# Reformulating the monitor placement problem: Optimal Network-wide Sampling

Gion-Reto Cantieni (EPFL)
Gianluca Iannaconne (Intel)
**Chadi Barakat (INRIA Sophia Antipolis)**
Patrick Thiran (EPFL)
Christophe Diot (Thomson)

CONEXT 2006 - Lisbon

# Motivation

- State of the art: Placing monitors to cover the network. But,
- Router-embedded monitoring functionalities are commonly used in small and large ISPs, e.g., Cisco's NetFlow.
- Packet Sampling to reduce the monitoring overhead.
  Set manually and independently of the measurement task.
- Challenge:

  *Instead of placing monitors, how to configure routers so as to realize the desired monitoring task?*

  - "Configure" means setting the sampling rates on all individual interfaces.
  - A sampling rate of zero means a router does not participate.

# Why is it a hard challenge?

- Sampling rates needs to be low to reduce stress on routers.
- Aggregate volume of information collected from the routers should be kept under control.
- Measurement task unknown a priori and so, a single fixed layout does not perform well
  - e.g., PoP-level traffic matrix estimation
    → all edge routers with low sampling rates.
  - e.g., focusing on specific prefix
    → few monitors, relatively higher sampling rates.

# Our objective

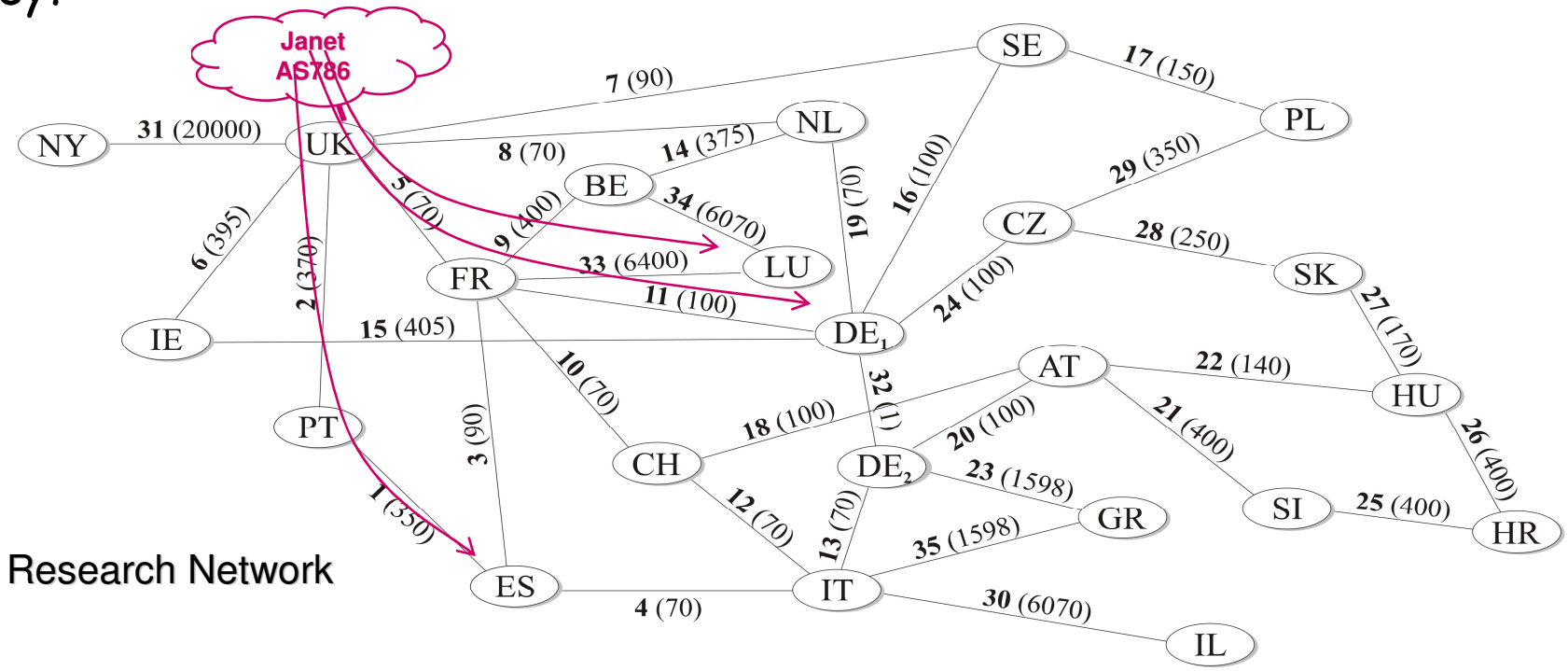
- Given a measurement task and a target accuracy, find a method that:
  - sets the sampling rate on all interfaces (some can be turned off).
  - guarantees optimal use of resources
    (in terms of processed packets and volume of collected data)
  - can adapt to changes in the traffic
- Method should apply to a general class of measurement tasks

# Related work

- Passive monitoring
  - Suh et al, "Locating Network Monitors...", Infocom 2005
    - two phase approach: select the monitors then optimize sampling
- Packet sampling
  - Duffield et al, "Estimating flow distributions...", ACM Sigcomm 2003.
    - Random sampling is a good approximation of real implementations.
    - Hard to invert the flow size distribution without additional information.
  - Duffield et al, « Predicting Resource Usage ... », IMC 2003.
    - Size-based sampling of NetFlow records.
- Improving NetFlow
  - Estan et al, "Building a better NetFlow", Sigcomm 2004
    - Adapt on a per-router basis the sampling rate to limit the collected traffic

# How do we proceed? First pick a measurement task...

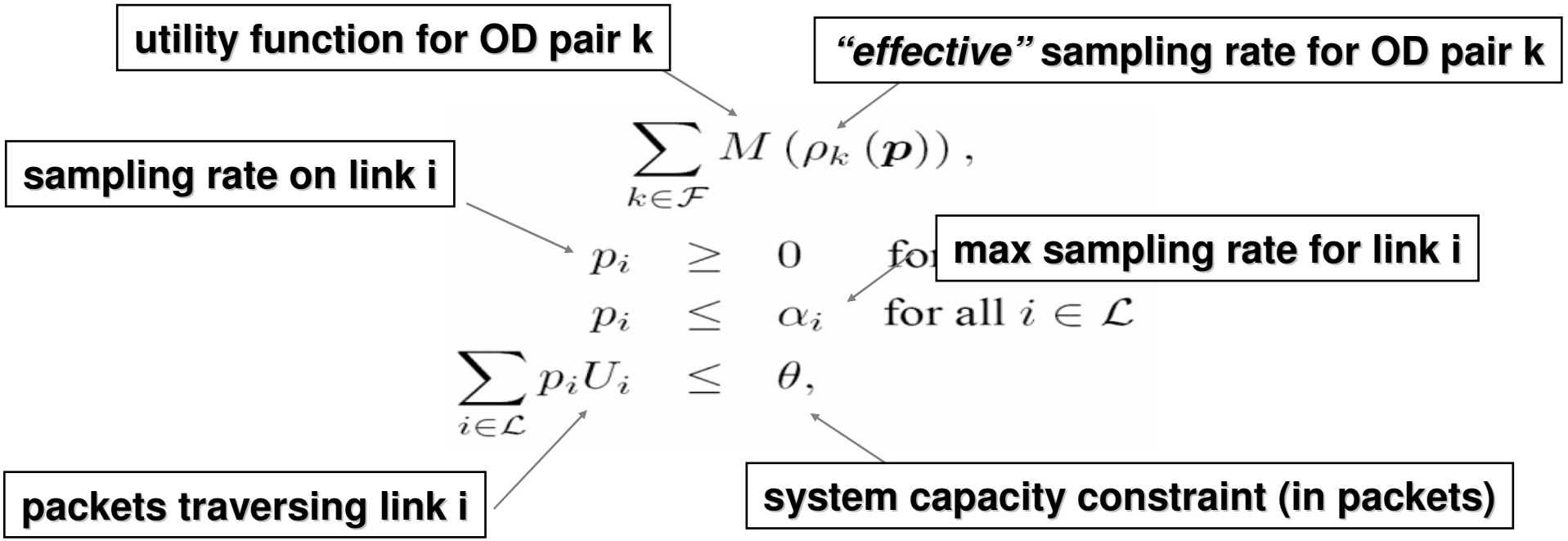
- Estimate amount of traffic flowing among a subset of origin-destination pairs (Common task for traffic engineering apps)
- Where to tune on the sampling and to which rate in order to maximize accuracy?




GEANT
European Research Network

# Problem formulation

**Choose vector of sampling rates *p* that maximizes**




- Effective sampling rate approximated by sum of sampling rates
- All constraints are linear and define a convex solution space
  - Unique maximizer exists as long as **M(.)** is strictly concave
- Problem solved numerically

# The utility function

- Measures the quality of sampling an OD pair
- Our example: 1 - Mean square relative error
  - $E[SRE] = E[((X/\rho - S) / S)^2]$

    where S is the size of the OD pair.
  - We need to estimate 1/S. A bootstrapping phase to do that.

    Start with a default p, estimate, optimize, estimate, and so on.
- Other functions could be possible to model other measurements tasks (left for future research):
  - accuracy of ranking/estimating the largest flows.
  - accuracy of estimating the flow size distribution.
  - accuracy of anomaly detection.

# Evaluation

- Consider NetFlow data from GEANT
  - Collected using Juniper's Traffic Sampling
  - 1/1000 periodic sampling
  - We scale the measurement by 1000
    (we just need a realistic mix of OD pair sizes)
- Get OD pair sizes and link loads every 5 minutes
- Solve the algorithm for the sampling rates that allow to estimate the sizes of the OD pairs originated at UK.
- Set θ to 100K packets. Don't limit the sampling rate.

# Summary of results

- Accuracy is on average in the range 89-99%
  - Worst accuracy for JANET – LU (it has just 20 pkts/sec)
- Measurement spread across 10 links
- Max sampling rates is 0.92% (lightly loaded links)
  - Most links are around 0.1%
  - No OD pair is monitored on more than two links
- Main observation:

  The high error is for small OD pairs especially if they go through highly loaded links. They need high sampling rate, but high sampling is lot of collected unwanted traffic !!

# Performance over time: Accuracy

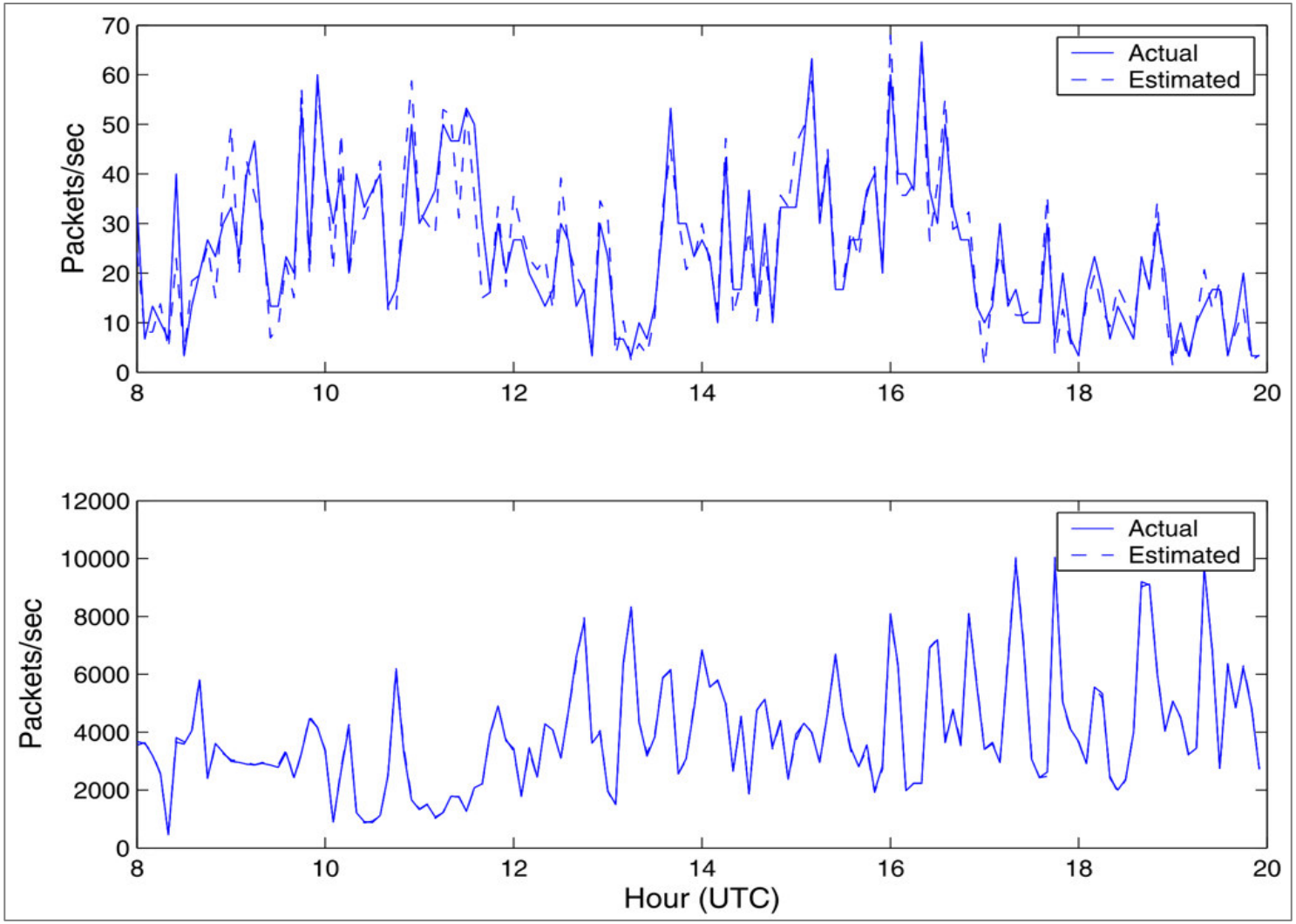
## Optimal sampling rates computed once at the beginning

UK-LU

Small OD pair

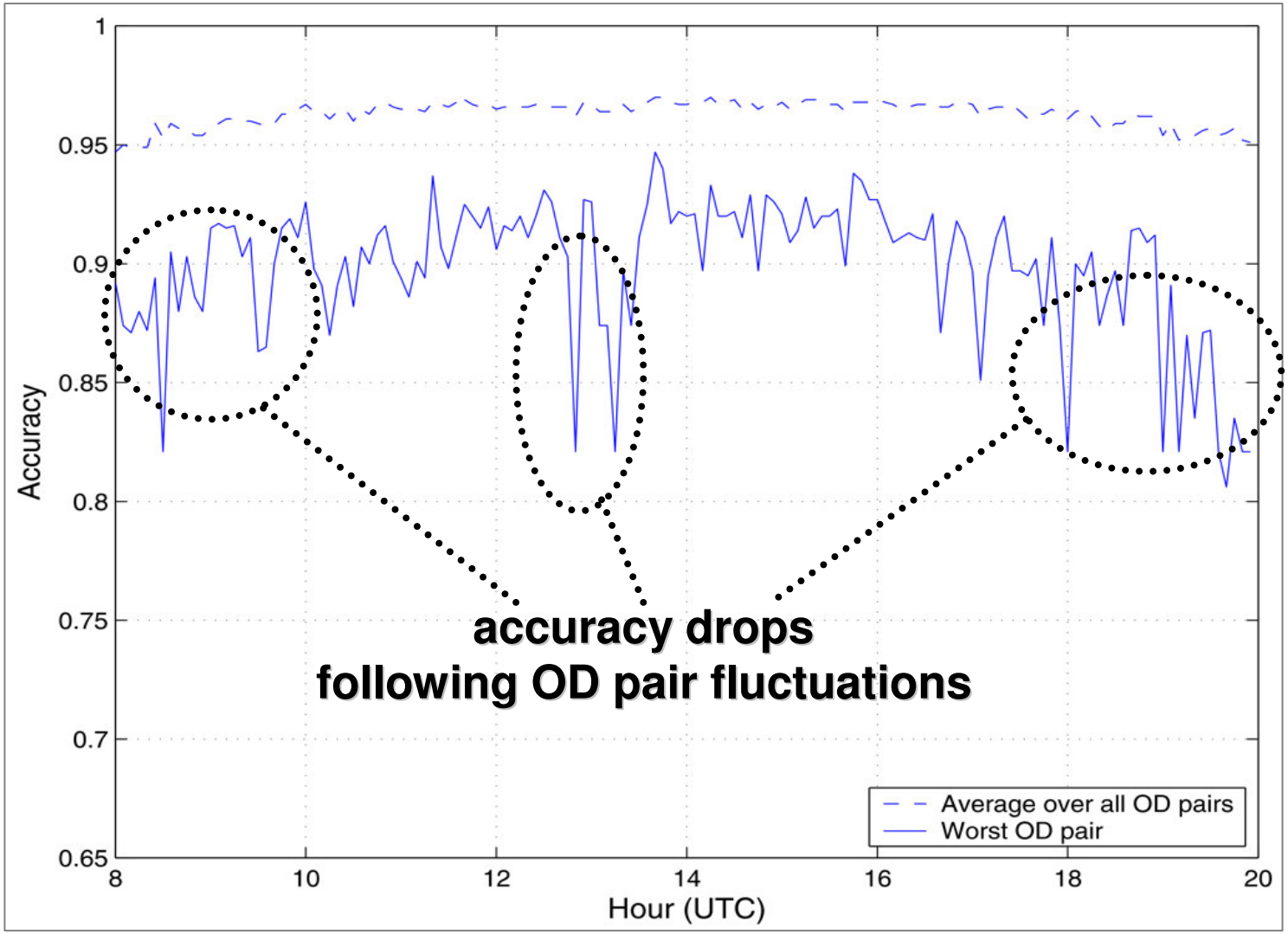
UK-CH

almost zero error



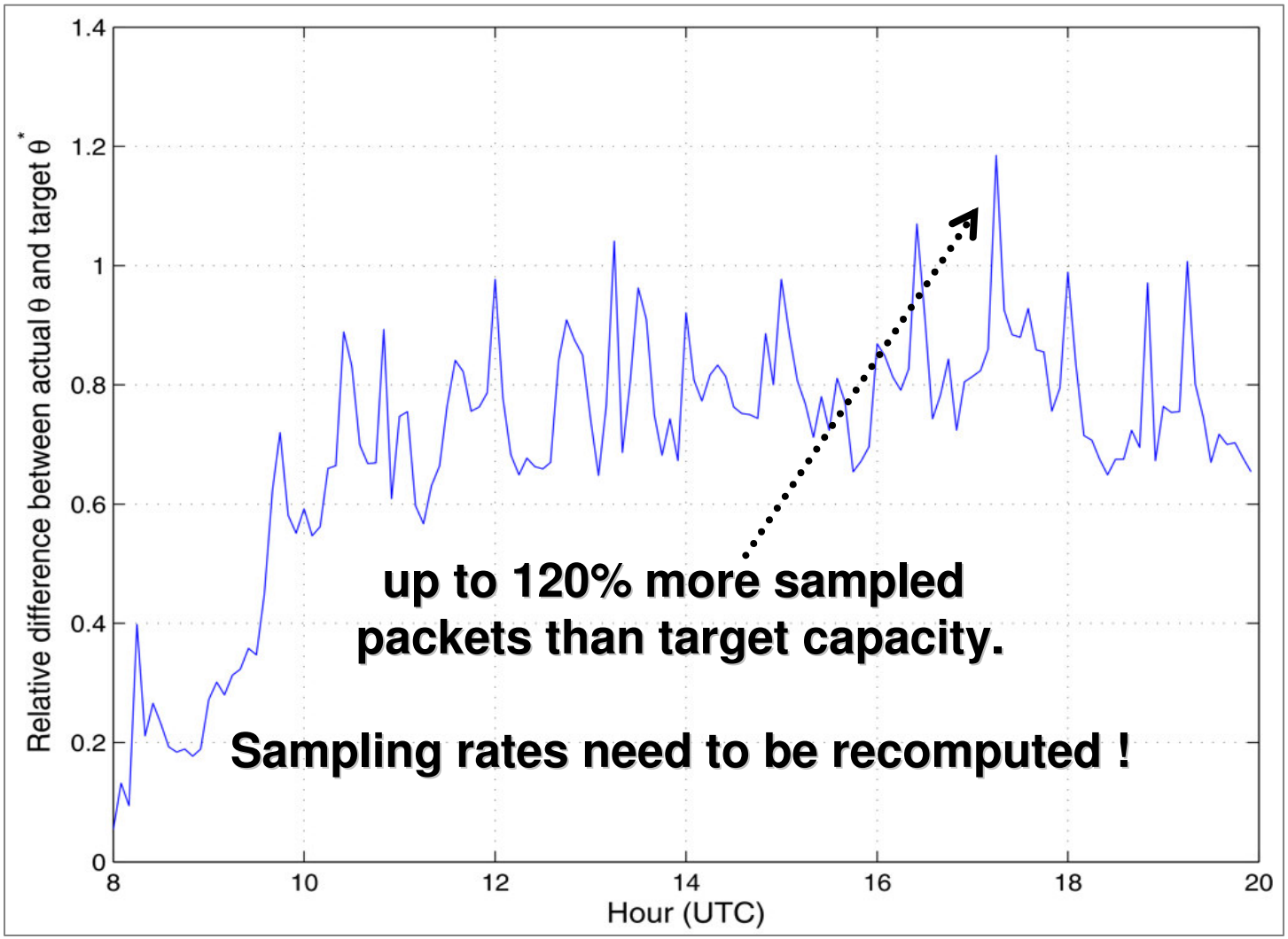
# Performance over time: Accuracy

Average results over 20 experiments



# Performance over time: Overhead

Collected traffic vs. capacity for one experiment



# Adapting to traffic fluctuations

Three different cases that require different approaches

- Link load increases
  - more sampled packets, exceeding capacity
- OD pair decreases in volume
  - poor accuracy because of bad 1/S estimate
- OD pair traverses different set of links
  - missing entire OD pair

find new sampling rates to enforce target capacity,
and if accuracy is not OK, adapt θ

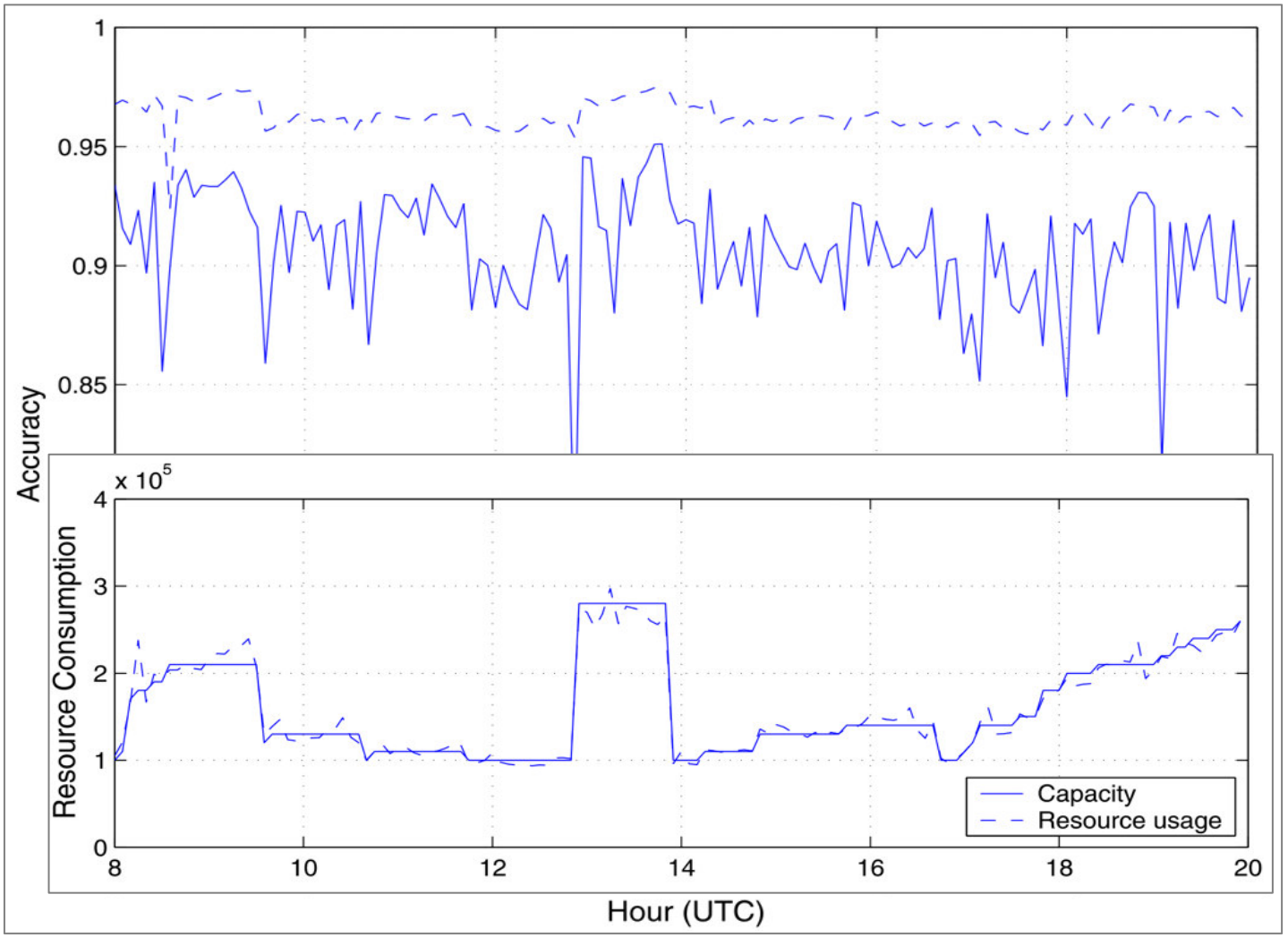
# Dynamic version of our algorithm

- Compute new sampling rates when estimated accuracy drops below target or collected traffic exceeds capacity
- If the estimated accuracy is still below target, increase capacity constraint by some factor say 10%
- Decrease capacity constraint if estimated accuracy is above target for more than some time (say one hour)

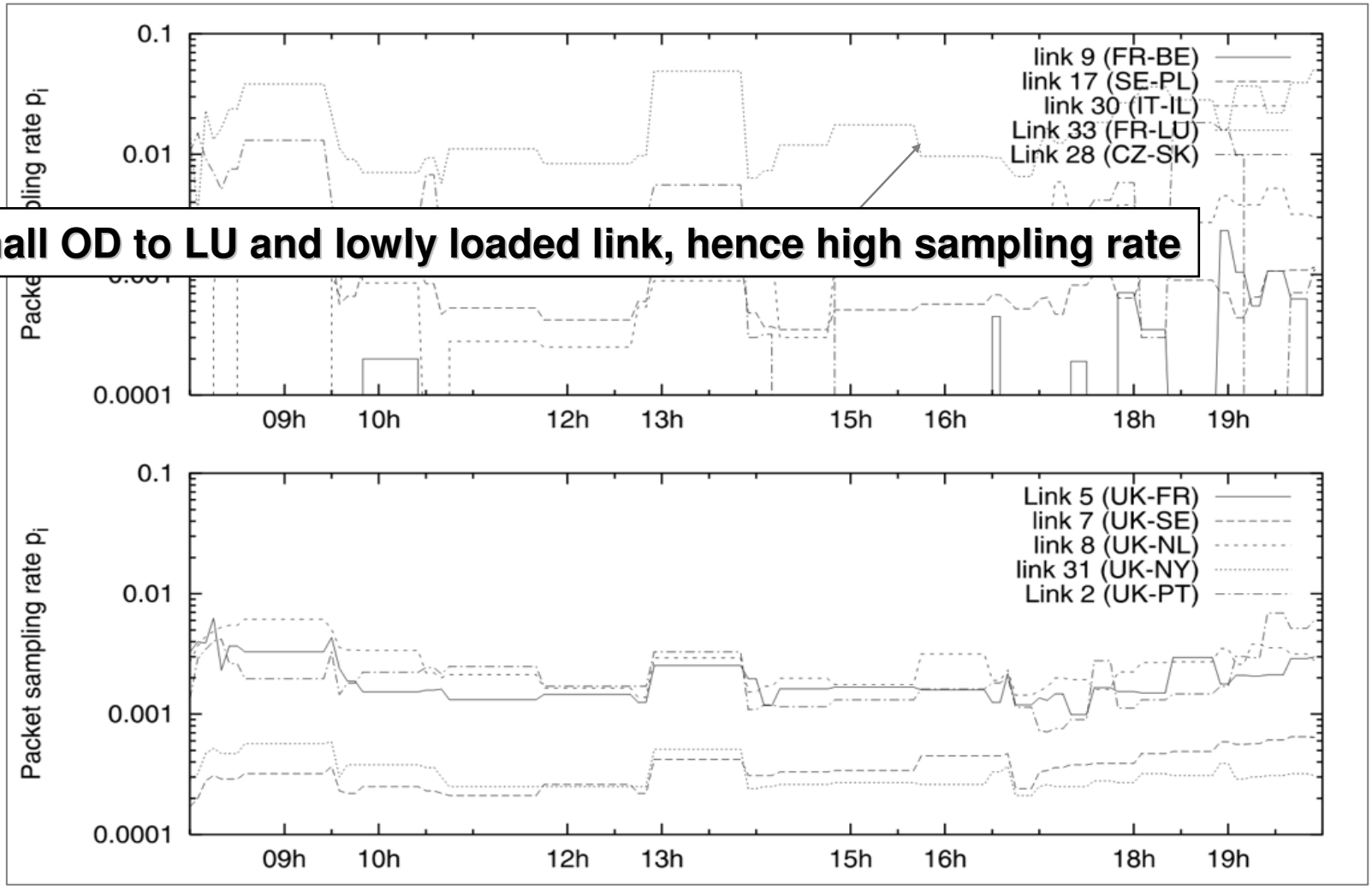
# Evaluation dynamic version

**Target accuracy 85%**

**Resource consumption**



# Evaluation dynamic version




**small OD to LU and lowly loaded link, hence high sampling rate**

# Conclusions

- Set sampling rates of a network of monitors.
- General enough framework for large class of measurement tasks

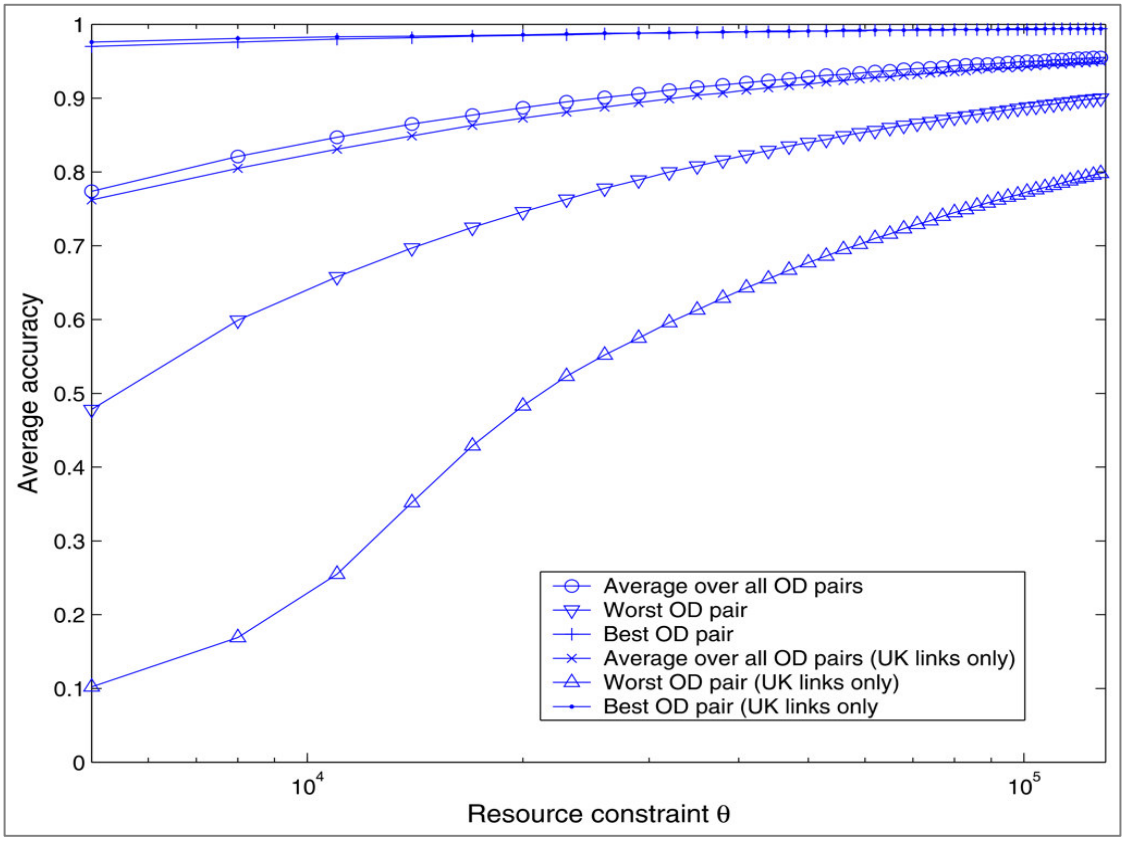
- Working on finding new utility functions
- Looking into using traffic predictors

- Open issue
  How long does it take to reconfigure NetFlow?

# Comparing to "naive" solutions

- Why not just monitoring JANET access link?
  - All the monitored traffic would be relevant!
  - To achieve same accuracy over all OD pairs we need ~1% sampling rate
    → 70% more packets are processed
  - It's not always possible to monitor both directions of access links
- Why not just monitoring all UK links?
  - There are just 6 links leaving the UK
  - Straightforward algorithm to set sampling rate (each OD pair is present on just one link), but...

# Monitoring all UK links




- Why does our method work better?
  - It looks across the entire network to find where small OD pairs manifest themselves without hiding behind large flows

# Deployment on real networks

- Two aspects need to be addressed
- What prior knowledge about the network does the method need?
  - need routing information
  - need estimate of E[1/S] for each OD pair
    → bootstrapping phase
- How does the method perform over time?
  - time of day effect change E[1/S] and $U_i$
  - routing event change path taken by OD pairs
    → adapt sampling rates

# Bootstrapping phase