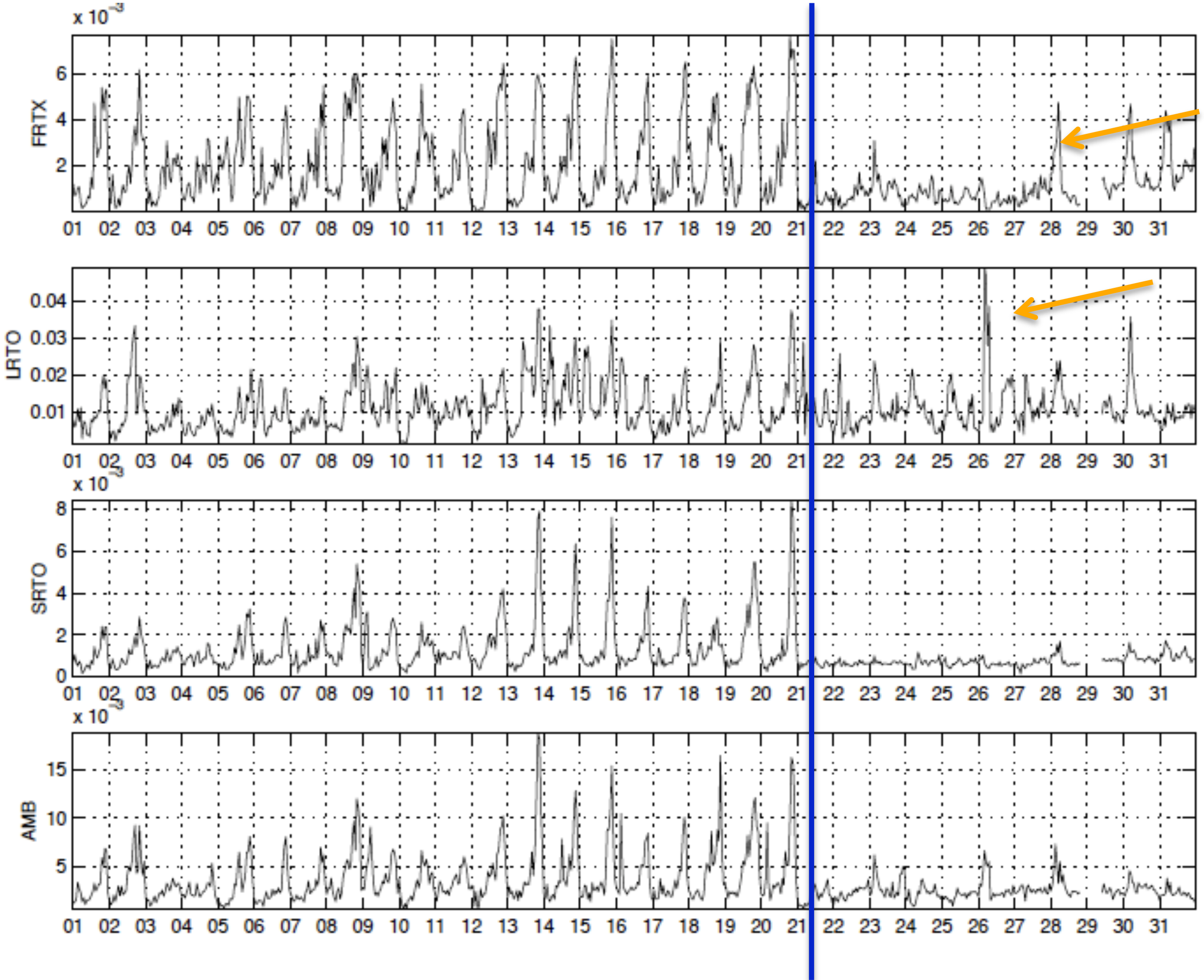


Fig. 2. SA total rate for the monitored period (10 s bins, rescaled values).

Raw RTO measurements

- On the right way, but “noise” would cause false alarms …



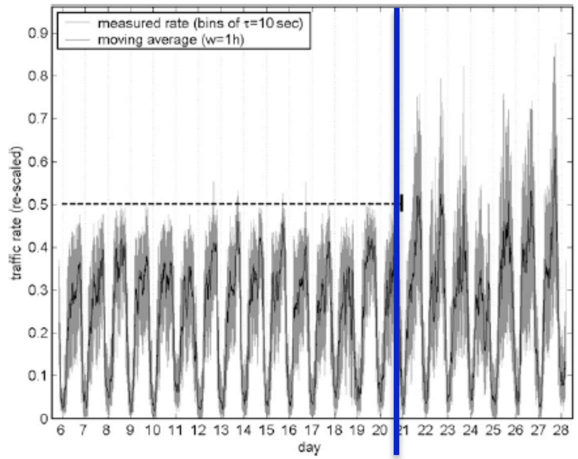
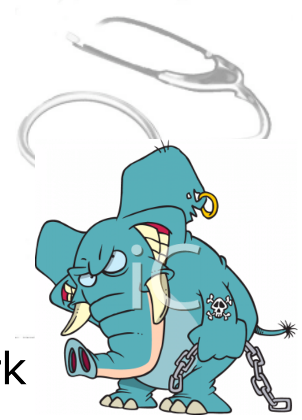
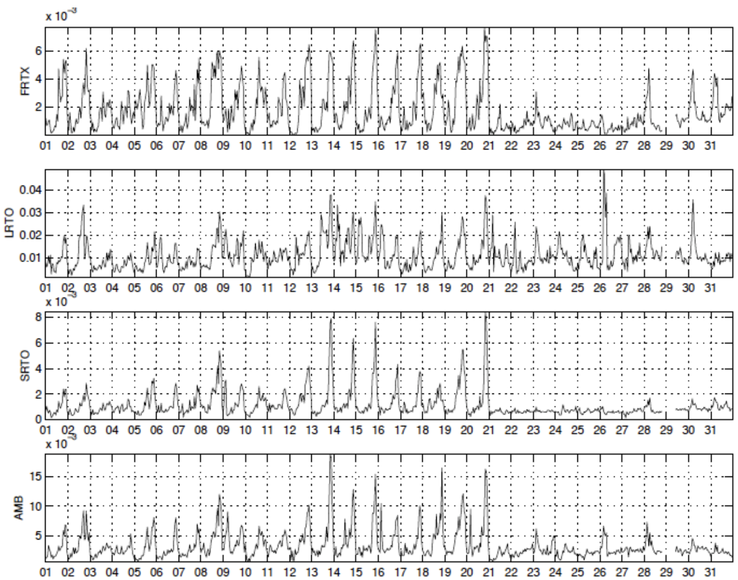


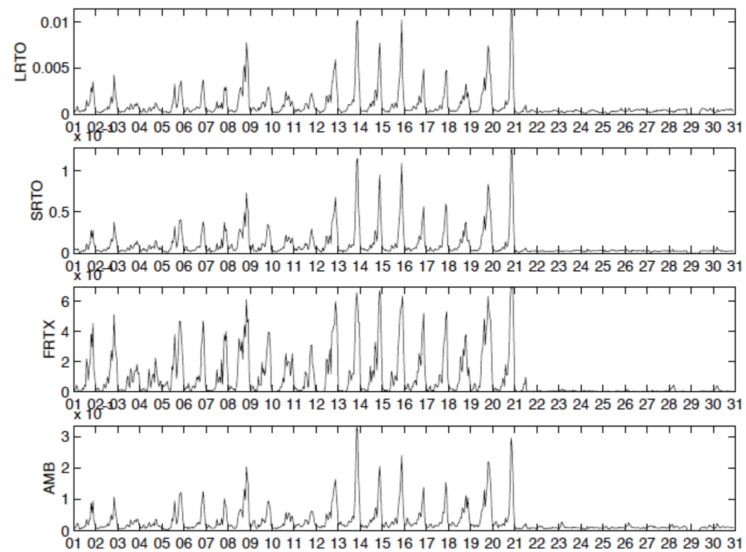
Fig. 2. SA total rate for the monitored period (10 s bins, rescaled values).

Filtered RTO measurements



- Noise due to heavy-hitters "bad elephants" (BE)
  - few clients, with high traffic volume (elephant) and poor network conditions ("bad") due to local causes
  - biasing the ratio $\frac{\#\text{of RTO events}}{\#\text{of DATA packets}}$ in some timebins
- Workaround
  - identify BE by some heuristic (e.g. the top-10 with most RTOs) and filter them away





Stepping into a more general problem…

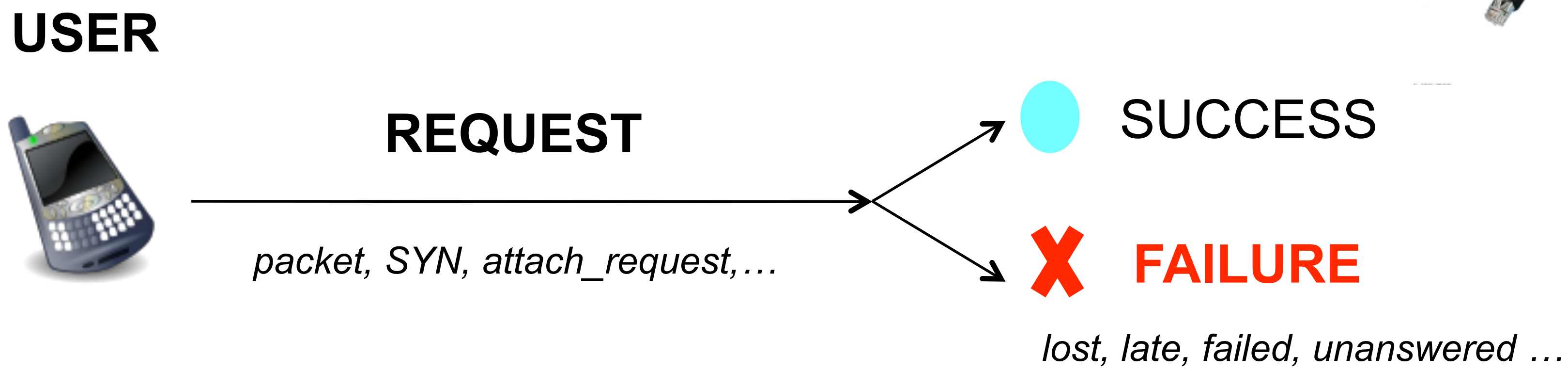
- The problem of “bad elephants” is more general, and is encountered often in the field of networking
  - traffic distributions are heavy-tailed
    - → there are always some “elephants” around
  - poor performance sometimes due to local conditions
    - e.g. congestion of dedicated resources, terminal errors …
    - → some elephants will be ‟ bad”
  - many common KPI are just global percentages!

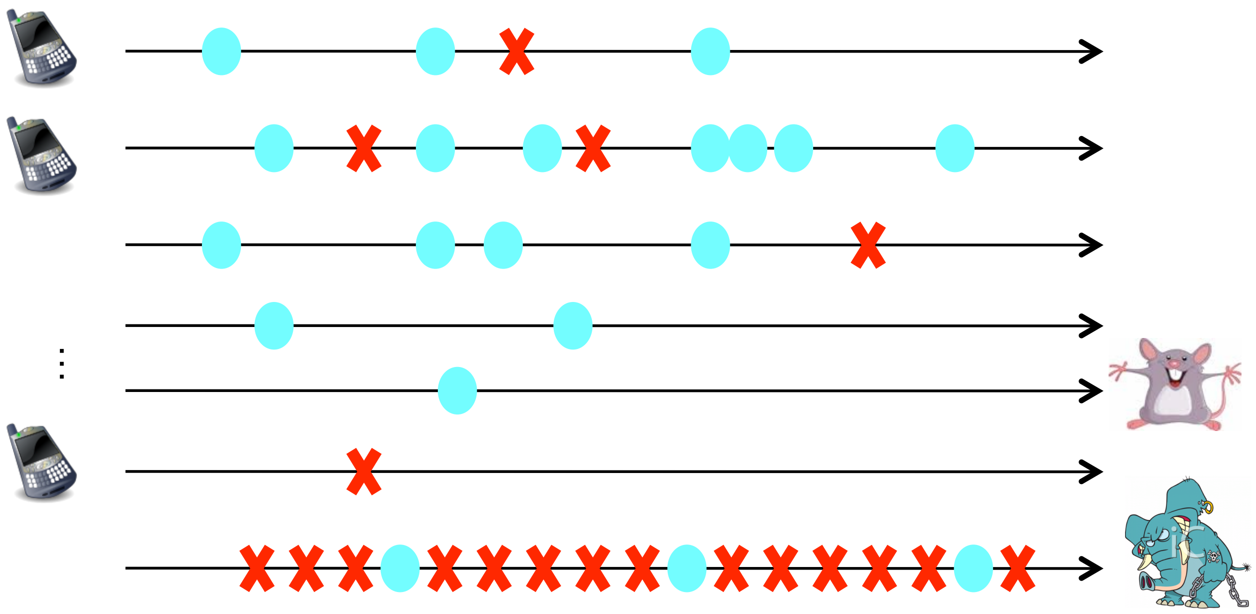
$$\mathrm{K}PI = \frac{\#\,\text{of failed attempts}}{\#\,\text{of attempts}}$$

- *Can we provide a theoretically-founded solution (in place of heuristics) ?*

A. Coluccia, F. Ricciato, P. Romirer, On Robust Estimation of Network-wide Packet Loss in 3G Cellular Networks, Proc. of 5th IEEE Broadband Wireless Access Workshop (BWA'09).

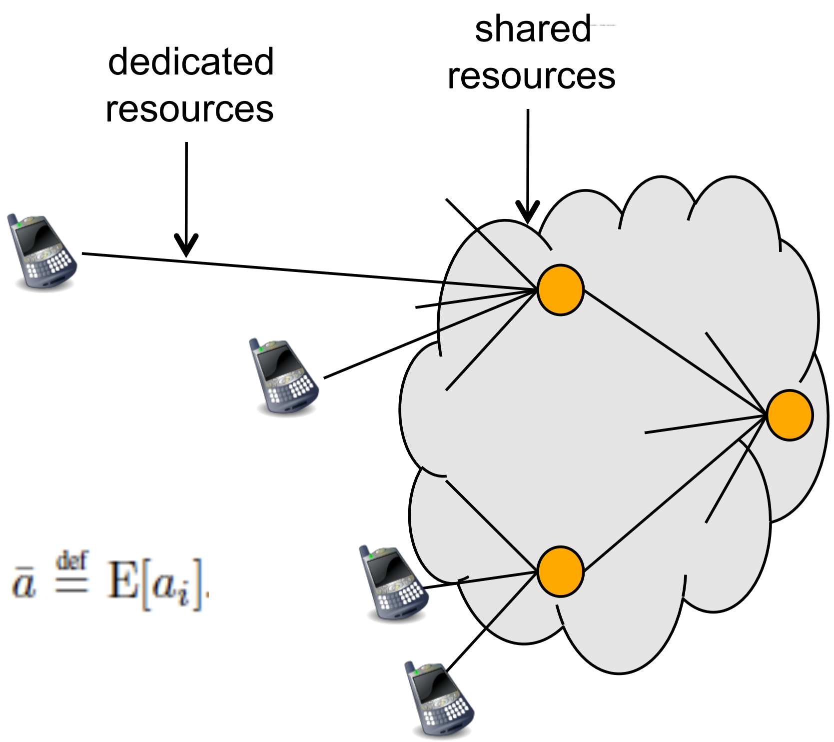
System Modeling





System Modeling

- Notation
  - in a generic timebin t user i ($i=1...I$)
  - generates n_i "requests" (e.g. packets)
  - out of which m_i "fail" (e.g. lost)



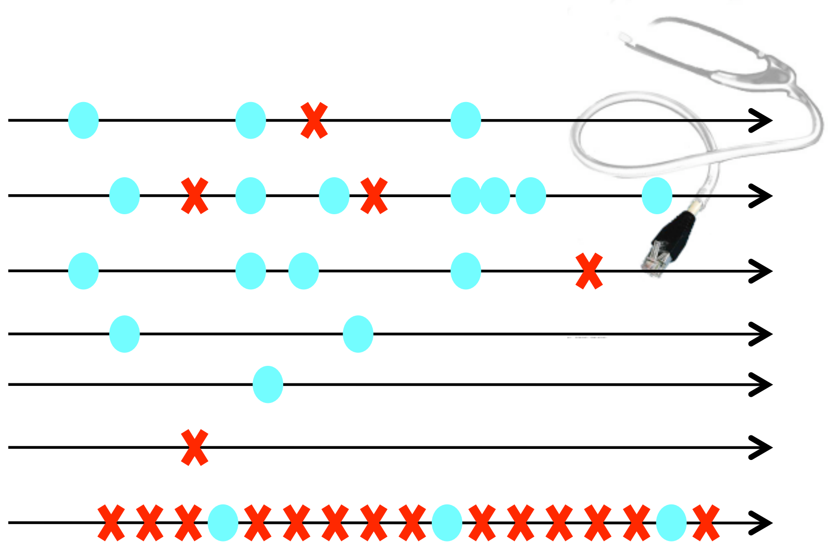
- Assumptions
  - failures are independent and occur with (unknown) probability a_i
  - a_i's are iid random variables with mean $\bar{a} \stackrel{\text{def}}{=} \mathrm{E}[a_i]$
  - independency between traffic volume n_i and failure prob. a_i

- Goal: estimate $\bar{a} \stackrel{\text{def}}{=} \mathrm{E}[a_i]$ given a set of measurements $\{n_i,m_i\}$

Estimators

- Goal: estimate $\bar{a} \stackrel{\text{def}}{=} \mathrm{E}[a_i]$ given $\{n_i, m_i\}$
- The Ideal Estimator
  - is <u>unbiased</u>
  - has <u>minimum variance</u>
  - is <u>simple</u>: fast to compute, easy to implement, easy to understand by practitioners!
  - is <u>general</u>: not bound to a specific (class of) traffic distribution (n_i's)

Basic Estimators



- Empirical Global Ratio
  - Global Percentage

$$EGR = \frac{\#\ \text{of total failures}}{\#\ \text{of total requests}} = \frac{\sum_i m_i}{\sum_i n_i}$$

- Empirical Mean Ratio
  - Arithmetic mean of individual ratios $r_i = \frac{m_i}{n_i}$

$$EMR = \frac{1}{I}\sum_i r_i = \frac{1}{I}\sum_i \frac{m_i}{n_i}$$

  - *NB: EMR is more costly to implement than EGR, requires per-user counters → need to extract event-to-user associations*

NB: $r_i = m_i / n_i$ is the minimum variance unbiased estimator (MVUE) for a_i

EWR - definition

- Both EGR and EMR can be "corrected"
  by filtering away very big (for EGR) or very small (for EMR) samples
  - discarding lots of data, especially for long-tailed n_i's

- Can we do something more clever than *discarding* data ?
  - *weighting* data !

- **Empirical Weighted Ratio (EWR)**

$$EWR = \sum_i w_i \frac{m_i}{n_i} = \sum_i w_i r_i = \underline{w}^T \underline{r} \qquad \text{with} \quad w_i > 0, \ \sum_i w_i = 1$$

- EGR,EMR are special cases of EWR
  - w_i *constant* → $w_i = 1/I$ → *EWR=EMR*
  - w_i *proportional to* n_i → $w_i = n_i/N$ → *EWR=EGR*

EWR - optimization

- Problem: Find the *optimal* weights w_i's that minimize the variance of the estimator VAR(S(**w**))

$$S(\mathbf{w}) = \mathbf{w}^T\mathbf{r} \quad \text{with} \quad |\mathbf{w}| = 1, \mathbf{w} \geq \mathbf{0}$$

- Resolution
  - compute variance of estimator as function of weights VAR(*S(**w**)*)=f(**w**)
  - constrained minimization, solve by Lagrangian multipliers

$$\dot{\mathbf{w}} = \underset{\substack{\mathbf{w}>0 \\ \sum_i w_i = 1}}{\arg\min} \, \mathrm{VAR}[S(\mathbf{w})]$$

- Exact optimal solution

$$\dot{w}_i = \frac{\dot{n}_i}{\sum_{j=1}^{I} \dot{n}_j} \quad \text{with} \quad \dot{n}_i \stackrel{\text{def}}{=} \frac{1}{\sigma_a^2 + \frac{(\bar{a} - \sigma_a^2 - \bar{a}^2)}{n_i}}$$

[BWA'09] A. Coluccia, F. Ricciato, Peter Romirer, On Robust Estimation of Network-wide Packet Loss in 3G Cellular Networks, IEEE BWA 2009, Honolulu, 30 November 2009

EWR – weight setting

piece-wise linear approximation (EPWR)

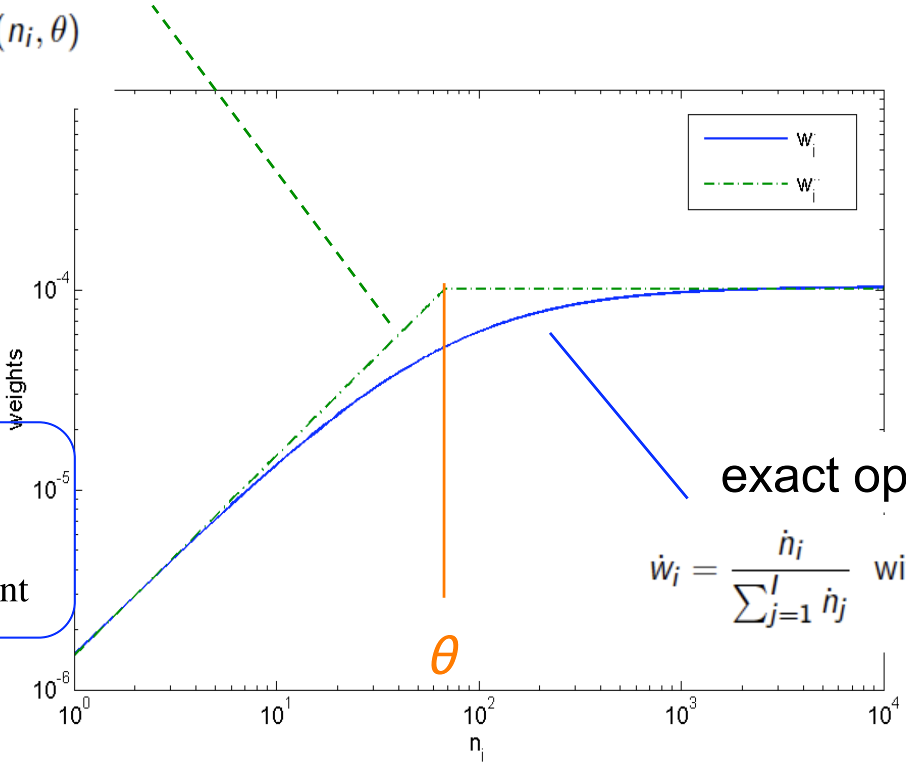
$$\ddot{w}_i = \frac{\ddot{n}_i}{\sum_{j=1}^{I} \ddot{n}_j} \qquad \ddot{n}_i \stackrel{\text{def}}{=} \min(n_i, \theta)$$

where $\dot{\theta} \stackrel{\text{def}}{=} \frac{\bar{a}-\sigma_a^2-\bar{a}^2}{\sigma_a^2}$

$n_i < \theta \rightarrow w_i = n_i \cdot c$

$n_i \geq \theta \rightarrow w_i = \theta \cdot c$

with c a normalization contant



exact optimal solution

$$\dot{w}_i = \frac{\dot{n}_i}{\sum_{j=1}^{I} \dot{n}_j} \quad \text{with} \quad \dot{n}_i \stackrel{\text{def}}{=} \frac{1}{\sigma_a^2 + \frac{(\bar{a}-\sigma_a^2-\bar{a}^2)}{n_i}}$$

- setting the knee-point θ
  - optimal value depends on first two moments of p(a): → unknown ☹
  - final estimator performance are weakly sensitive to exact location of knee-point, as far as "extreme" settings (very low, very high) are avoided
  - simplest solution: set to fixed value, e.g. θ=20.

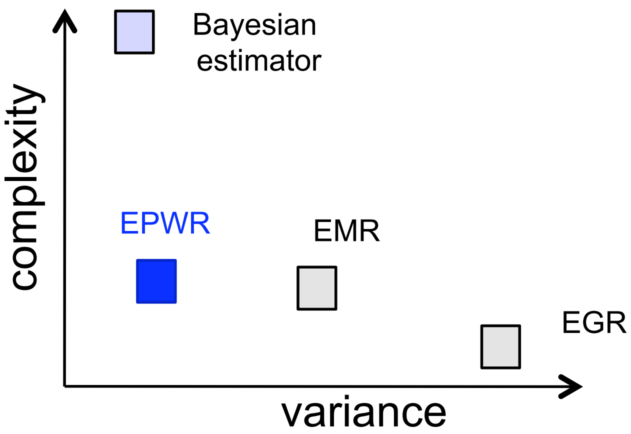
Bayesian Estimators

- Empirical Piece-wise Linearly Weighted Ratio (EPWR)
  - single parameter θ (set heuristically to $\theta=20$)
  - very simple conceptually and computationally
  - requires individual per-user counters, as EMR

$$S(\mathbf{w}) \stackrel{\text{def}}{=} \sum_{i=1}^{I} w_i r_i = \mathbf{w}^T \mathbf{r}$$

$$\ddot{w}_i = \frac{\ddot{n}_i}{\sum_{j=1}^{I} \ddot{n}_j} \qquad \ddot{n}_i \stackrel{\text{def}}{=} \min(n_i, \theta)$$

- Alternative approach: Bayesian estimators
  - Bayesian hierarchical model: $p(\mathbf{a}) \rightarrow \{a_i\} \rightarrow \{m_i\}$
  - Approach: empirical parametric Bayes + conjugate prior (*)
  - elegant maths, closed formula, but in practice same performance as EPWR



(*) Fabio Ricciato, Angelo Coluccia, and Peter Romirer, Bayesian Estimation of Network-wide Mean Failure Probability in 3G Cellular Networks, Proc. of PERFORM 2010 Workshop, Vienna, 14-16 October 2010, LNCS vol. 6821.

Results DATA:INV

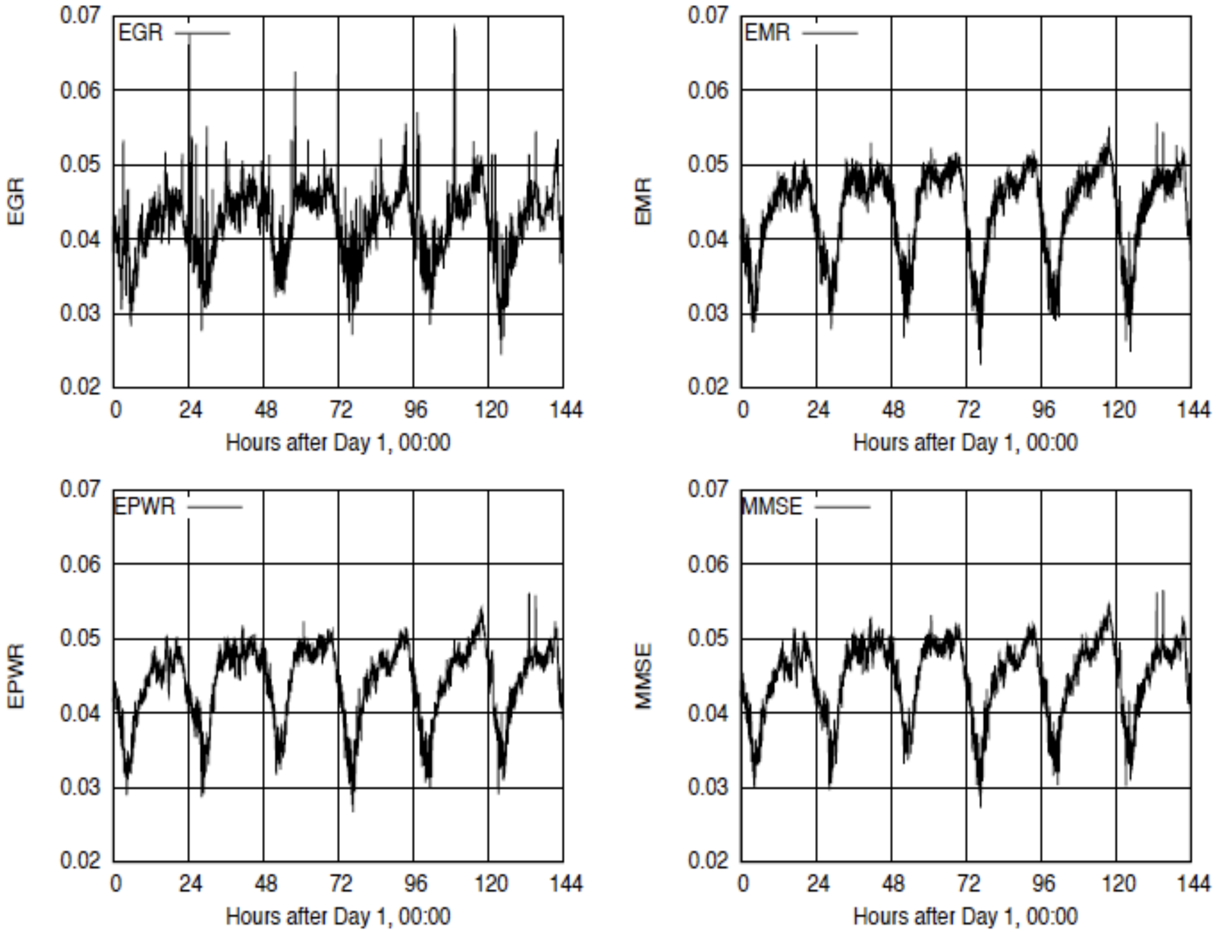


Fig. 2. Estimated mean failure probability for DATA:INV dataset (missing or ambiguous SYNACK/ACK associations).

- 54 -

Results DATA:RTT

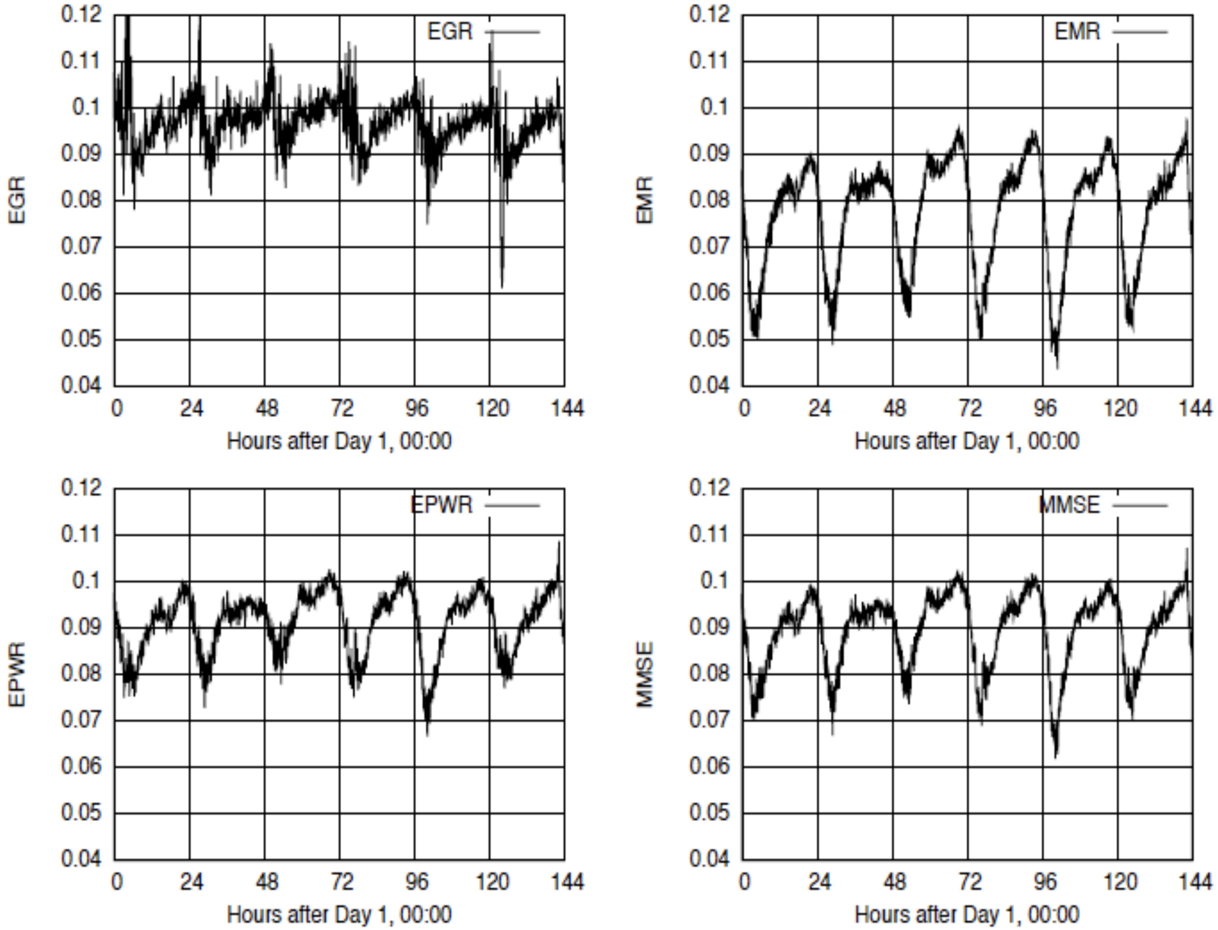
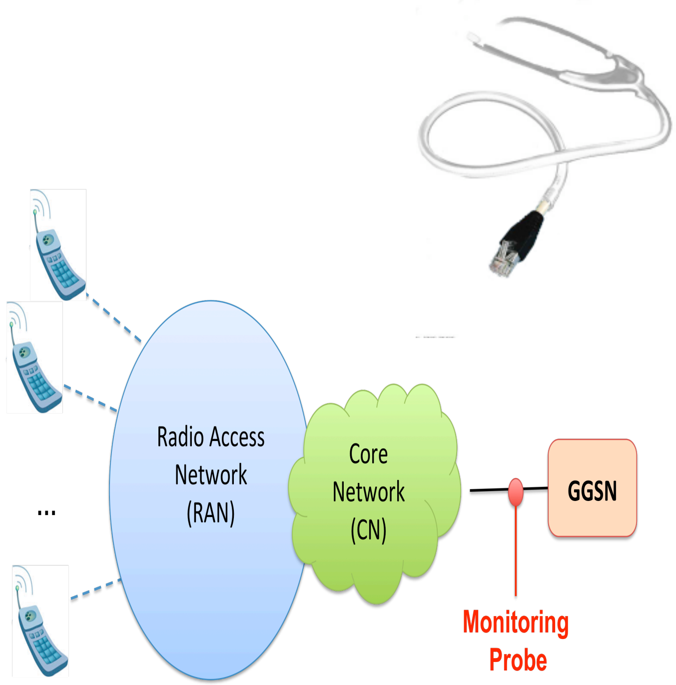


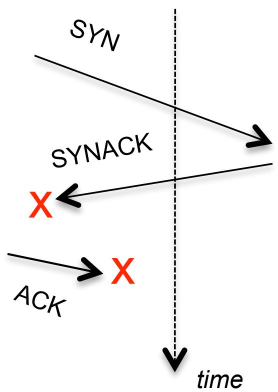
Fig. 3. Estimated mean failure probability for DATA:RTT dataset (unambiguous SYNACK/ACK pairs with semi-RTT exceeding 500 ms).

Results from a real dataset

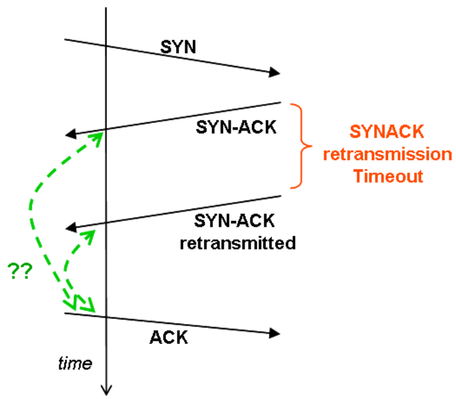
- Datasets from on a real UMTS/HSPA network
- DATA:INV
  - REQUEST := every SYNACK in DL
    SUCCESS := unambiguous ACK in UL
- DATA:RTT
  - REQUEST := unambiguous SYNACK/ACK pair
  - SUCCESS := RTT < 500 ms

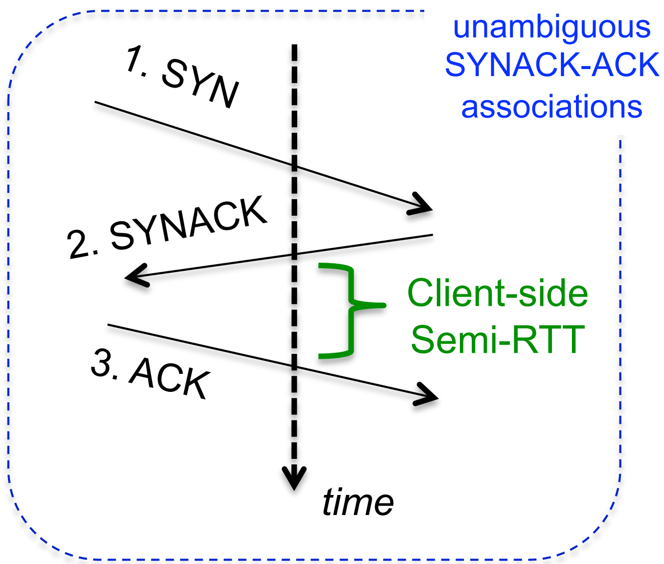


missing ACK



ambiguous association

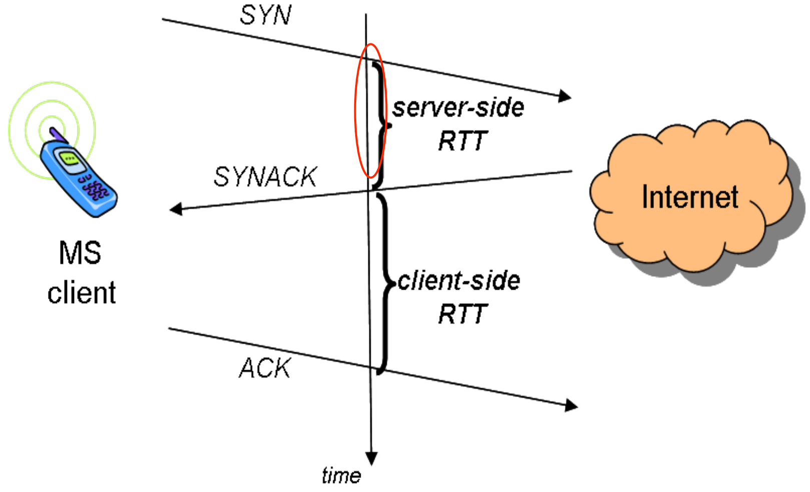


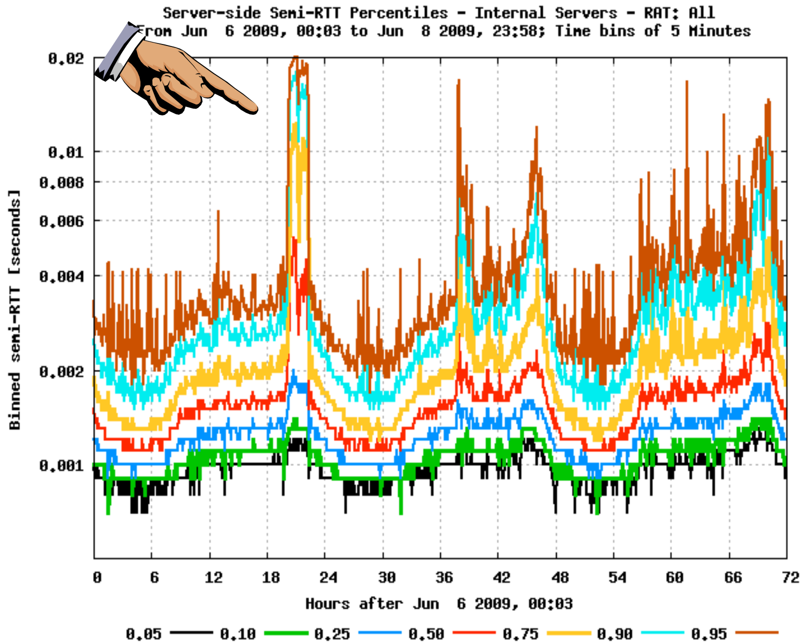


- 56 -

Detecting congestion bottleneck

- Detecting *upstream* congestion from server-side RTT
  - SYN-SYNACK associations



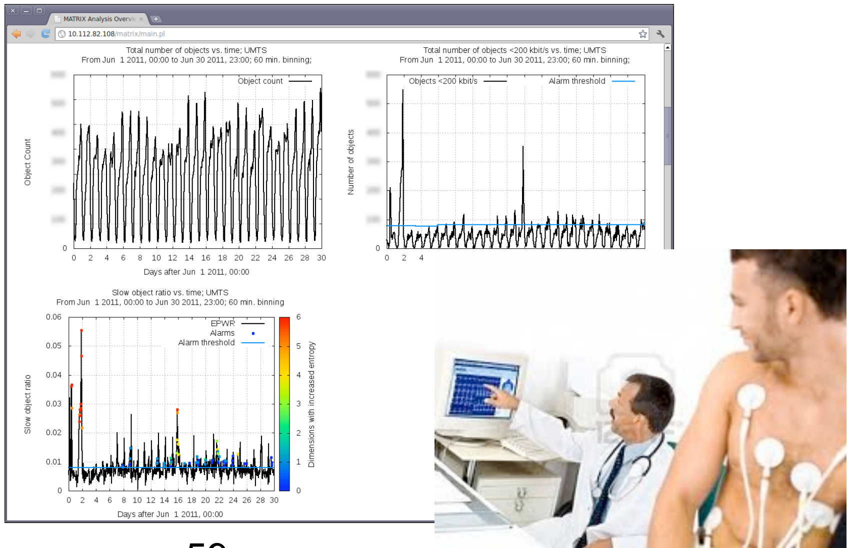


Next step

- The proposed approaches (rate/RTO/RTT) helped to detect instances of serious congestion in the real network
- Next goal: detecting *pre*-congestion events
  - 3G network capacity is becoming the bottleneck
- Approach: analyse per-flow throughput
  - extract relevant “signal” from per-flow throughput measurements
  - ongoing work ….

Working with online data

- Doctors need to practice on real patients …
- It's important to analyse recent data → online analysis
  - external information needed for drill-down is still available
    - *you can still talk to the patient*
  - timely identification of real anomalies has immediate impact
  - makes research more interesting, but also more costly
    - *need cooperation with the patient*



Wrap-up on Lessons Learned

- Research on 3G Traffic Monitoring …

- is interesting
  - as any research on real systems
- is useful
  - 3G network systems are too complex/large to be error-free
- is costly
  - data collection eats lot of engineer works
  - analysis and explorations of real data is often lengthy
  - every analysis task requires own methodology
  - need domain-specific knowledge about 3G networks, protocols

- lot of space for problem-driven & curiosity-driven research

Wrap-up on Lessons Learned

- Automatic anomaly-detection like a medical tool

- **Yes, it empowers the doctor**
- **No, it cannot replace the doctor**

- About network problems
  - Recurring problems: can be detected and diagnosed automatically (but also prevented…)
  - Novel problems: symptoms can be detected automatically, but need doctor for interpretation and drill-down

  - *healthy network = no recurring problems, only new ones*

Follow-up

- Publications:
- http://userver.ftw.at/~ricciato/publications.html

- email:
- ricciato@ftw.at