GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

AutoMon

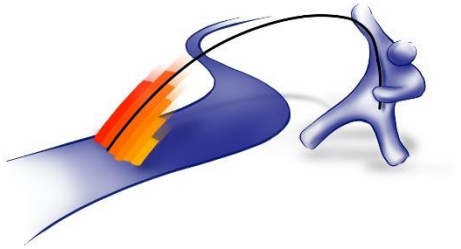
Indirect passive measurement of network characteristics in the AutoMon project

17. July 2017

IRTF NMRG

isarNet
Software Solutions GmbH

sebastian.meier@isarnet.de
jochen.koegel@isarnet.de



Agenda

- AutoMon project
- Project vision of measurement control
- Problem statement: unobserved parts
- Approach: passive sampled measurements
- First results
- Conclusion and outlook

AutoMon Project – Facts

Project goal: Automated performance monitoring

Funded by the German government

- Innovation program for Small and Medium Enterprises (SME) „KMU-innovativ“
- Volume: 2.69 M€

Time frame: June 2016 … May 2019

https://automon-projekt.de/en

AutoMon Project – Partners

Application partners

DB Systel: Service provider

For German railway, global logistics

Problem statements, use cases, scenarios, labs

Suitable solutions, concepts and ideas for future plans

MultiNetwork WAN services
For airlines, global enterprises,…

Research partners

Technical University Munich

Chair of Network Architectures and Services, Prof. Carle

SME in Munich

- IsarFlow network monitoring
- Network consulting

SME in Dresden

- exply.io (data exploration)
- Contributor to Neos CMS

AutoMon – Problem Statement

Challenges in network monitoring

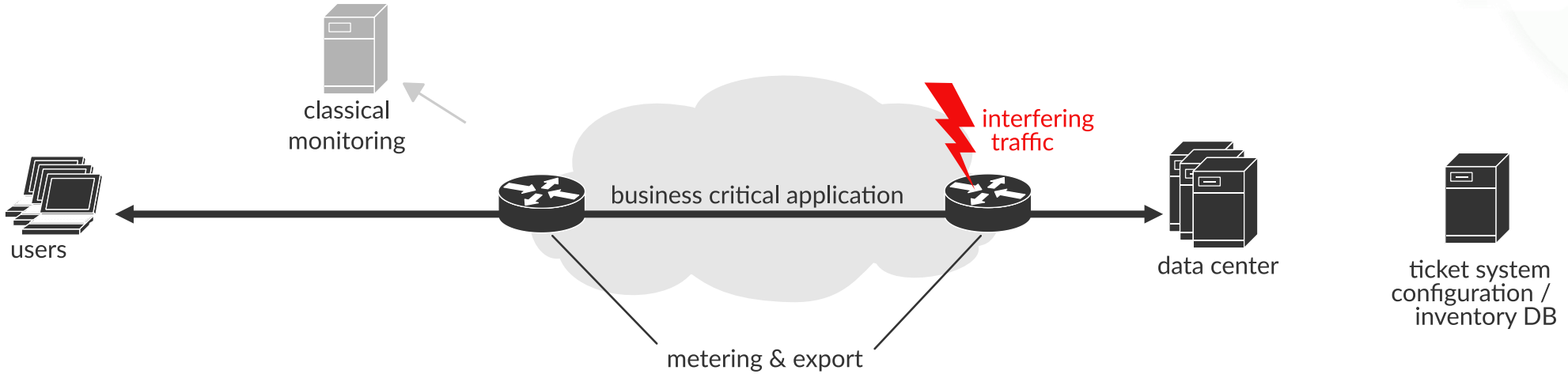
- network infrastructure becomes even more business critical
- fewer and fewer people operate increasingly large networks
- high dynamic in networks due to softwarerization and automation

➔Automation of network monitoring mandatory

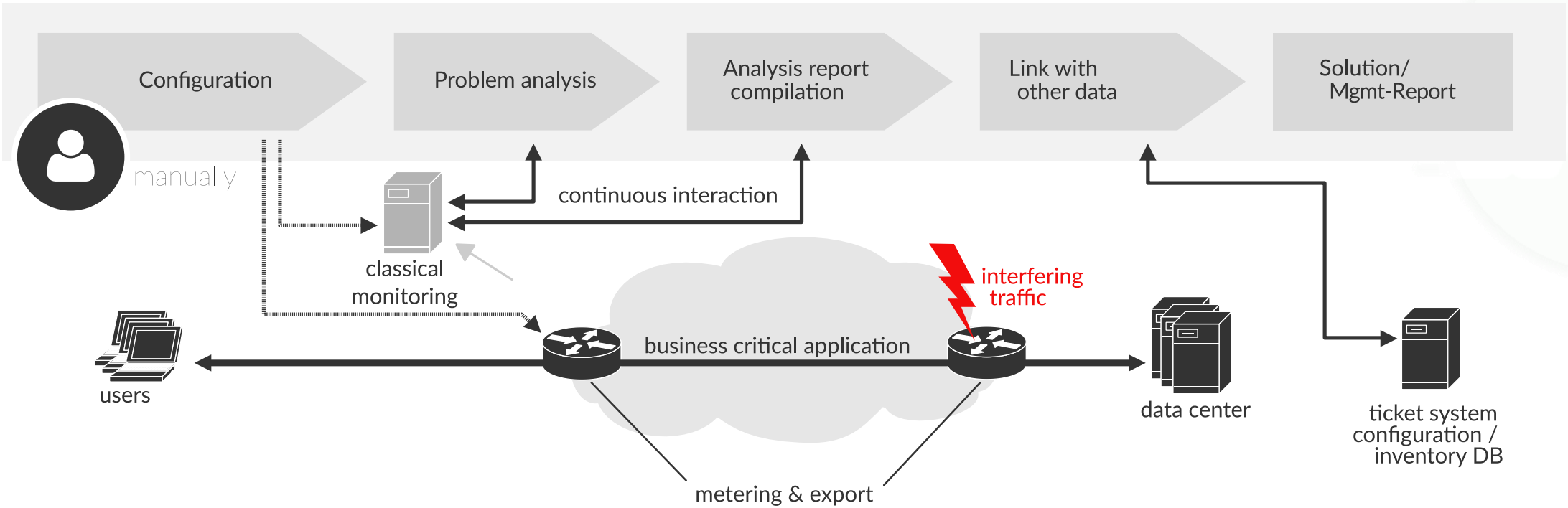
➔Continuous discussion: also automatically reconfigure network in case of problems?

AutoMon Vision

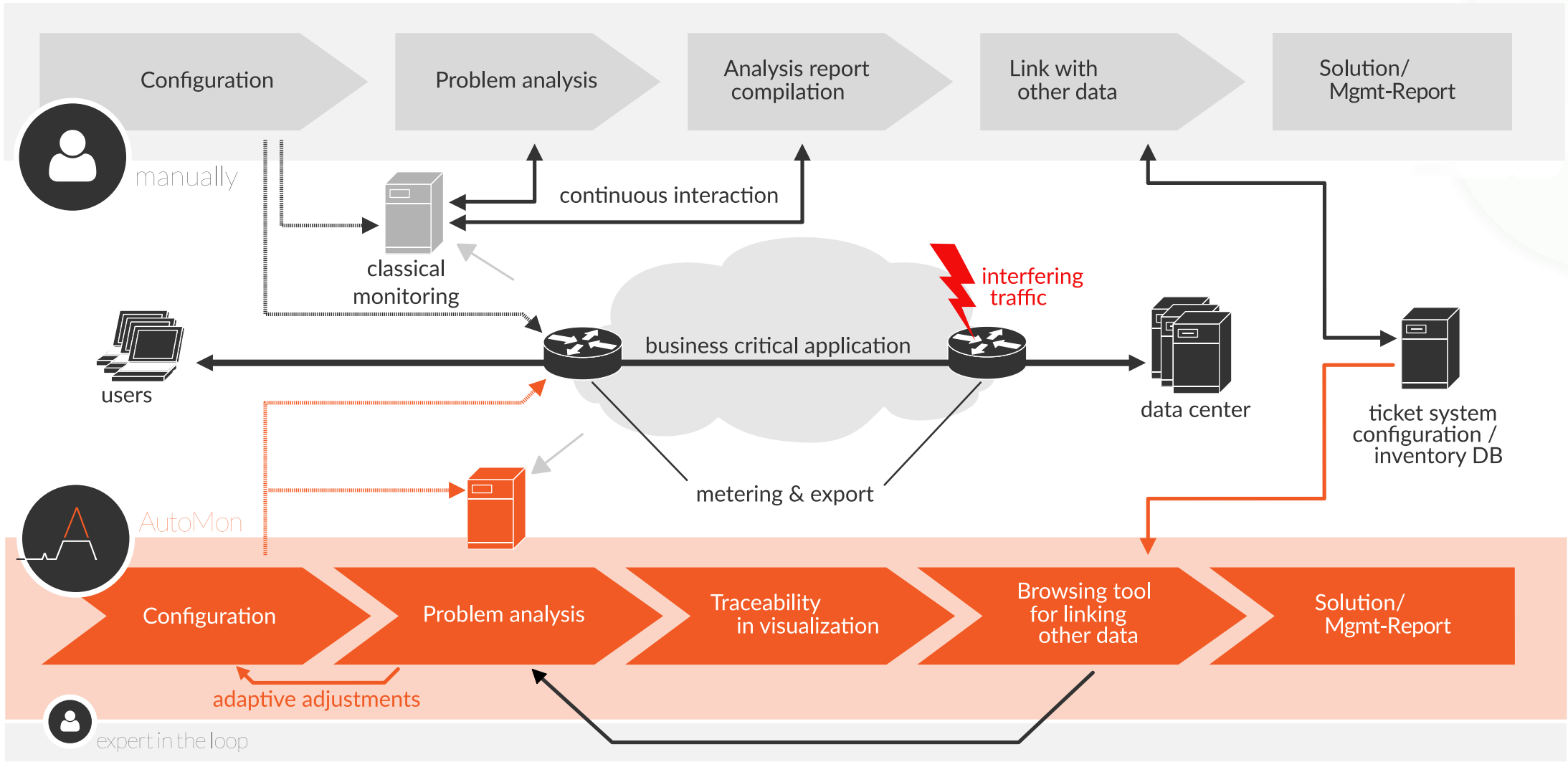
AutoMon Vision



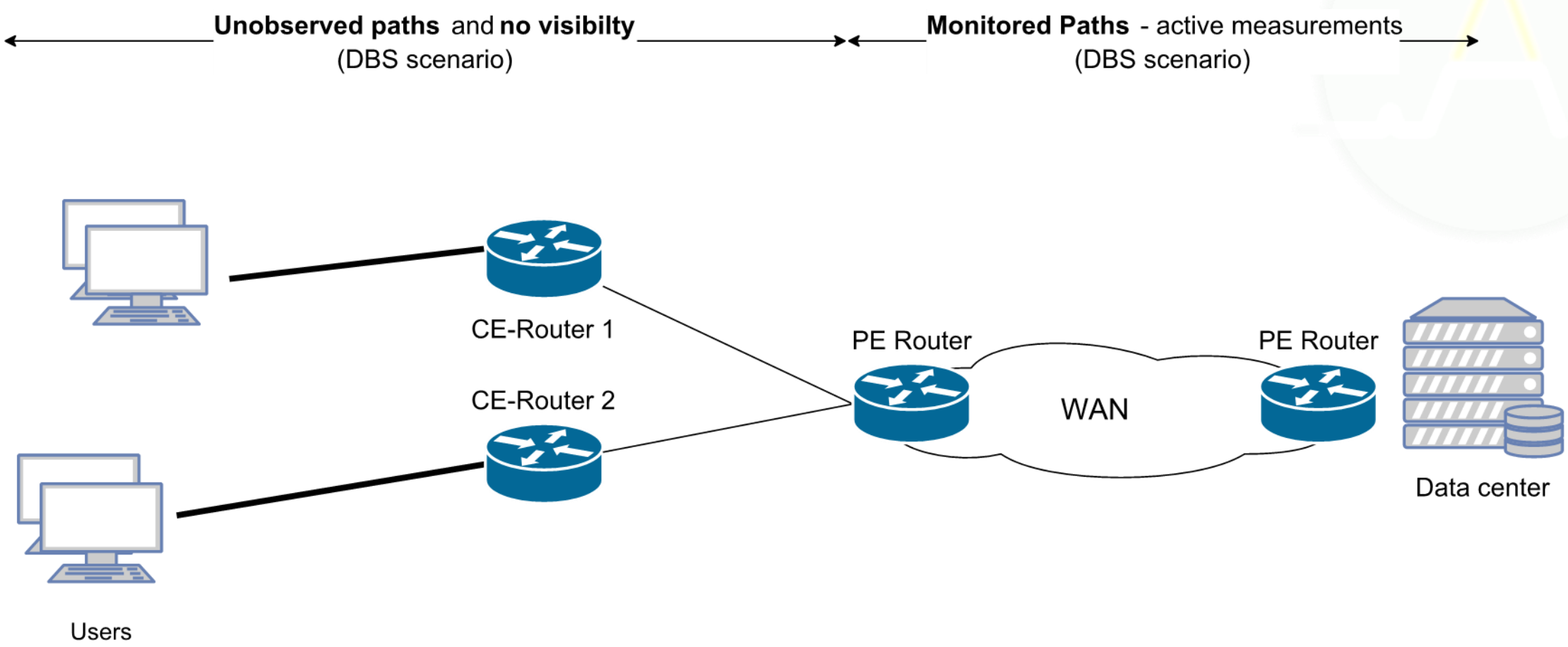
AutoMon Vision



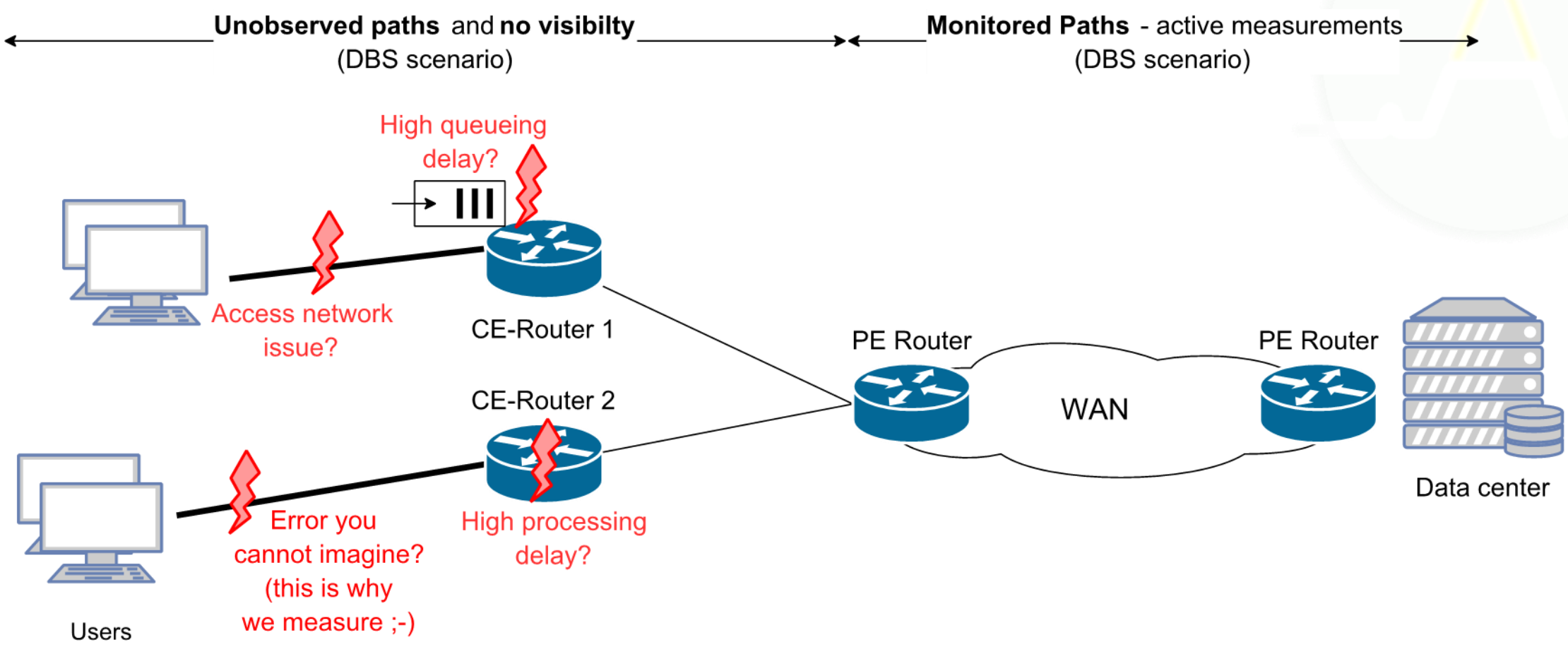
AutoMon Vision



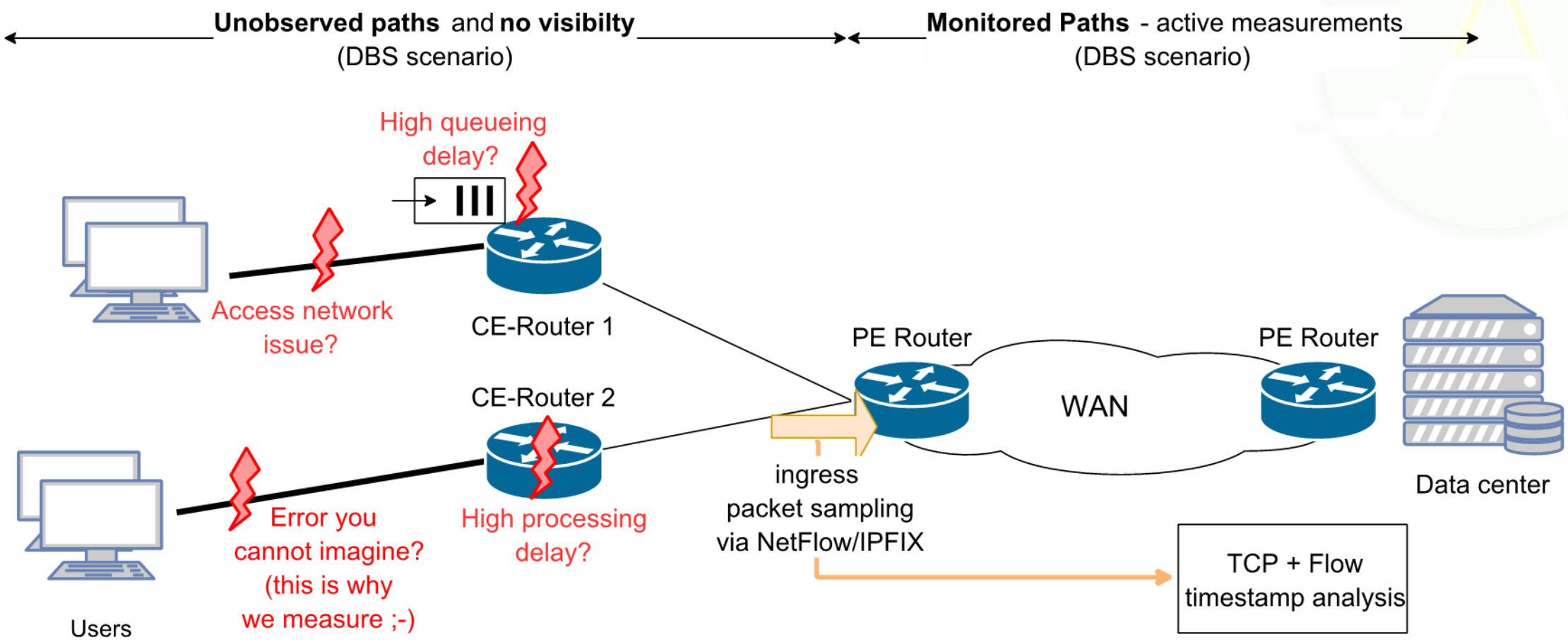
Down to earth – unobserved paths



Down to earth – unobserved paths



Down to earth – unobserved paths



Idea born while discussing skew-based sibbling detection [1]

Down to earth - problem statement

Focus: Larger scale delay variations

- not only packet-to-packet jitter (impacts Voice)
- but: generally worsening network conditions
  - impact interactive business applications
  - absolute delay values not required in the first place

- possible actions
  - bad condition: Trigger further automated investigation
  - good condition: Application performance issue ?
    →“Everything is fine in WAN – check DC“

Exemplary RTTs [2]

Research Question

How well can we passively measure jitter / delay increases?

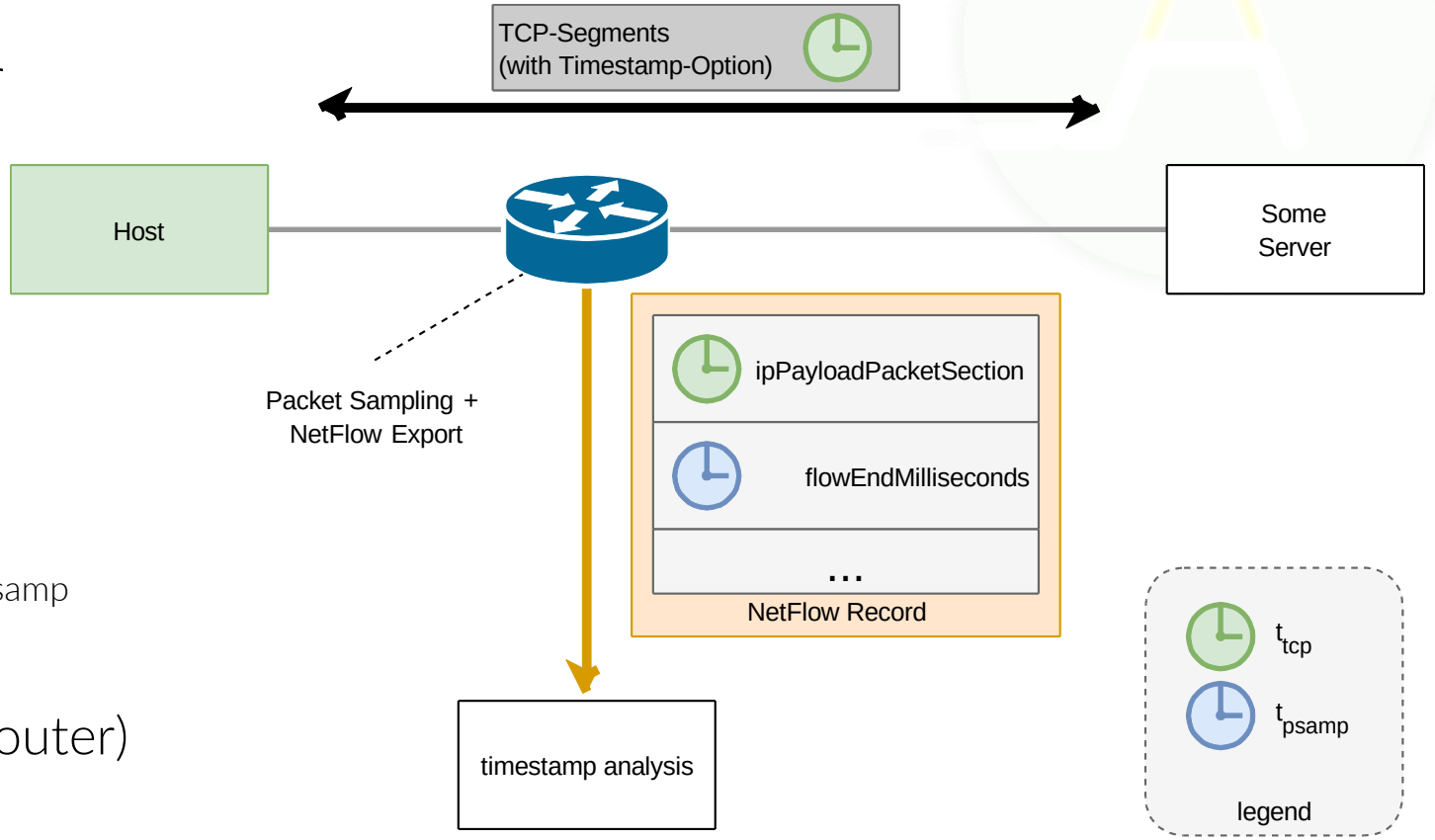
Timestamp sampling

Approach

- enable IP payload sampling on router
- export packet samples via NetFlow
- export two timestamps per packet sample
  - TCP timestamp (t_{tcp})
  - sampling timestamp (t_{psamp})
- establish relation between t_{tcp} and t_{psamp}

Challenges

- clock / timestamp accuracy (host & router)
- TCP timestamp availability
- suitable (per flow) sample size



Timestamp relation

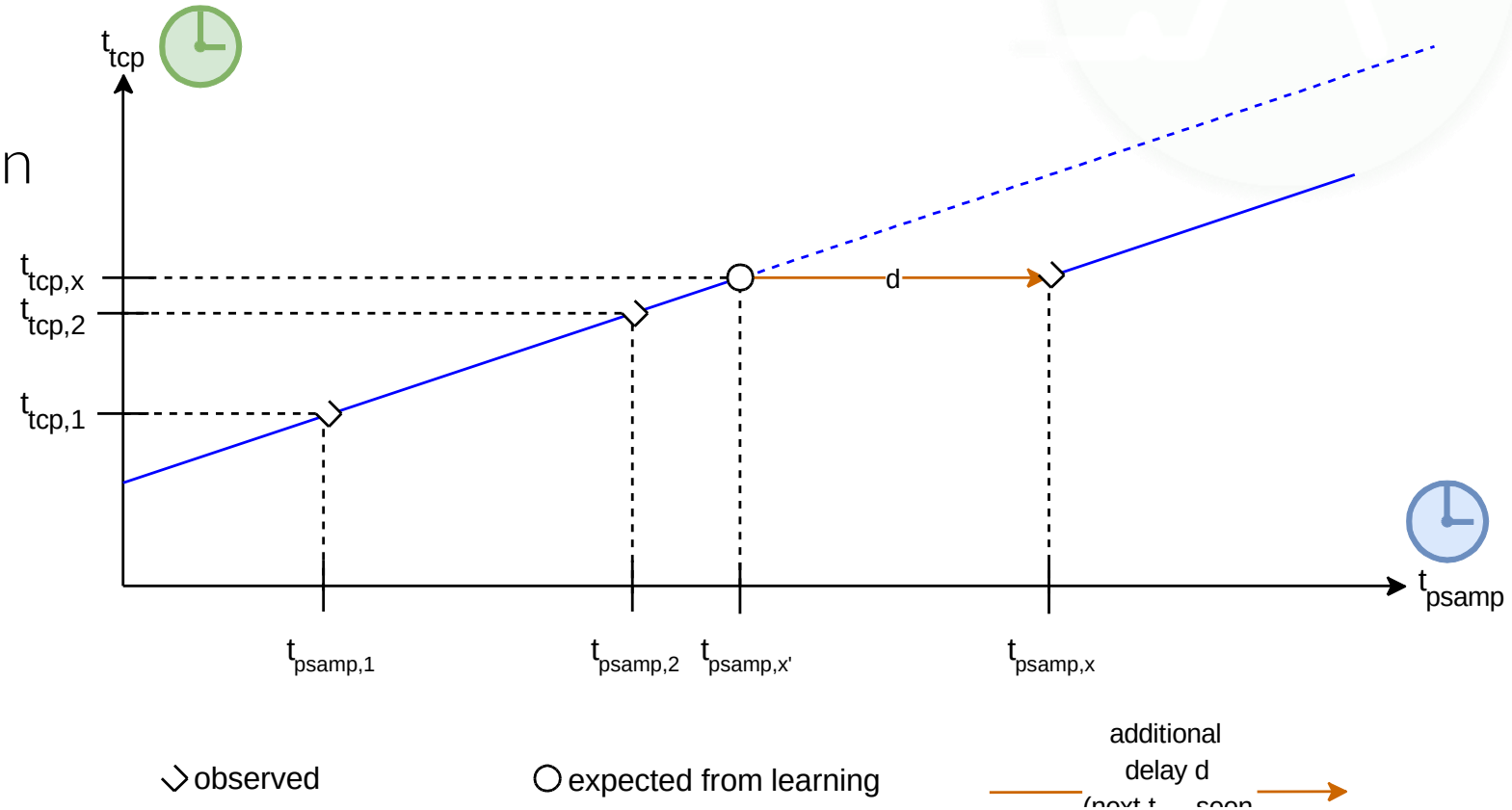
Assumptions

- clock drift negligible
- clocks do not jump

→ linear relation between t_{tcp} and t_{psamp}

Linear Algebra

- $y = m * x + b$
- $t_{tcp} = m * t_{psamp} + b$



Estimation of slope *m*

Slope

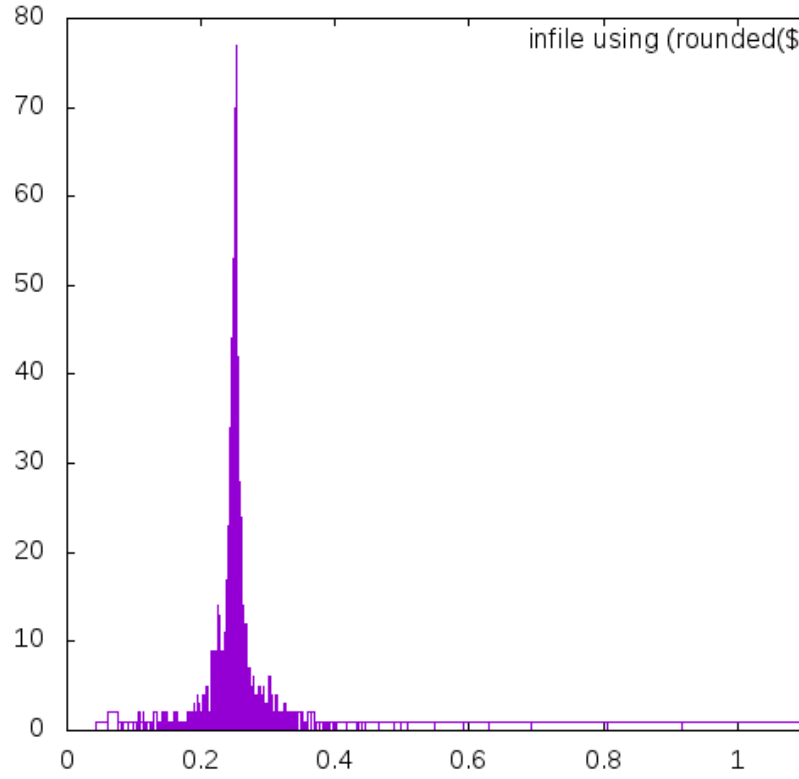
how fast advances time in router compared to time in host

Approach

- consider consecutive samples of same TCP flow
- for each pair: estimate slope m:

$$m = \frac{\Delta t_{tcp}}{\Delta t_{psamp}}$$

- „guess“ most likely slope after *n* slope estimations



Result

approach seems feasible (at least for lab setup)

Estimation of slope *m*

Slope

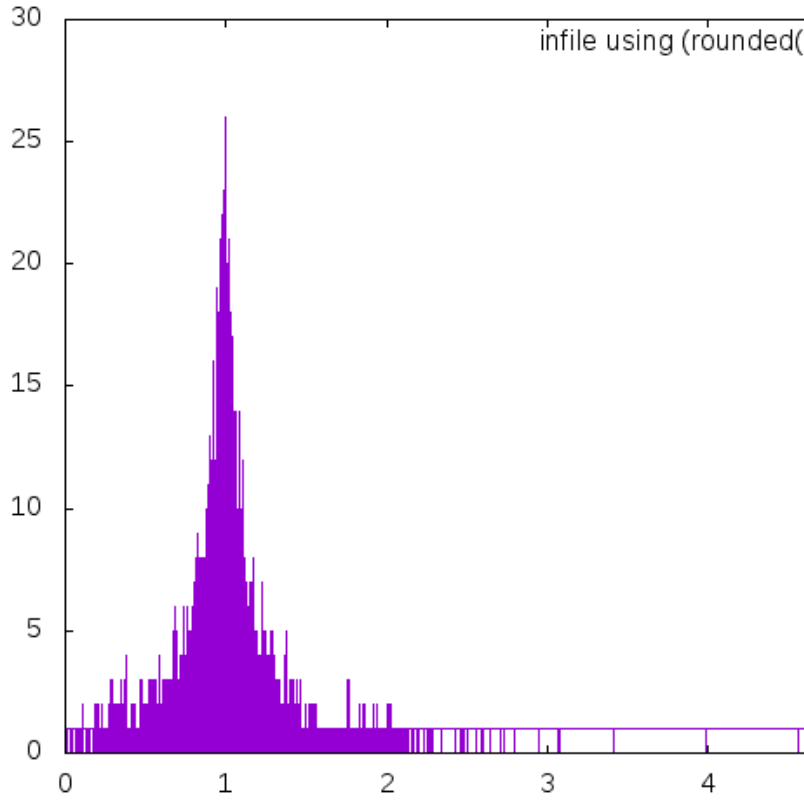
how fast advances time in router compared to time in host

Approach

- consider consecutive samples of same TCP flow
- for each pair: estimate slope m:

$$m = \frac{\Delta t_{tcp}}{\Delta t_{psamp}}$$

- „guess“ most likely slope after *n* slope estimations



Result

approach seems feasible (at least for lab setup)

Estimation of offset *b*

Offset

(constant?) difference between t_{tcp} and t_{pcap} timestamp values

Approach

1. calculate initial offset *b* with first **obs**erved packet sample

$$b = t_{tcp,obs,1} - m * t_{psamp,obs,1}$$

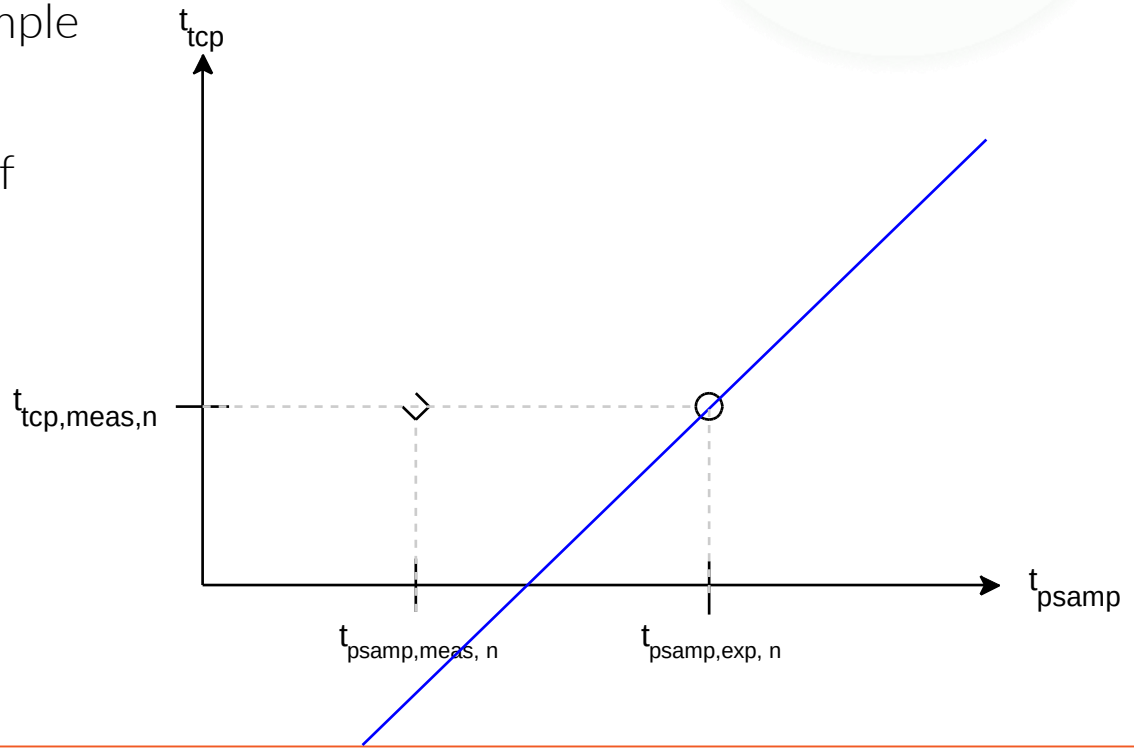
2. use initial offset for calculating **exp**ected timestamp of next sample

$$t_{psamp,exp,2} = \frac{t_{tcp,obs,2} - b}{m}$$

3. update *b* if $t_{psamp,exp,2} > t_{psamp,obs,2}$
4. repeat calculations for some/all subsequent samples to determine minimum/maximum offset

Open Issue

examine convergence behavior of offset



Estimation of offset *b*

Offset

(constant?) difference between t_{tcp} and t_{pcap} timestamp values

Approach

1. calculate initial offset *b* with first observed packet sample

$$b = t_{tcp,obs,1} - m * t_{psamp,obs,1}$$

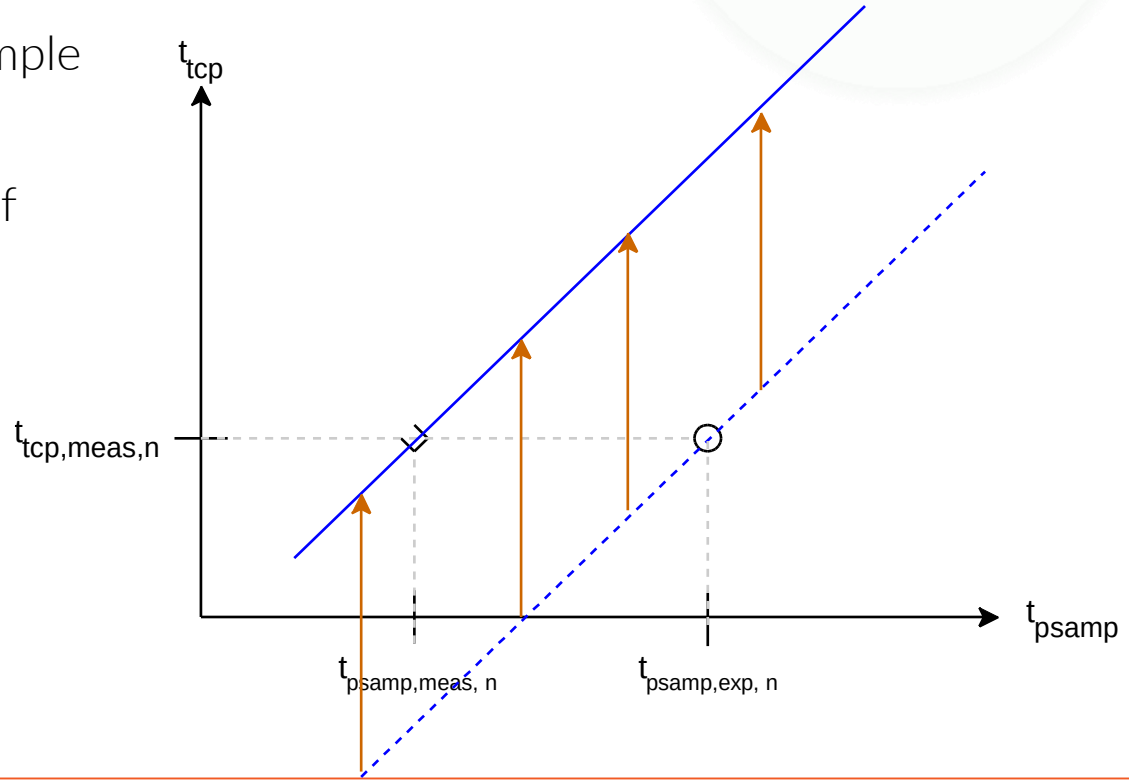
2. use initial offset for calculating expected timestamp of next sample

$$t_{psamp,exp,2} = \frac{t_{tcp,obs,2} - b}{m}$$

3. update *b* if $t_{psamp,exp,2} > t_{psamp,obs,2}$
4. repeat calculations for some/all subsequent samples to determine minimum/maximum offset

Open Issue

examine convergence behavior of offset



Estimation of offset *b*

Offset

(constant?) difference between t_{tcp} and t_{pcap} timestamp values

Approach

1. calculate initial offset *b* with first observed packet sample

$$b = t_{tcp,obs,1} - m * t_{psamp,obs,1}$$

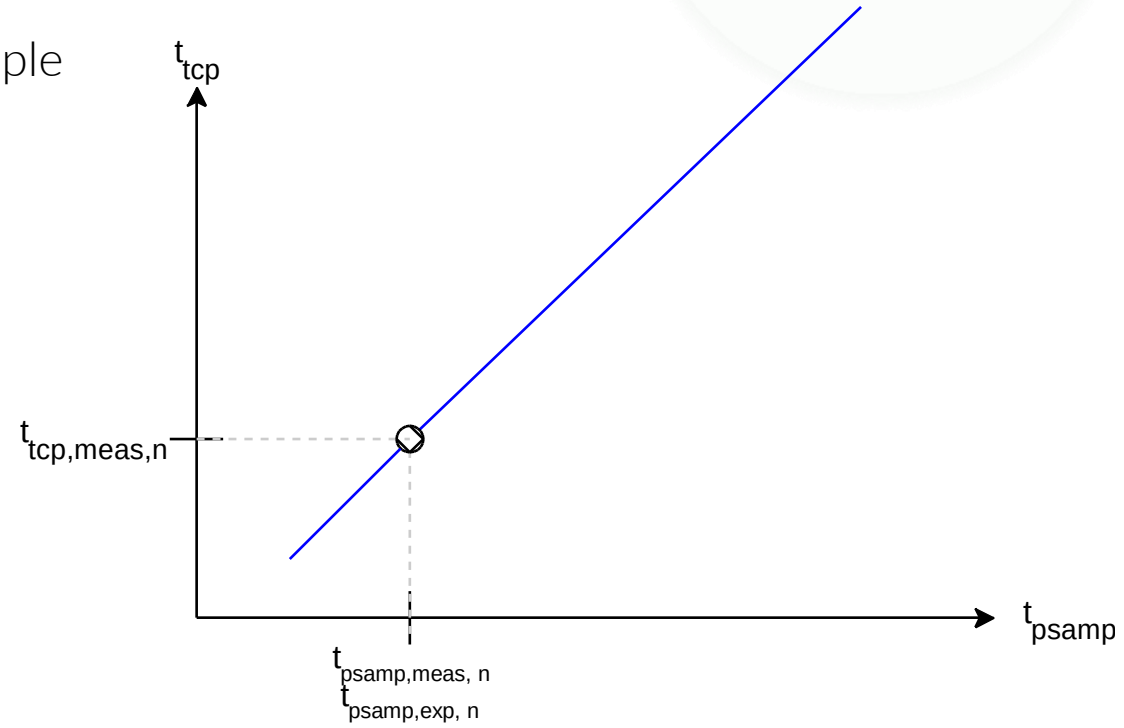
2. use initial offset for calculating expected timestamp of next sample

$$t_{psamp,exp,2} = \frac{t_{tcp,obs,2} - b}{m}$$

3. update *b* if $t_{psamp,exp,2} > t_{psamp,obs,2}$
4. repeat calculations for some/all subsequent samples to determine minimum/maximum offset

Open Issue

examine convergence behavior of offset



Preliminary Results

Measurement & processing setup

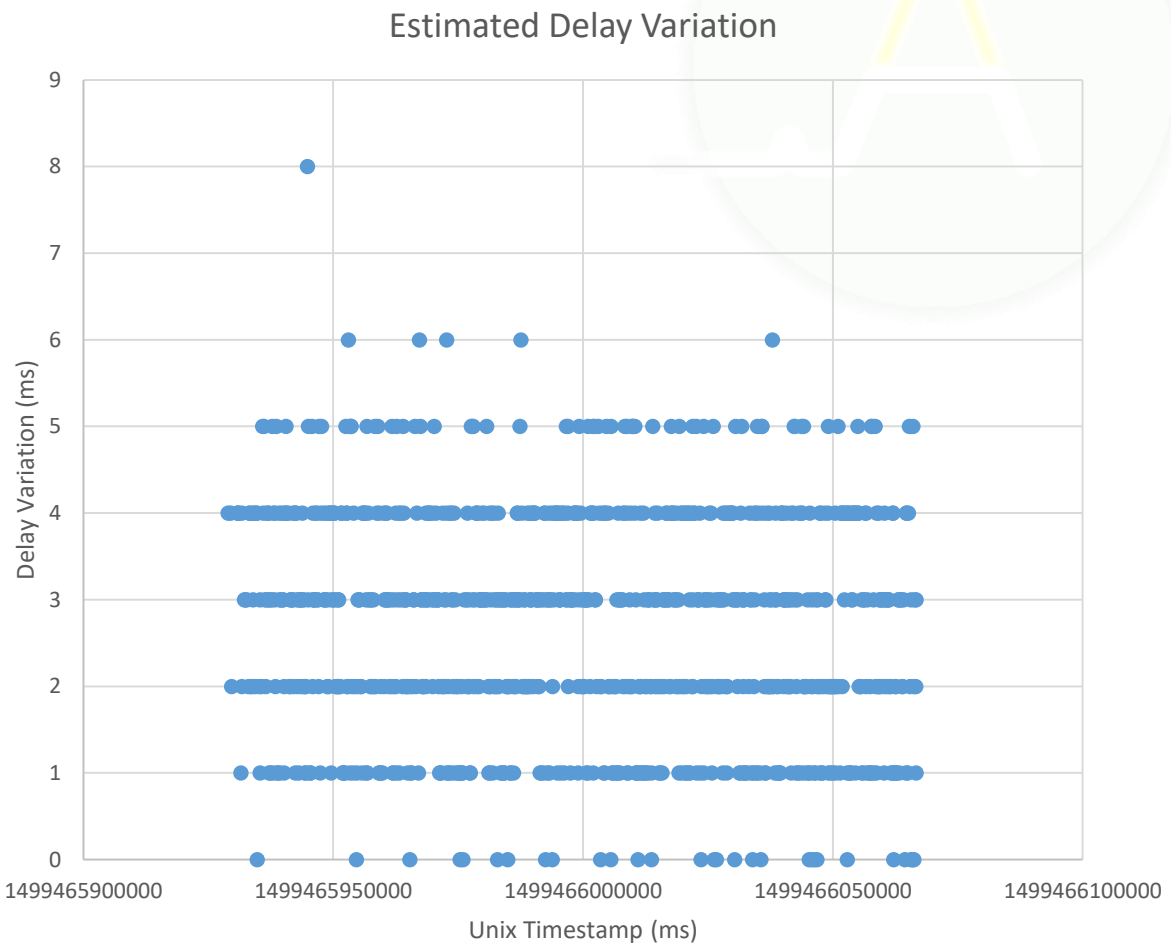
- packet sampling in IsarNet intranet
  - LAN + WAN traffic
  - no well-known test traffic
  - no well-known delay/jitter

  →no lab conditions
- offline processing

LAN-Traffic

- delay variation typically ~ 1-5ms
- at first glance no outliers

→measurement accuracy probably ~5ms



Preliminary Results

Measurement & processing setup

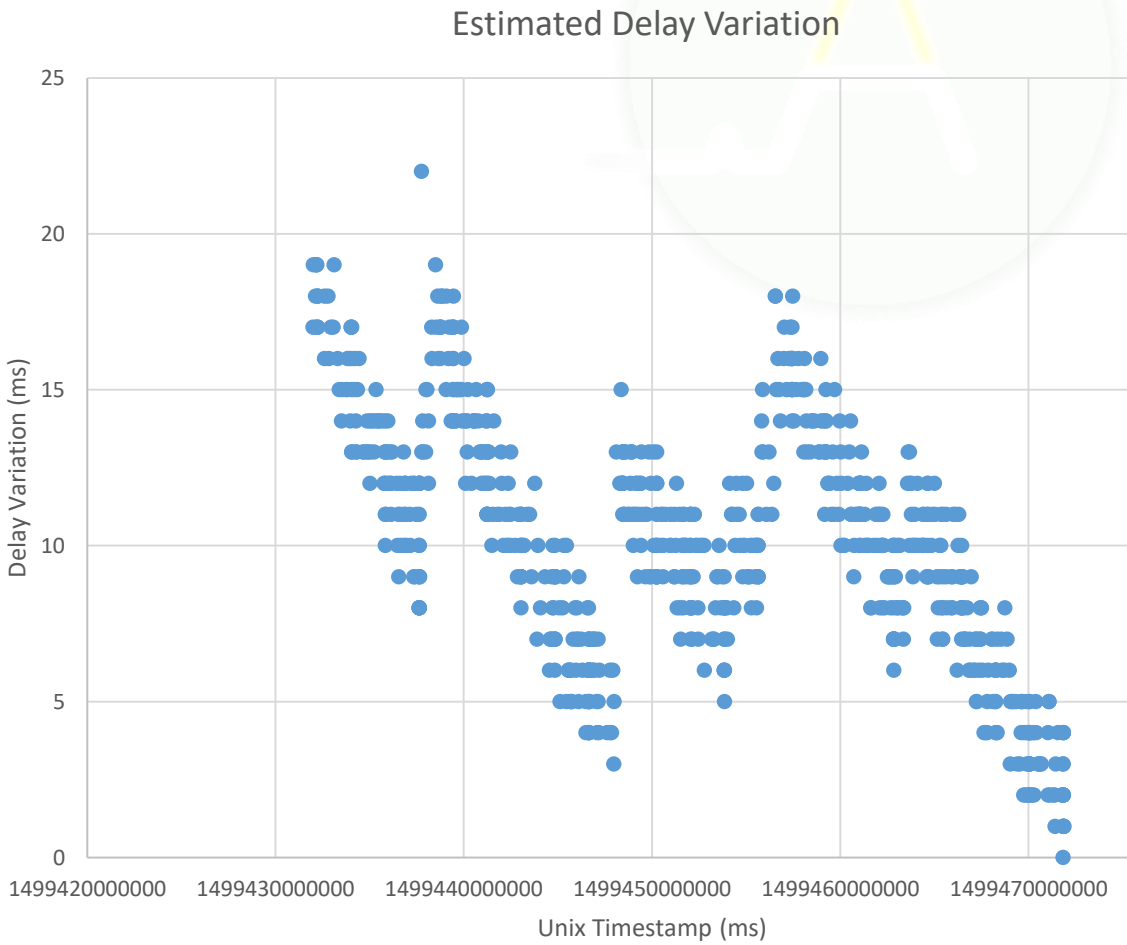
- packet sampling in IsarNet intranet
  - LAN + WAN traffic
  - no well-known test traffic
  - no well-known delay/jitter

  →no lab conditions
- offline processing

Other first observations

- **some** long lived flows (here: ~12h) show saw tooth pattern

  →probably clock drift in host
- might have to consider clock drift, and other clock effects in future work



AutoMon Control

Bigger picture of closed loop control

- TCP-timestamp analysis as first indicator
- starts further monitoring / data analysis automatically
- automatic drill-down without need for 100% fine-grained monitoring

→ AutoMon Controller

Closed loop control for timestamp analysis

- self-adaption of sampling rate
- …measurement points, exported fields
- …analysis confidence

Conclusion and Outlook

Conclusion

- passive monitoring of delay variation using TCP timestamps seems feasible in our initial scenarios
- assumption of negligible clock drift does not hold
- timestamp accuracy of flow data has improved a lot

Outlook

- further evaluation in
  - lab setup under well-known conditions
  - production network of application partner
- migration towards online processing – also taking into account clock drift

References

[1] Q. Scheitle, O. Gasser, M. Rouhi and G. Carle:
Large-Scale Classification of IPv6-IPv4 Siblings with Variable Clock Skew, 2017.

[2] J.Kögel: One-Way Delay Measurement based on Flow Data in Large Enterprise Networks,
Dissertation, Universität Stuttgart, 2013.

Acknowledgement

This work was partly funded as part of the AutoMon project by the German Federal Ministry of Education and Research (BMBF) under contract No. 16KIS0408K. Responsibility for the information and views expressed in this publication lies entirely with the authors.