Universidade do Minho
Portugal

A modular architecture for deploying self-adaptive traffic sampling

João Marco C. Silva
Paulo M. Carvalho
Solange Rito Lima

CENTROALGORITMI

Introduction

- Traffic sampling plays a key role to achieve efficient network measurements
    - to select a subset of packets which will be used to estimate network parameters, avoiding processing all network traffic
- Many sampling techniques have been proposed, but not deployed in measurement points

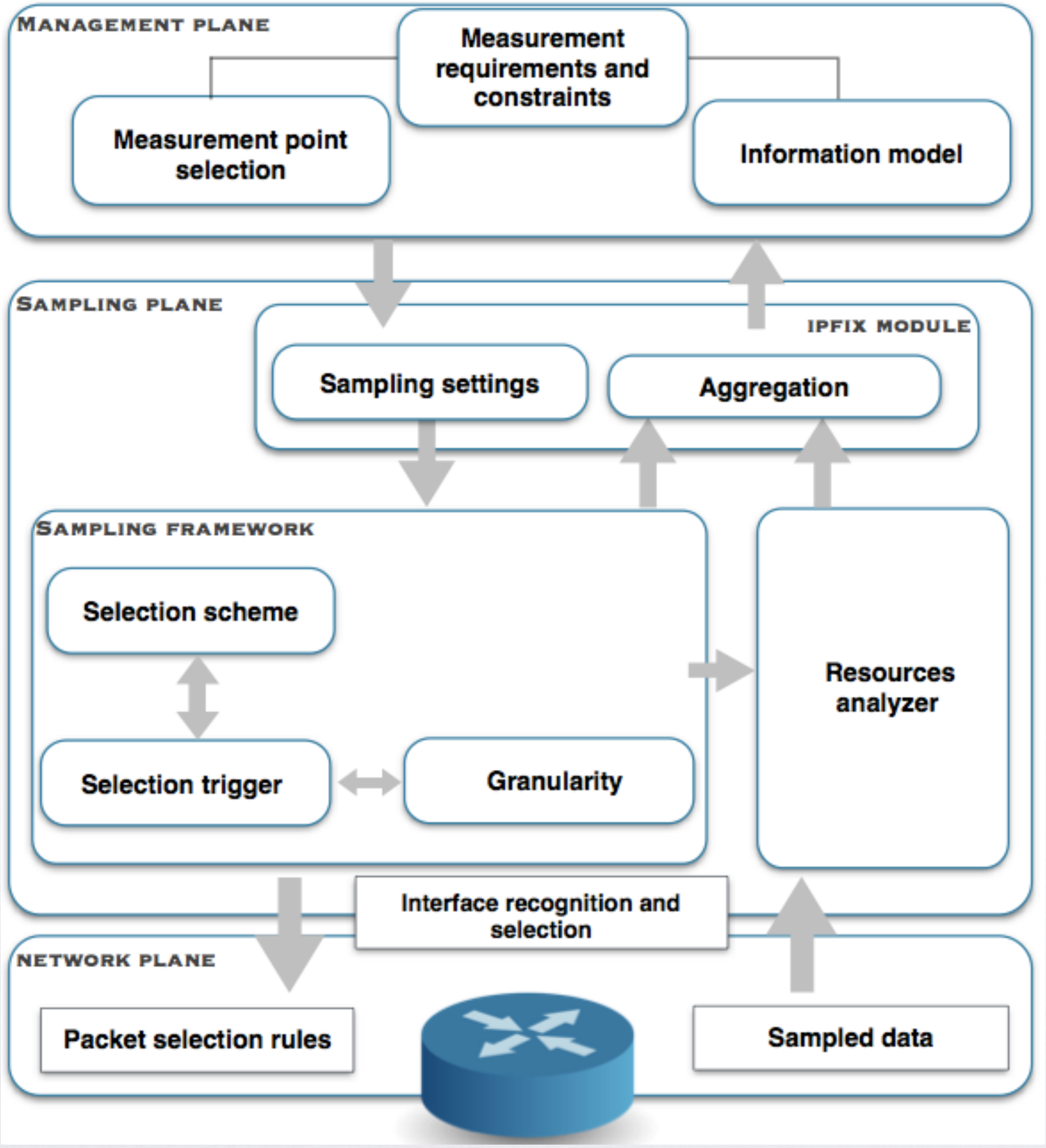
Introduction

- **Main issue: the balance between computational burden and accuracy**
  - Memory and CPU usage
  - Volume of data stored and transmitted
  - Accuracy in metric estimations

Objective

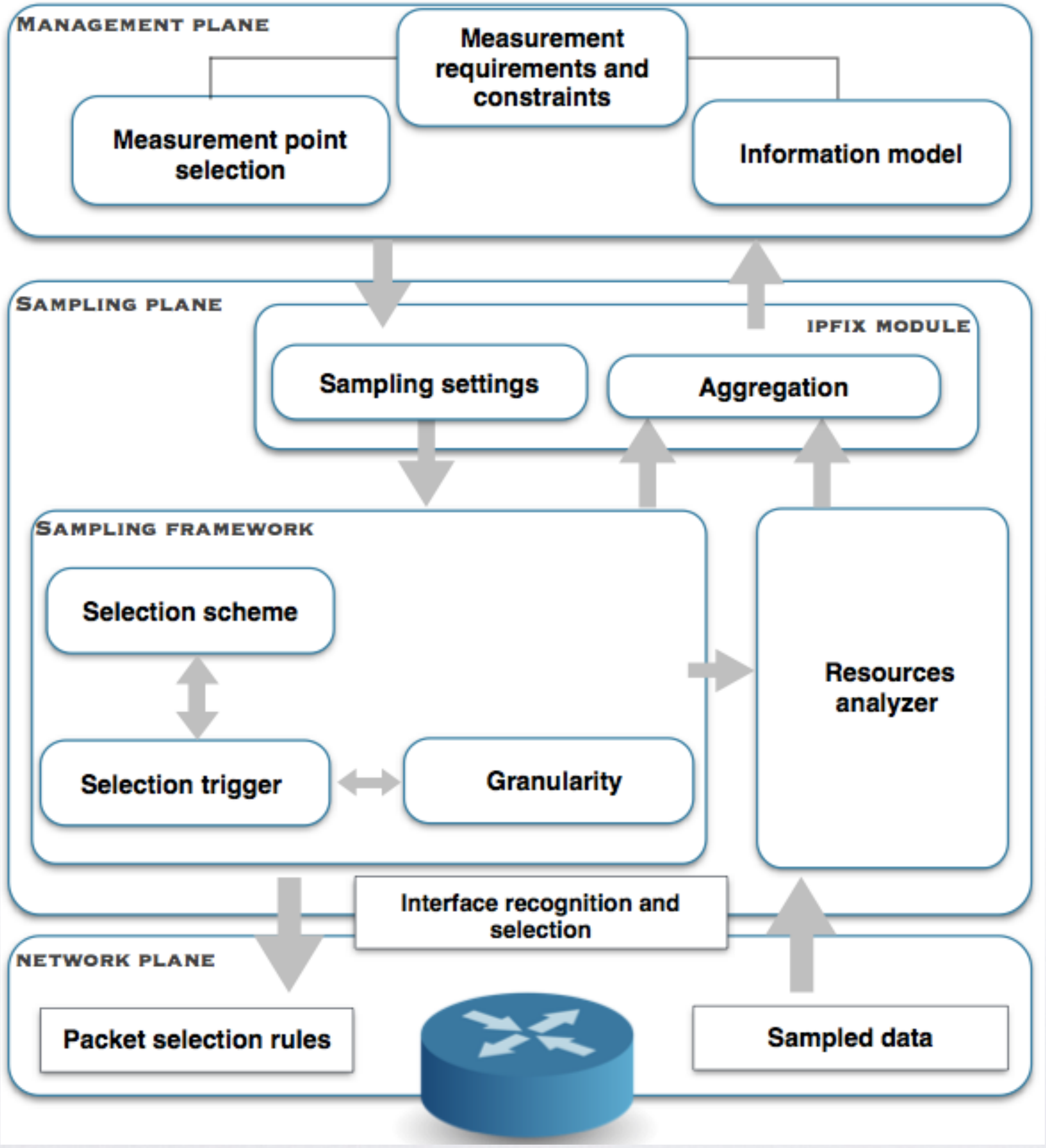
- **To deploy a modular and self-adaptive architecture able to accommodate the selection and configuration of sampling techniques according to the requirements of accuracy and resources available**

Measurement architecture



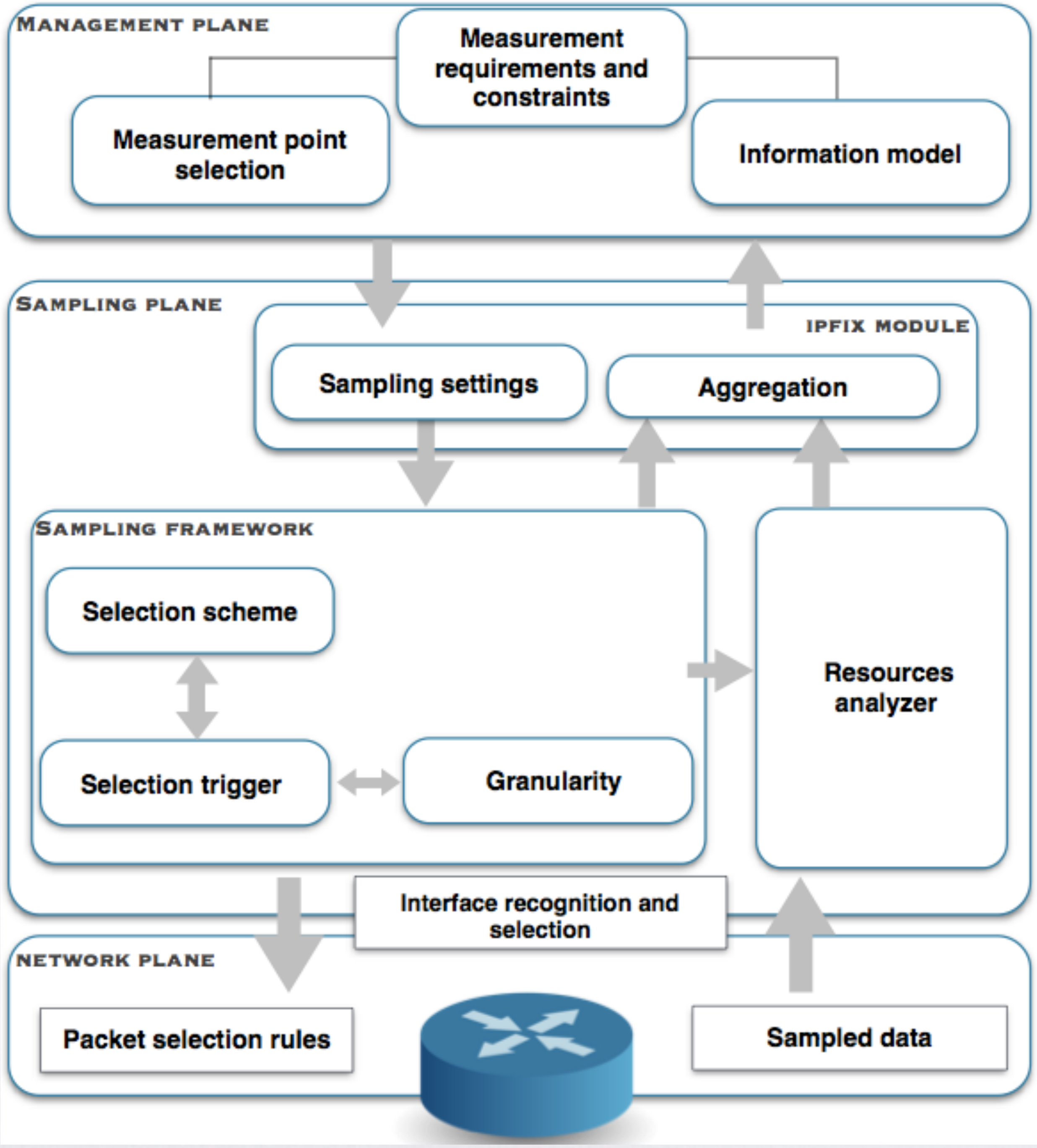
- Three layers design
  - Management plane
  - Sampling plane
  - Network plane
- Modular components
  - scalability

Measurement architecture



- Management plane
  - measurement needs and constraints identification
  - Sampling technique selection and configuration
  - Self-adaptive behavior

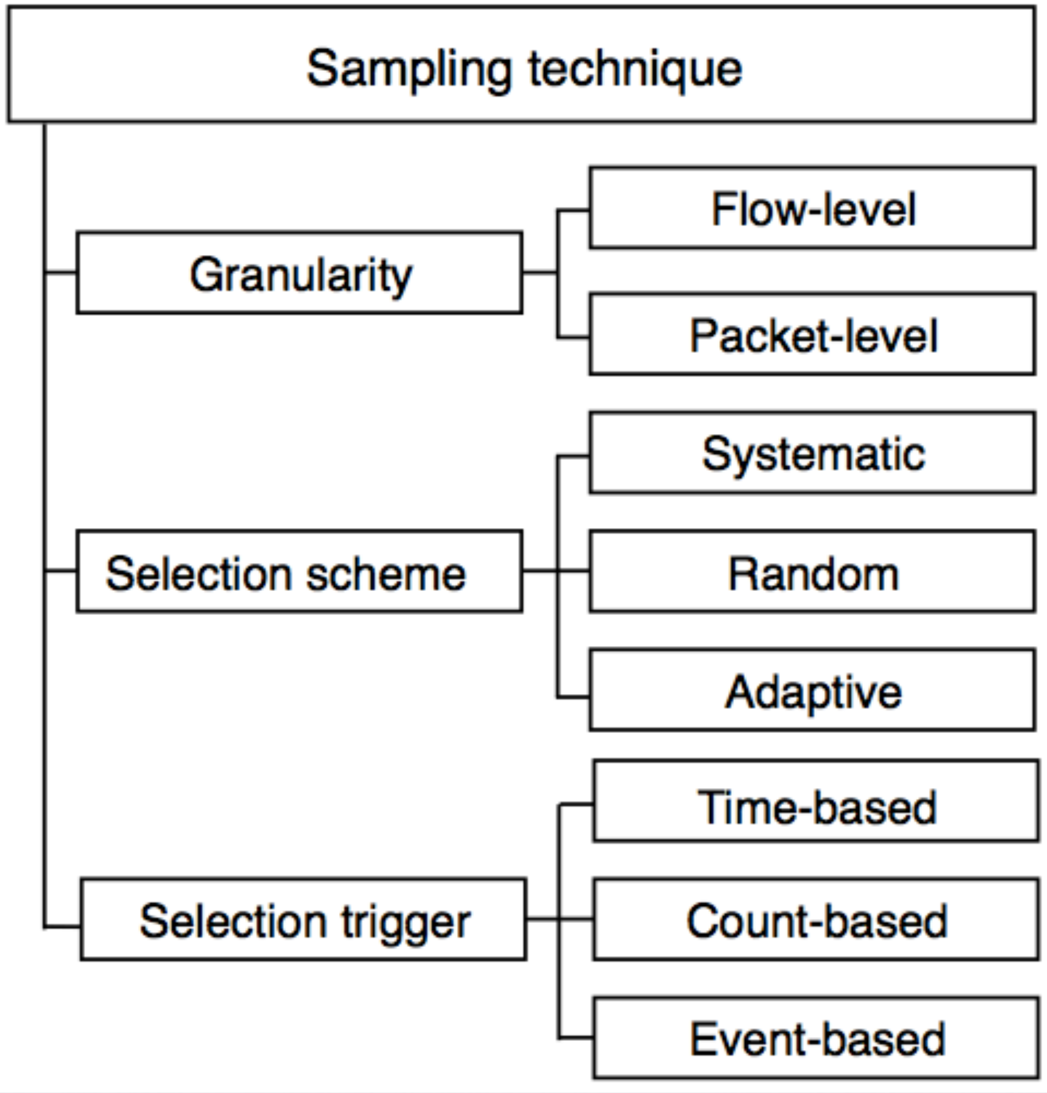
Measurement architecture



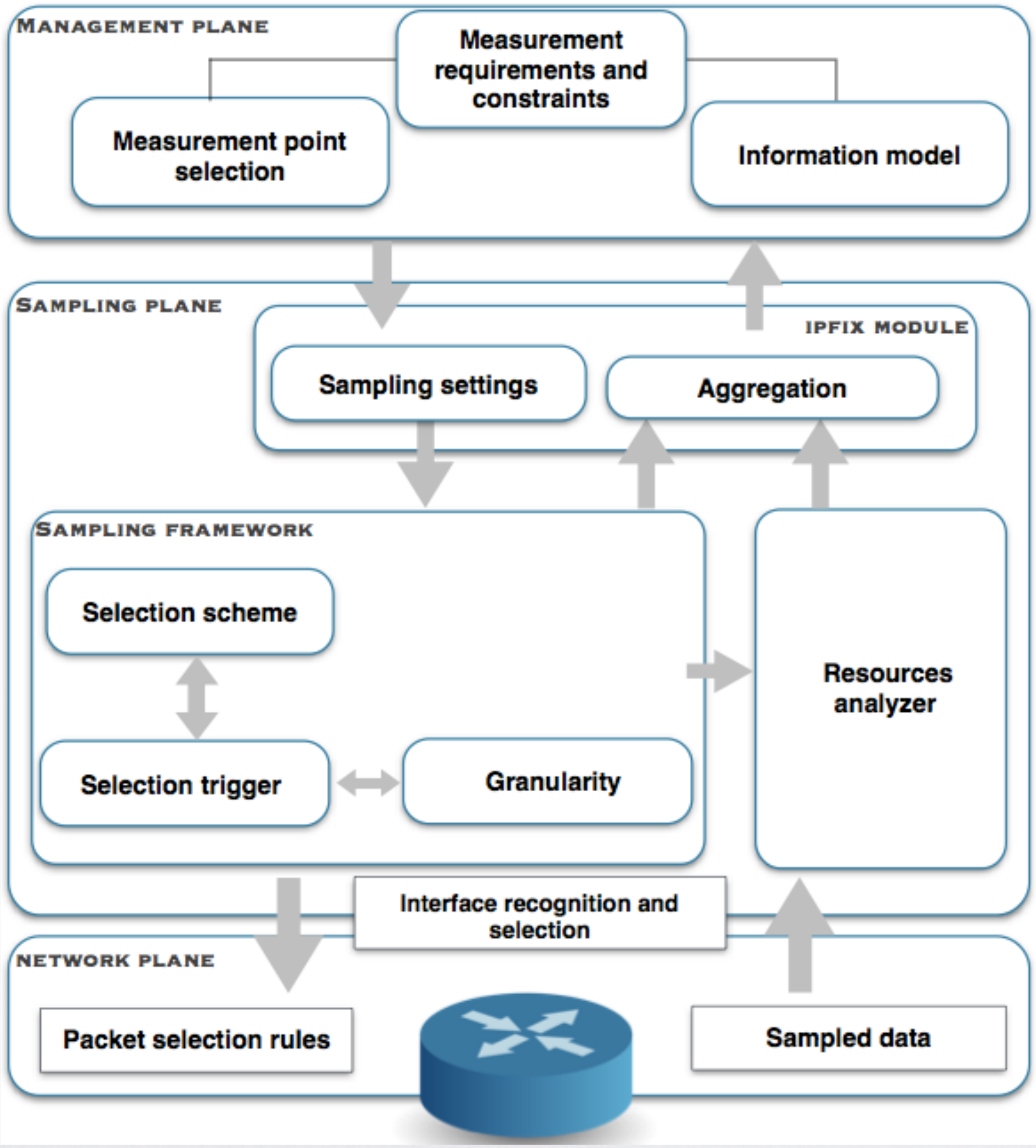
- Sampling plane
  - modular sampling framework
  - resources analyzer
  - aggregation and exporting
    - IETF IPFIX

Measurement architecture

- **Sampling plane - framework**



Measurement architecture



- Network plane
  - simple tasks
    - traffic capture from network interface
    - reporting unprocessed collected packets

Ongoing works

- Deployment of the sampling plane and network plane
- Quantitative comparison of the computational burden / accuracy of different sampling techniques
  - in presence of similar workload
  - to support the design of an efficient adaptive module

Ongoing works

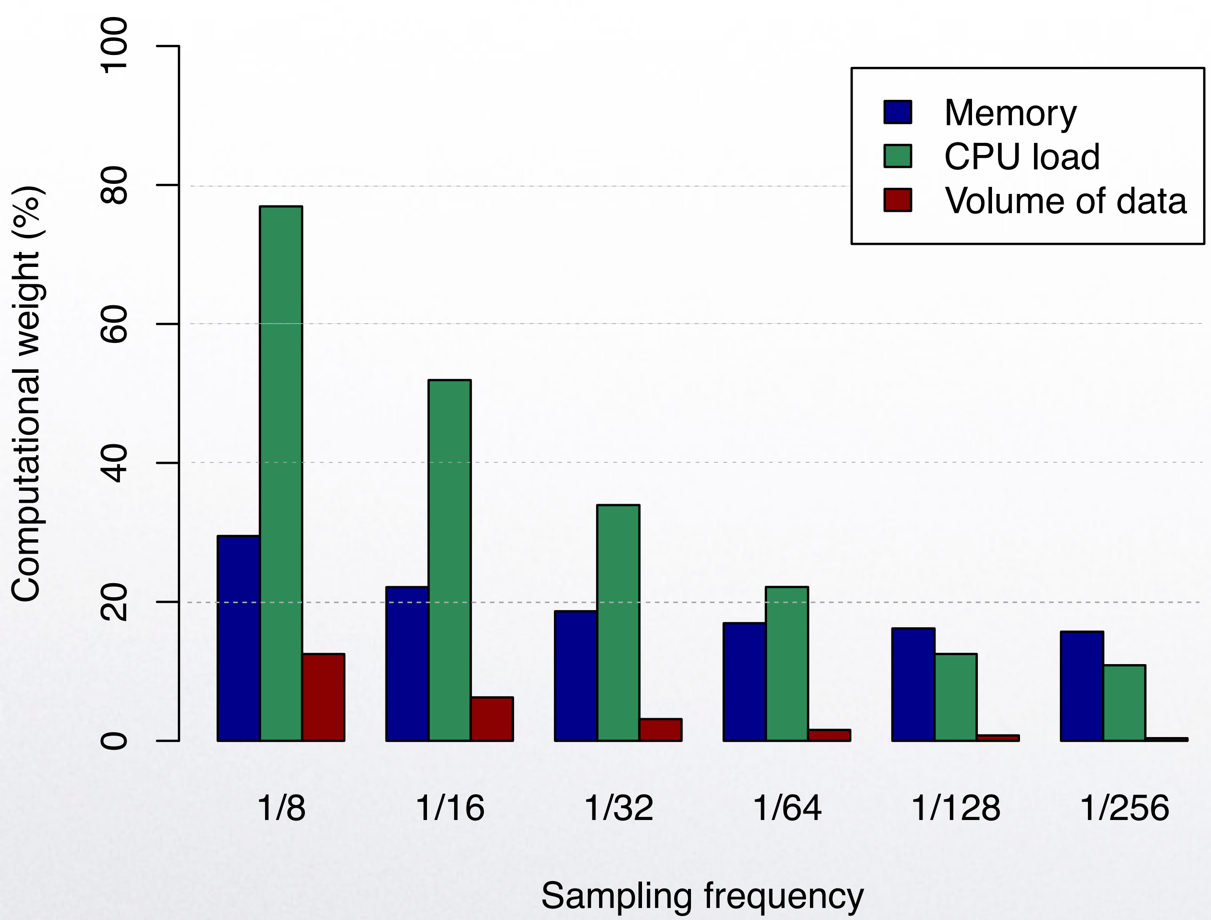
- Sampling techniques analyzed
  - Systematic count-based (SystC)
  - Systematic time-based (SystT)
  - Random count-based (RandC)
  - Adaptive linear prediction (LP)
  - Multiadaptive (MuST)

Ongoing works

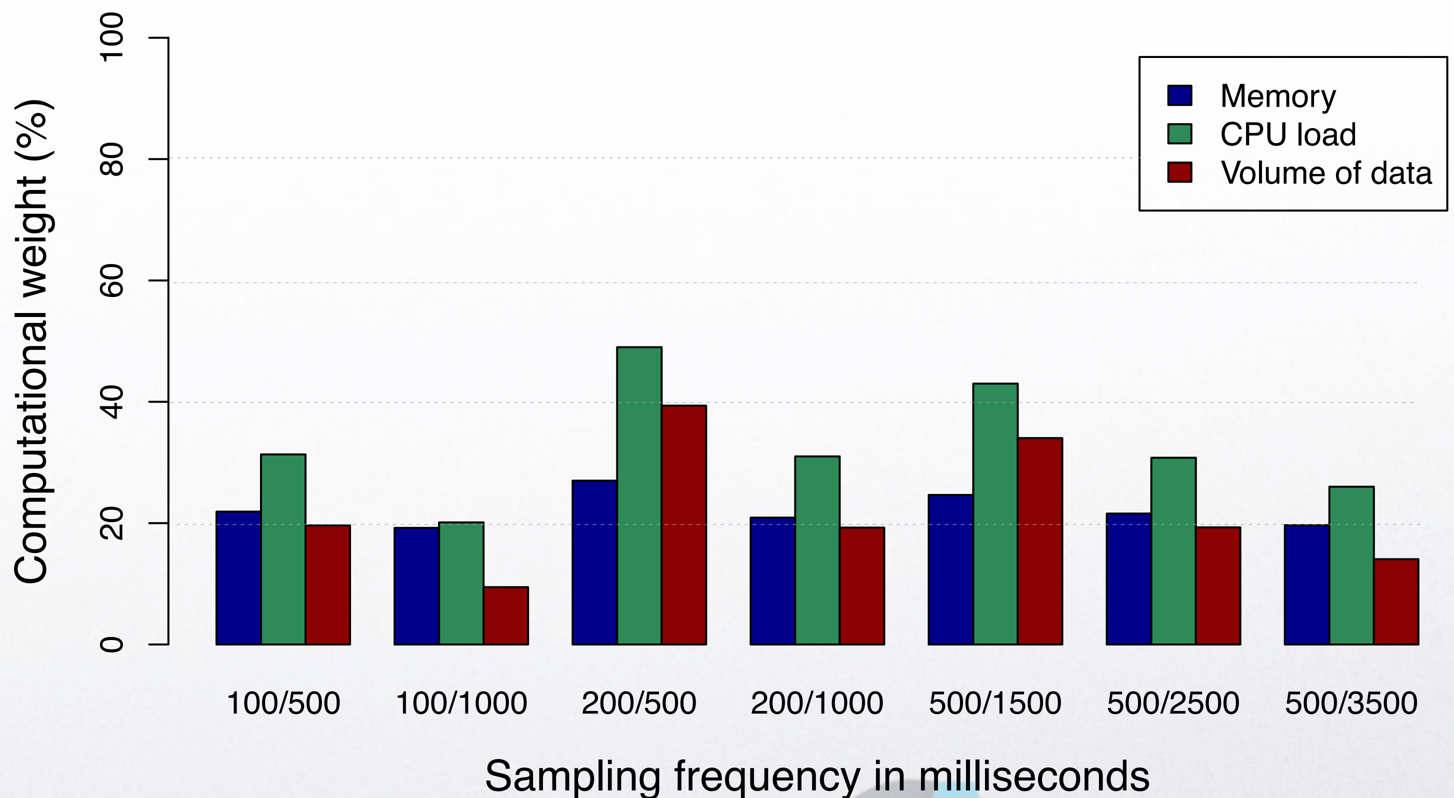
- Comparative parameters
  - Computational weight
    - CPU load, memory usage and volume of data
  - Accuracy
    - instantaneous throughput, mean throughput, mean packet size
      - resorting to descriptive statistics

Early results

- Systematic techniques - high workload scenario



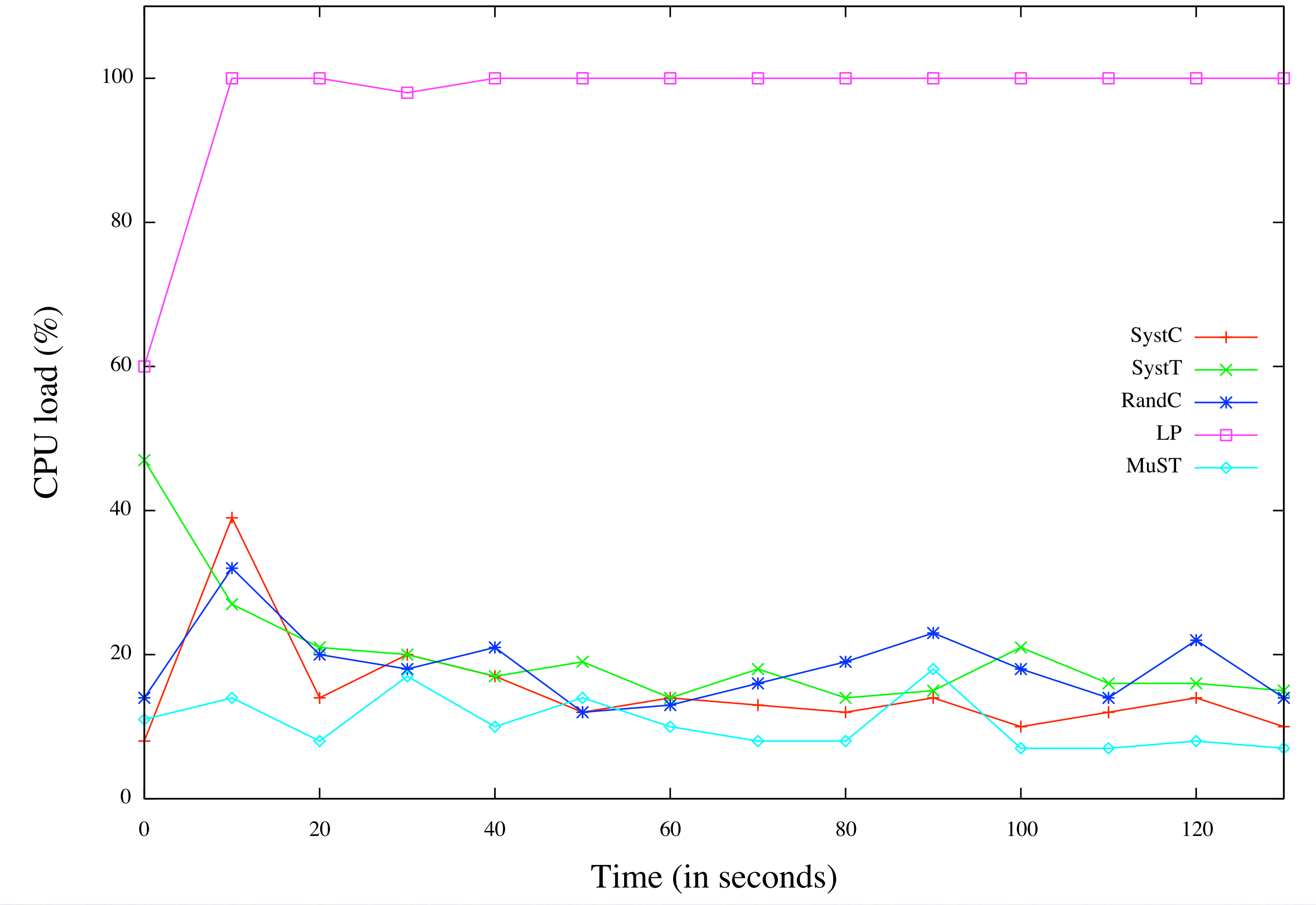
Count-based

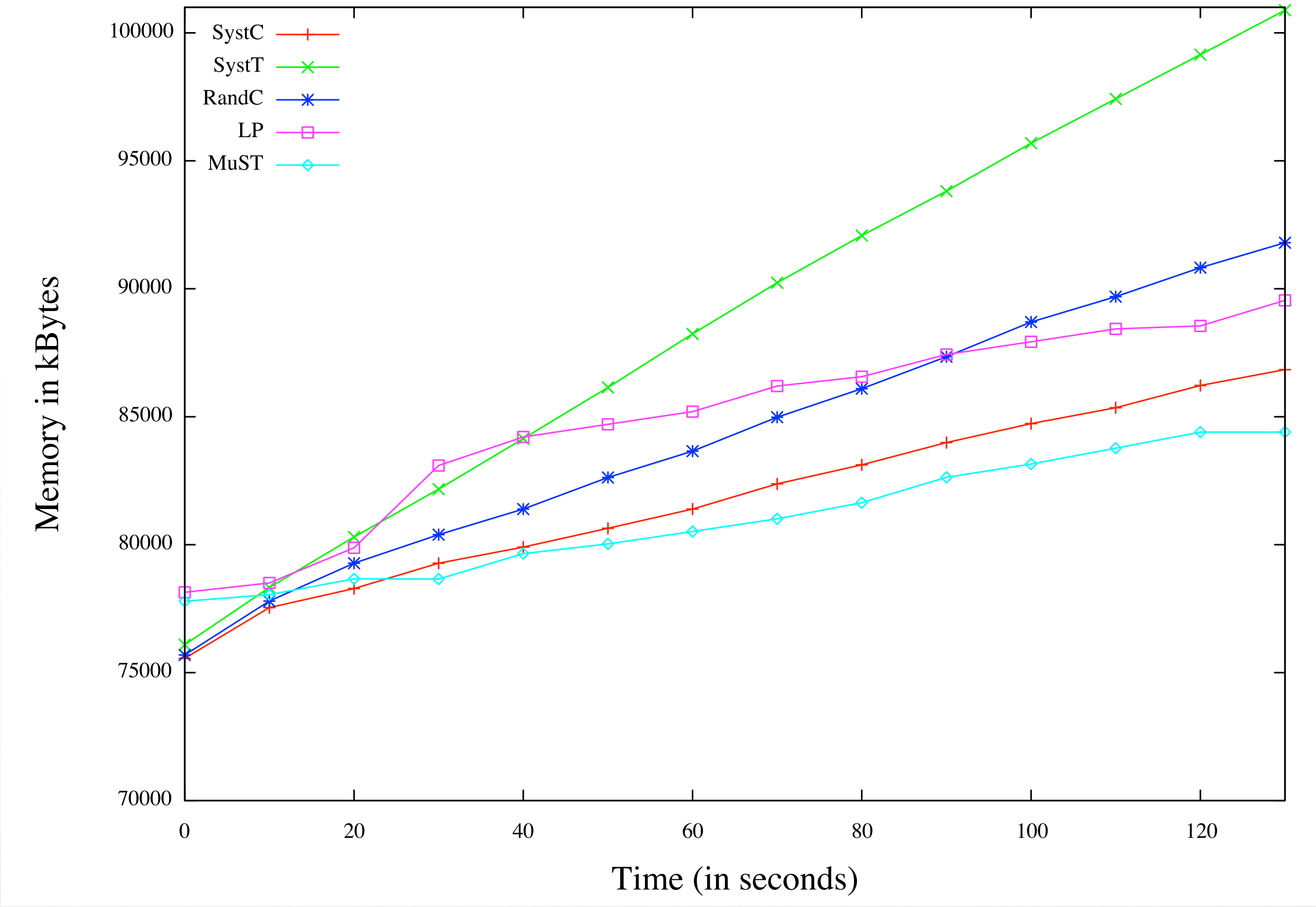


Time-based

Early results

- **Computational weight - high workload scenario**





Early results

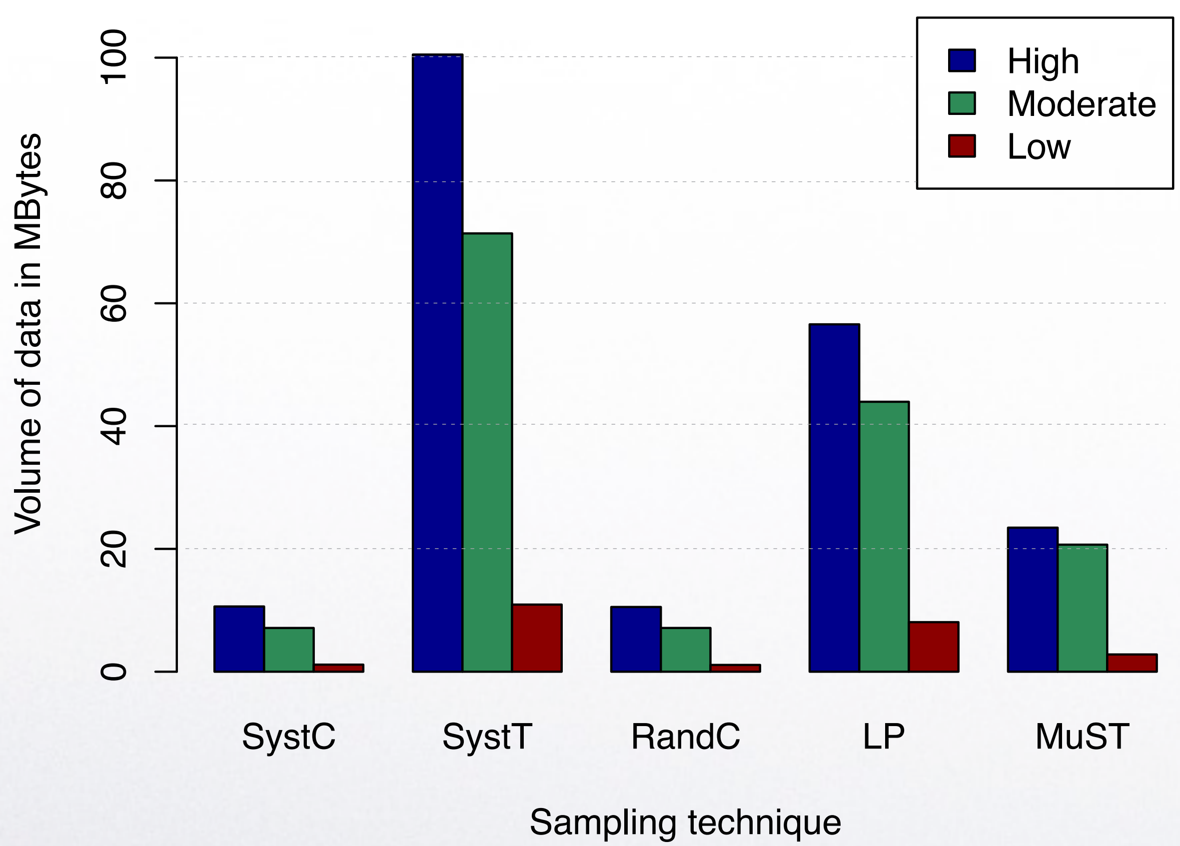
- Computational weight - all traffic scenarios

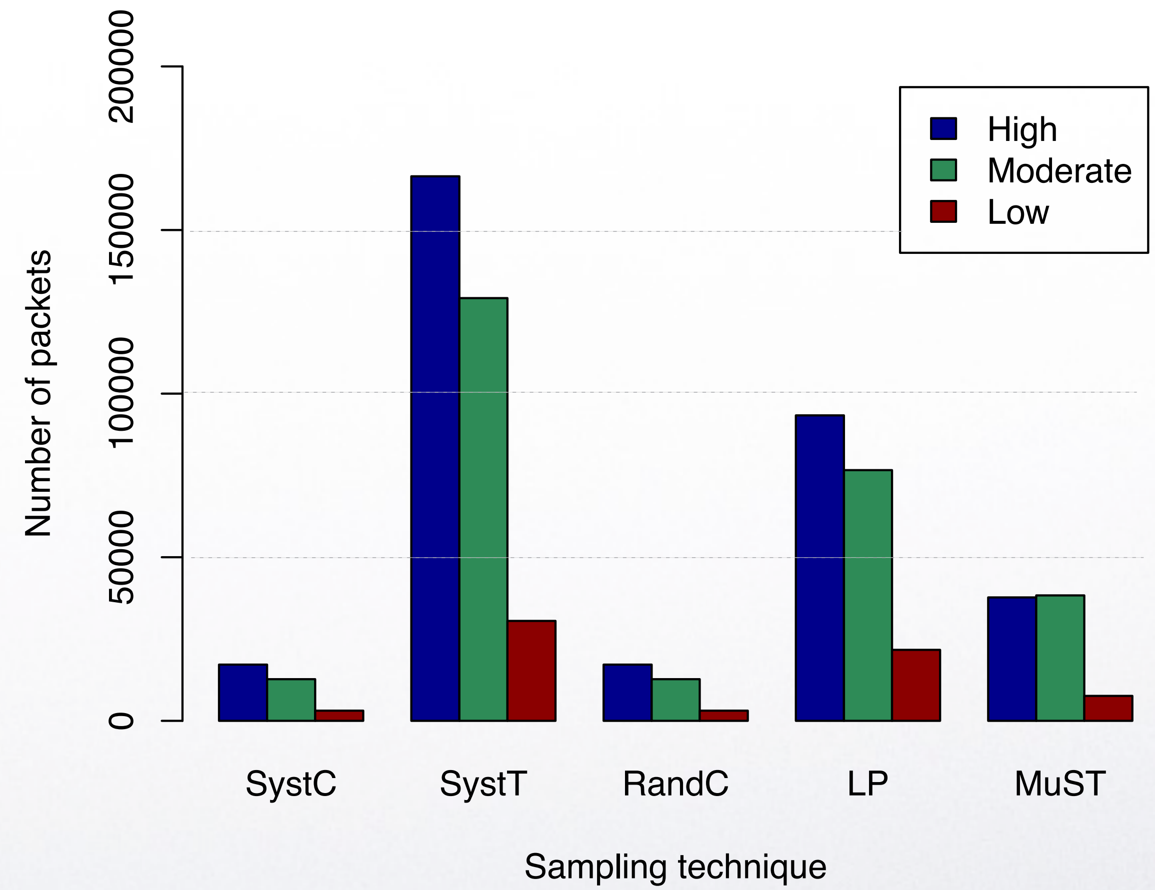
Parameter		SystC	SystT	RandC	LP	MuST
CPU load (%)	Low workload	5.03	14.55	5.50	27.35	8.82
Memory (kBytes)	Low workload	76566	95900	81222	82440	85295
CPU load (%)	Moderate workload	10.80	17.95	16.86	96.68	10.72
Memory (kBytes)	Moderate workload	80773	96410	84042	87698	84371
CPU load (%)	High workload	14.92	20.12	18.26	97.27	10.76
Memory (kBytes)	High workload	81801	90754	86163	85551	80765

CENTROALGORITMI

Early results

- Volume of data - all traffic scenarios





Ratio	SystC	SystT	RandC	LP	MuST
%CPU / MByte	1.40	0.20	1.73	1.71	0.45
%Memory / MByte	1.63	0.16	1.73	0.32	0.72

Early results

- Accuracy

Parameter		Total	SystC	SystT	RandC	LP	MuST
Mean throughput (Mbps) / MRE	Low workload	3.90	3.72 / 0.04	3.70 / 0.04	3.66 / 0.05	3.85 / 0.01	3.81 / 0.02
Mean pkt size (Bytes)	Low workload	377.58	387.87	375.65	371.39	390.32	386.70
Peak-to-average	Low workload	4.00	3.90	4.03	4.07	3.87	3.91
Mean throughput (Mbps) / MRE	Moderate workload	26.65	25.40 / 0.04	25.08 / 0.05	23.73 / 0.11	25.51 / 0.04	25.44 / 0.04
Mean pkt size (Bytes)	Moderate workload	587.26	586.90	579.42	586.53	589.38	587.82
Peak-to-average	Moderate workload	2.57	2.57	2.61	2.58	2.56	2.57
Mean throughput (Mbps) / MRE	High workload	68.79	65.54 / 0.04	64.06 / 0.06	65.05 / 0.05	64.28 / 0.06	68.47 / 0.004
Mean pkt size (Bytes)	High workload	648.59	647.95	633.33	643.15	635.49	652.36
Peak-to-average	High workload	2.33	2.33	2.39	2.35	2.38	2.32

Conclusions

- Preliminary results evince the relevance of tuning traffic sampling in order to meet distinct measurement needs and constraints
- Despite the extensive deployment of count-based techniques, the time-based approach achieves a better tradeoff volume of data / computational resources usage

Future work

- Deploy the adaptive controller module
  - reactive
    - threshold driven
    - fuzzy logic driven
  - proactive
    - linear prediction
    - nonlinear prediction

Universidade do Minho
Portugal

A modular architecture for deploying self-adaptive traffic sampling

João Marco C. Silva
Paulo M. Carvalho
Solange Rito Lima

CENTROALGORITMI