Performance Analysis of Time-Triggered Ether-Networks Using Off-The-Shelf-Components

AMICS Workshop 2011

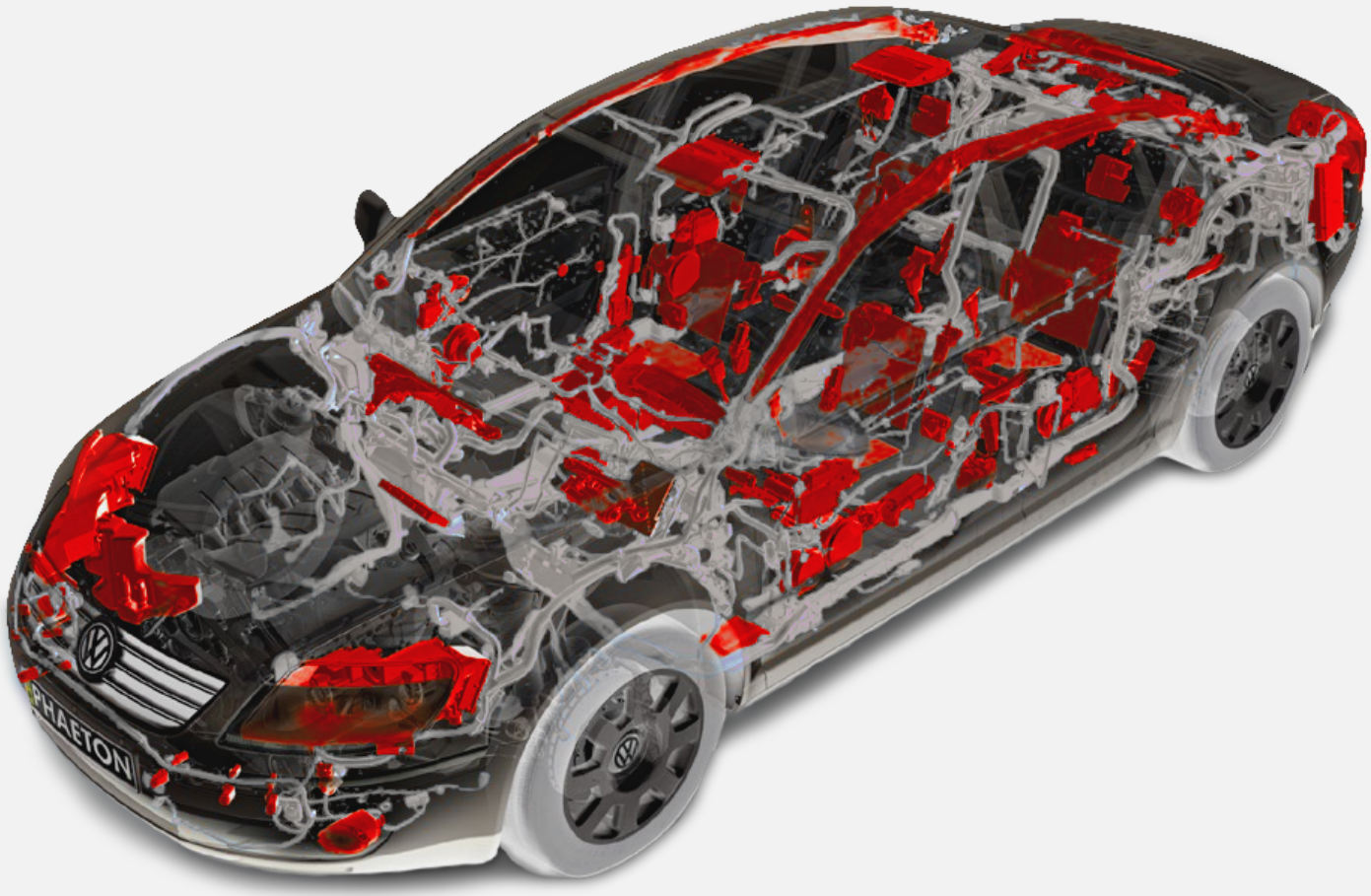
Florian Bartols, Till Steinbach, Franz Korf, Thomas C. Schmidt
{florian.bartols,till.steinbach,korf,schmidt}
@informatik.haw-hamburg.de

Hamburg University of Applied Sciences

March, 31st 2011

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Agenda


PHAETON

Real-time Ethernet in automotive applications:

- In-vehicle networks are very complex
- New automotive applications require more bandwidth
- Real-time Ethernet is widely deployed in backbones of industrial plants
- Comparable real-time characteristics in vehicles

Low cost performance analysis with time-triggered packet generation:

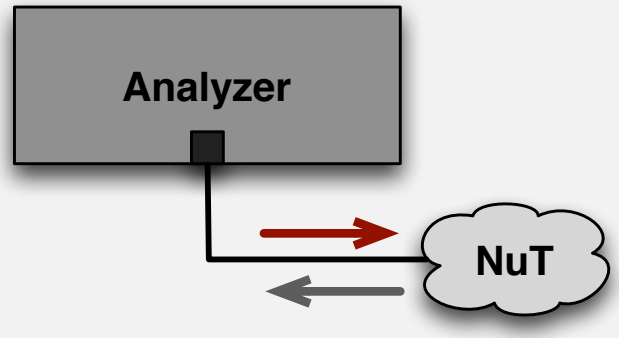
- Missing performance analyzer instruments in tool chain
- Tools for standard switched Ethernet are not suitable
- Flexible embedded pc based tools gain importance

Agenda

Distinction between software and hardware based measurements

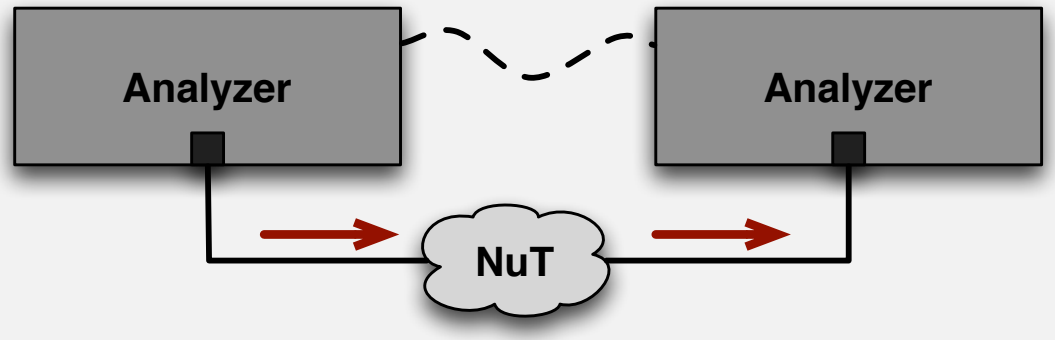
Distinction between software and hardware based measurements

- Collection on one network port only



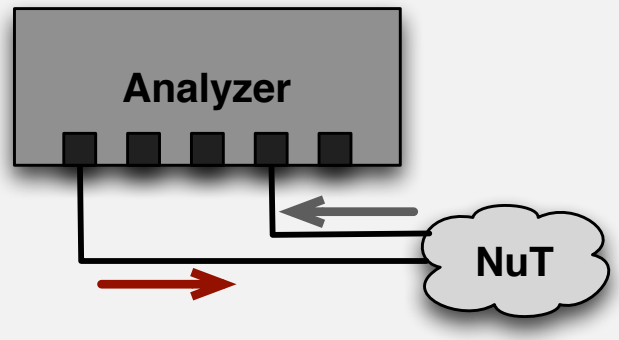
Distinction between software and hardware based measurements

- Collection on one network port only
- Distributed measurement with synchronized timebase



Distinction between software and hardware based measurements

- Collection on one network port only
- Distributed measurement with synchronized timebase
- One physical clock and several network ports



3 strategies to enable real-time characteristics on Ethernet-based networks

3 strategies to enable real-time characteristics on Ethernet-based networks

- Token-based systems
  - Used in Ethercat in process automation

3 strategies to enable real-time characteristics on Ethernet-based networks

- Token-based systems
  - Used in Ethercat in process automation
- Bandwidth limiting systems
  - For instance AFDX in Airplanes (A380 and Boeing 787)

3 strategies to enable real-time characteristics on Ethernet-based networks

- Token-based systems
  - Used in Ethercat in process automation
- Bandwidth limiting systems
  - For instance AFDX in Airplanes (A380 and Boeing 787)
- Time-triggered systems
  - Automotive industry (FlexRay) and process automation (Profinet)

3 Traffic Classes in TTEthernet

3 Traffic Classes in TTEthernet

1. Time-Triggered
   - Highest priority and used for hard real-time data
   - Strictest requirements in transmission

3 Traffic Classes in TTEthernet

1. Time-Triggered
   - Highest priority and used for hard real-time data
   - Strictest requirements in transmission
2. Rate-Constrained
   - Also suitable for real-time communication
   - Complies to the AFDX protocol

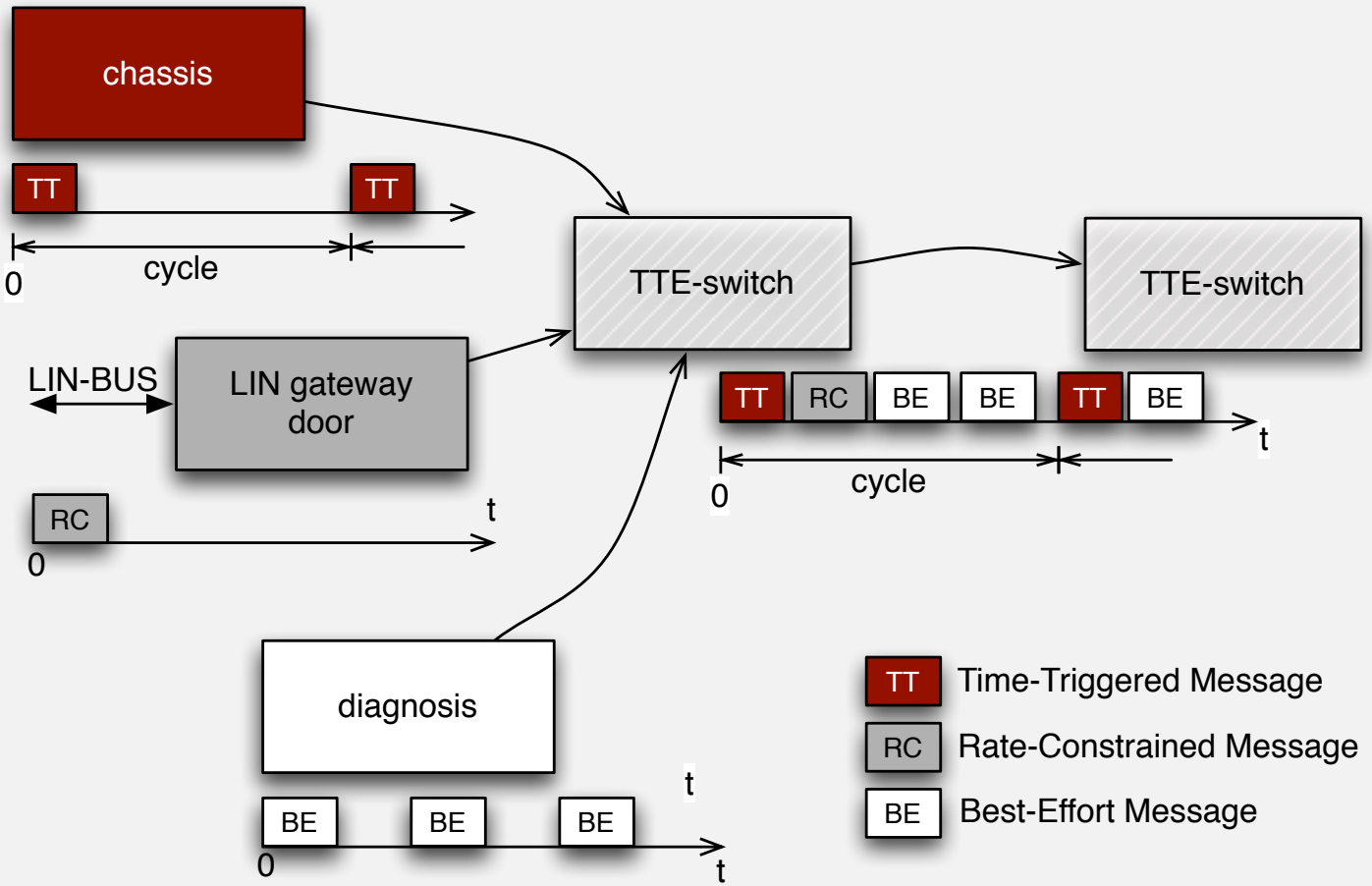
3 Traffic Classes in TTEthernet

1. Time-Triggered
   - Highest priority and used for hard real-time data
   - Strictest requirements in transmission
2. Rate-Constrained
   - Also suitable for real-time communication
   - Complies to the AFDX protocol
3. Best-Effort
   - Standard Ethernet traffic
   - No guarantee in transmission

3 Traffic Classes in TTEthernet

1. Time-Triggered
   - Highest priority and used for hard real-time data
   - Strictest requirements in transmission
2. Rate-Constrained
   - Also suitable for real-time communication
   - Complies to the AFDX protocol
3. Best-Effort
   - Standard Ethernet traffic
   - No guarantee in transmission

- Time synchronization for a global time base
- Works transparent to the Ethernet-Protocol-Layer


chassis
TT
TT
cycle
0
TTE-switch
TTE-switch
LIN-BUS
LIN gateway door
RC
0
t
TT
RC
BE
BE
TT
BE
t
0
cycle
diagnosis
BE
BE
BE
t
0
t
TT Time-Triggered Message
RC Rate-Constrained Message
BE Best-Effort Message

Agenda

1 Introduction

2 Related Work & Background

3 Implementing the Measurement Facility
- Concept
- Accessing the Current System Time
- Assigning Timestamps to Frames

4 Validation & Measurements

5 Conclusion & Outlook

Available OS tools:

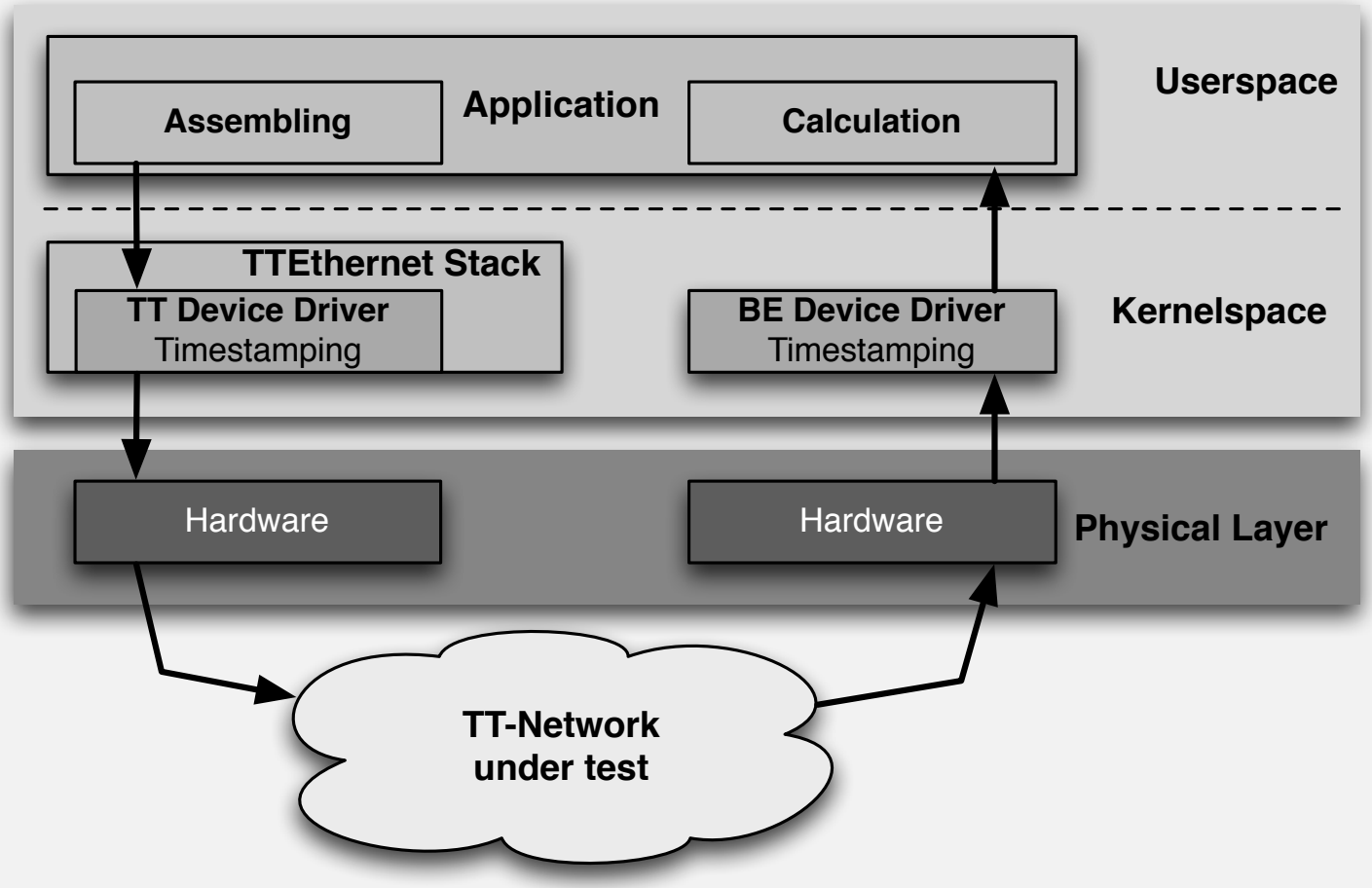
- Tools running in Userspace (e.g. ping, traceroute)
- Precision in milliseconds provided by OS
- Scheduling and hardware-layer produce overhead
- OS tools measure round-trip-time

Our approach:

- Measurement on a low-level base → Kernelspace in Linux
- RT-Linux utilization for enhanced interrupt handling
- Exact timing behavior by using special functions

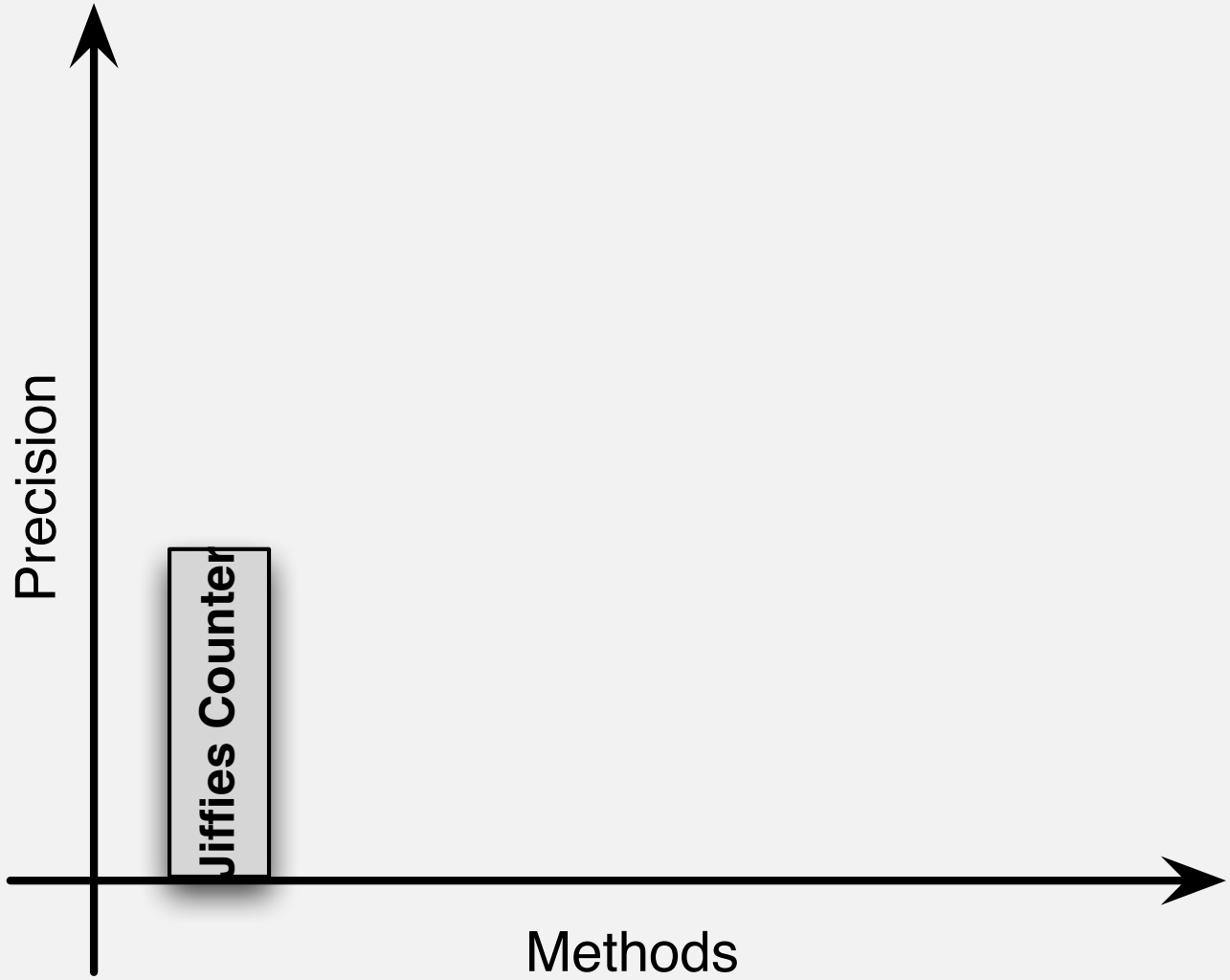
Implementing the Measurement Facility

Concept



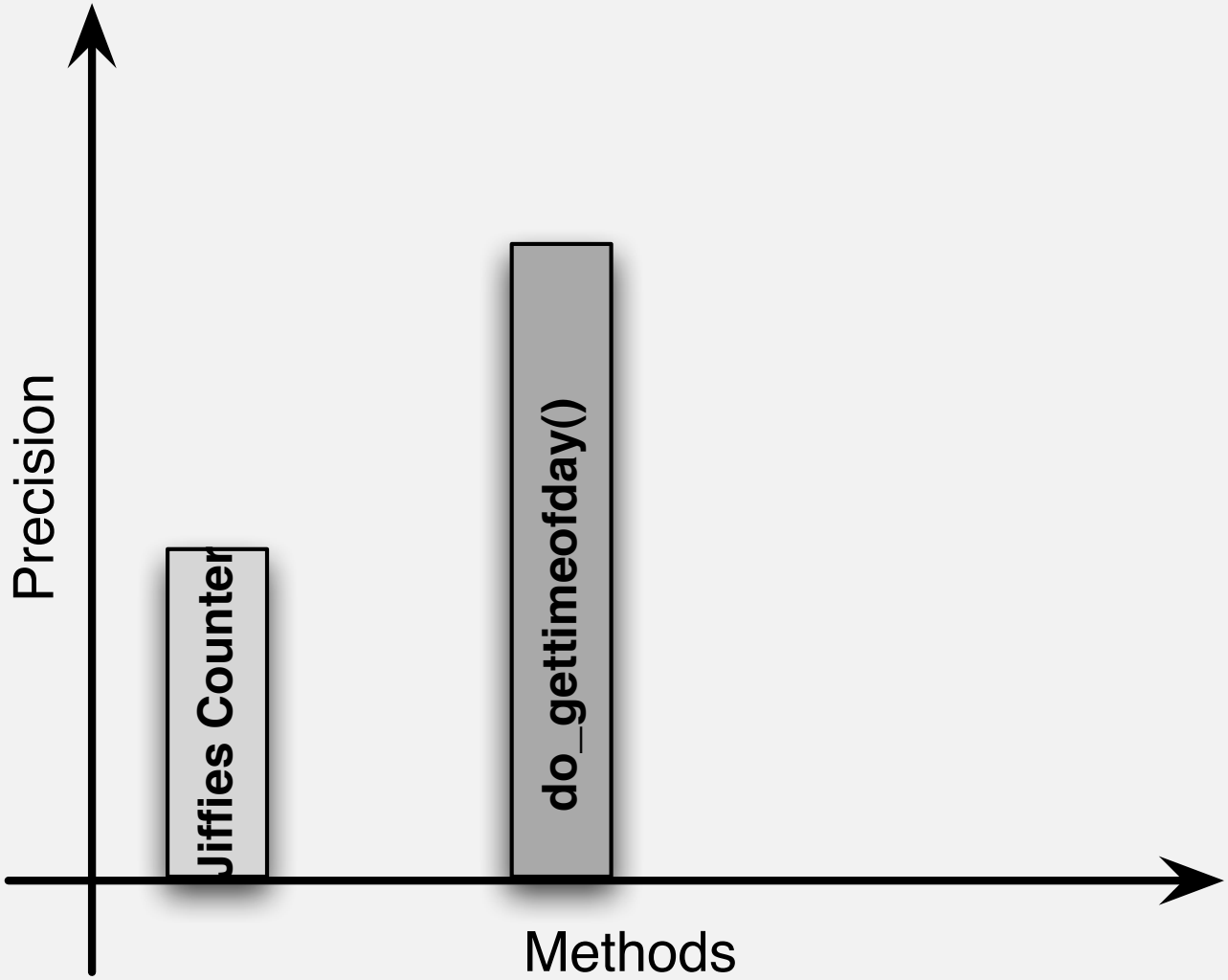
Implementing the Measurement Facility

Accessing the Current System Time



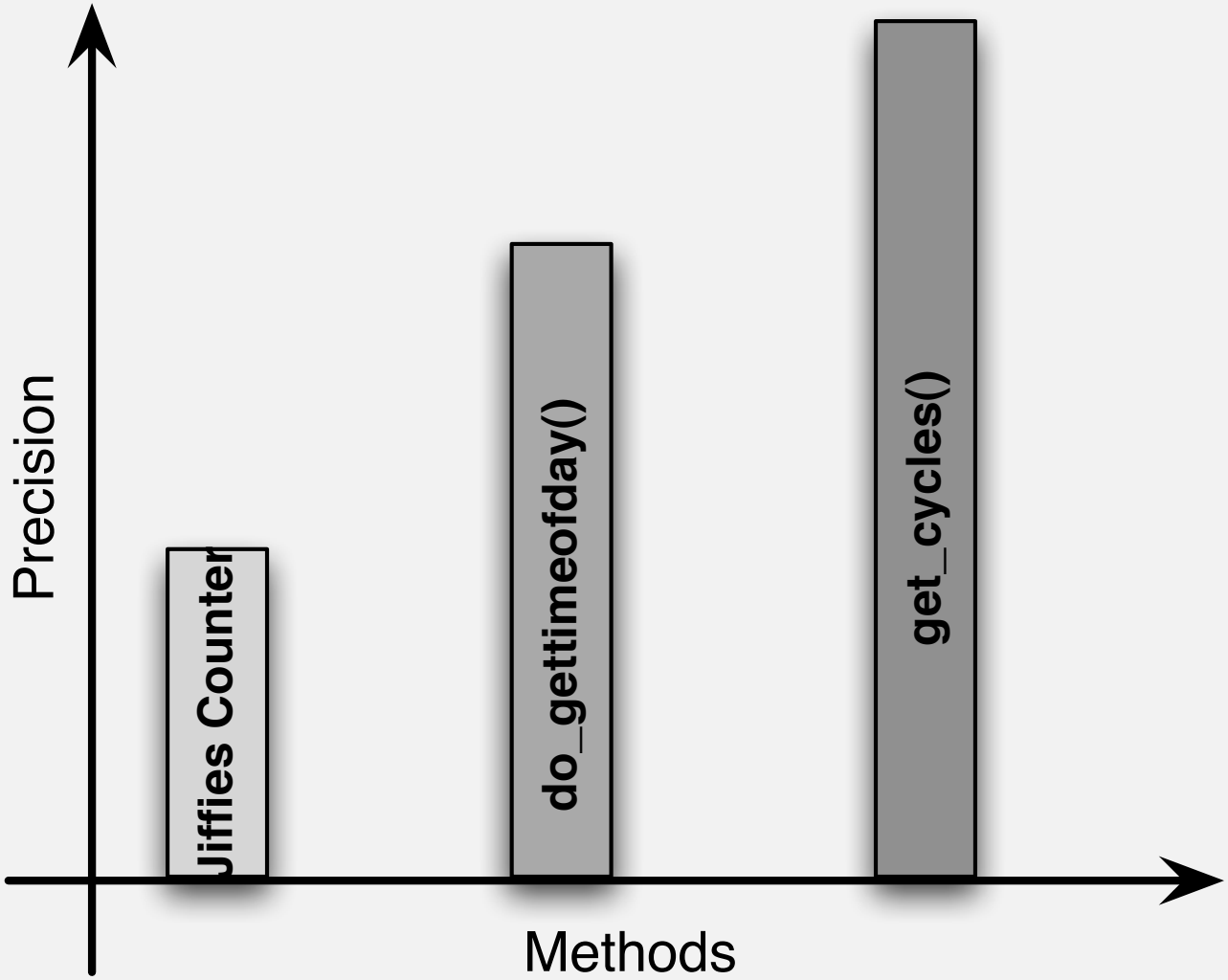
Implementing the Measurement Facility

Accessing the Current System Time



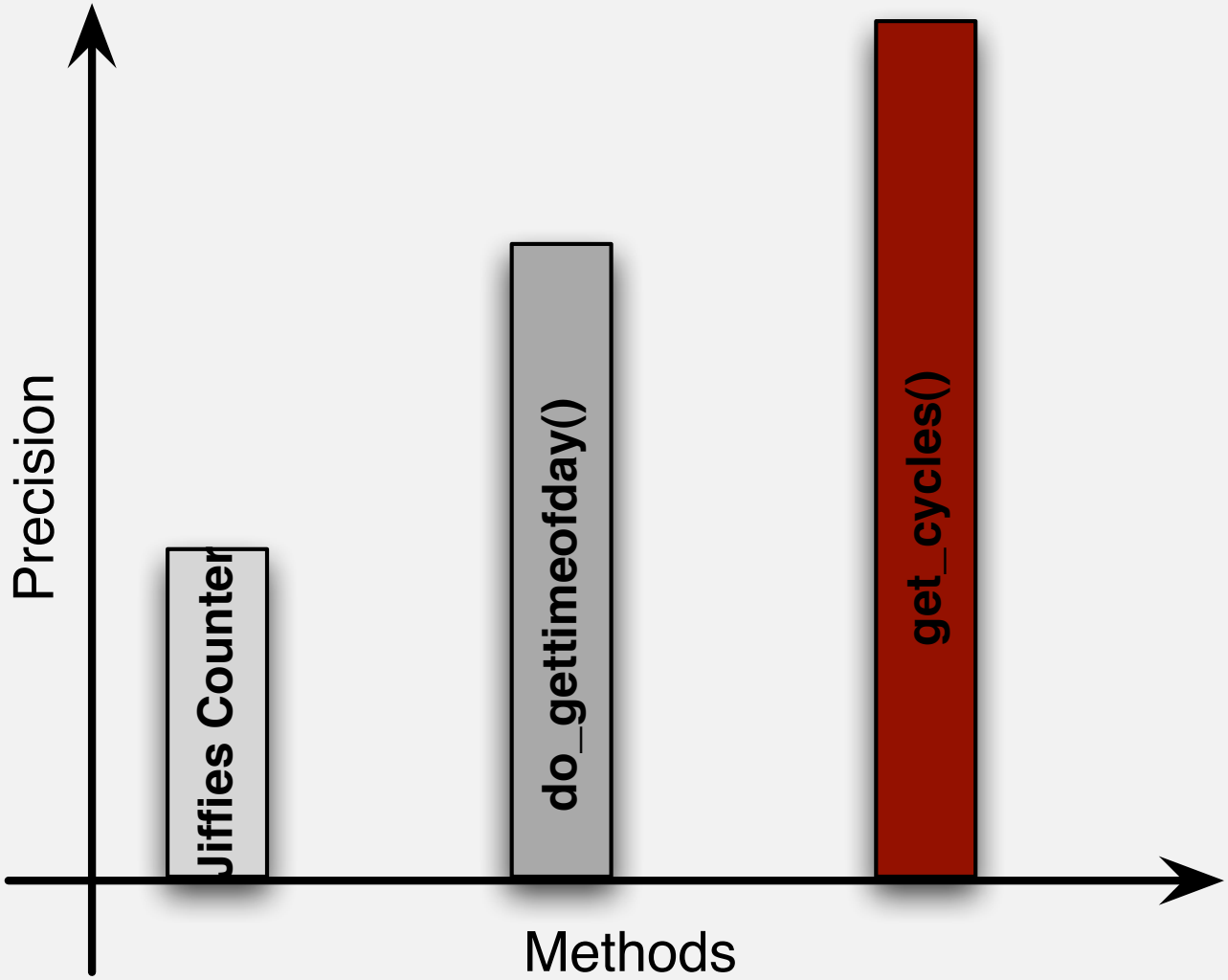
Implementing the Measurement Facility

Accessing the Current System Time



Implementing the Measurement Facility

Accessing the Current System Time

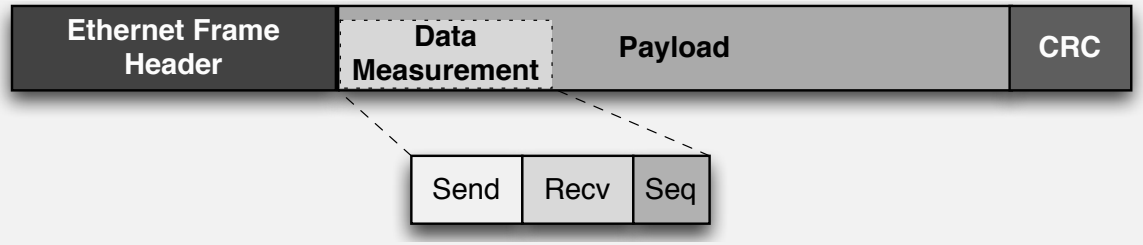


2 approaches to get timing information of a single frame

1. Storing information in Kernelspace
   - High implementation cost (limited Kernel functions)
   - Additional information has be stored

2 approaches to get timing information of a single frame

1. Storing information in Kernelspace
   - High implementation cost (limited Kernel functions)
   - Additional information has be stored
2. Modifying Ethernet frames with timestamps
   - A lightweight approach
   - Assigning timestamps to frames in an easy way
   - Measurement can be done promptly without reading and storing information



Agenda

1 Introduction

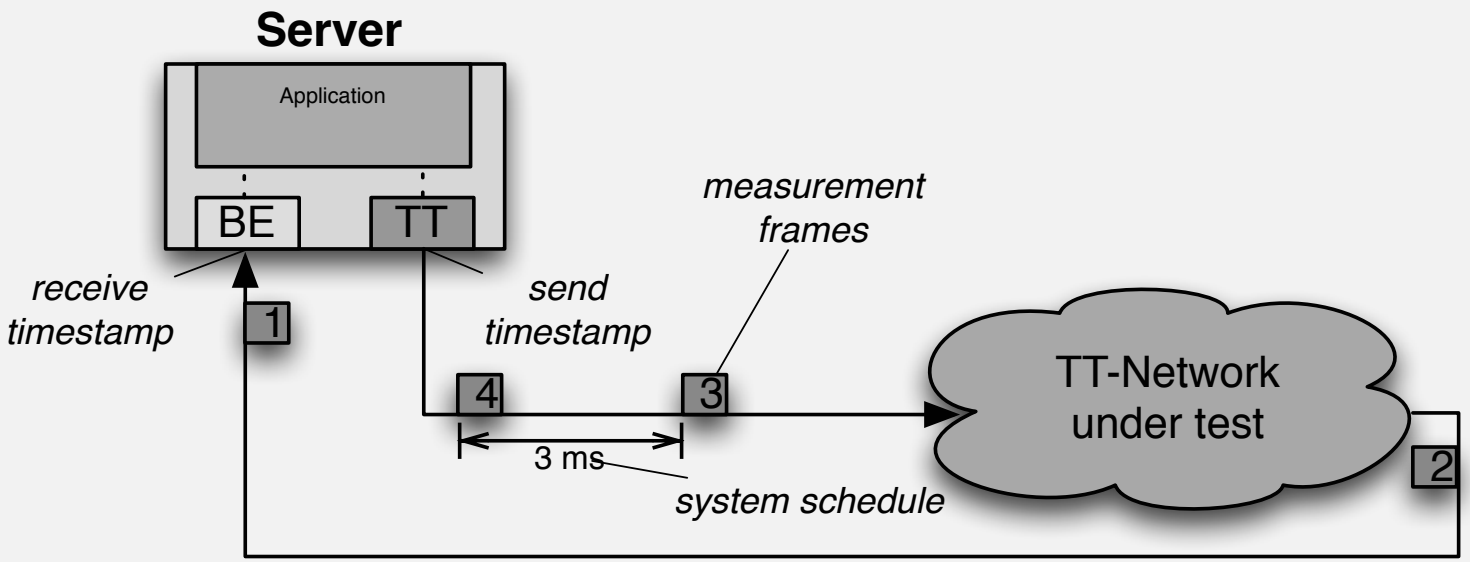
2 Related Work & Background

3 Implementing the Measurement Facility

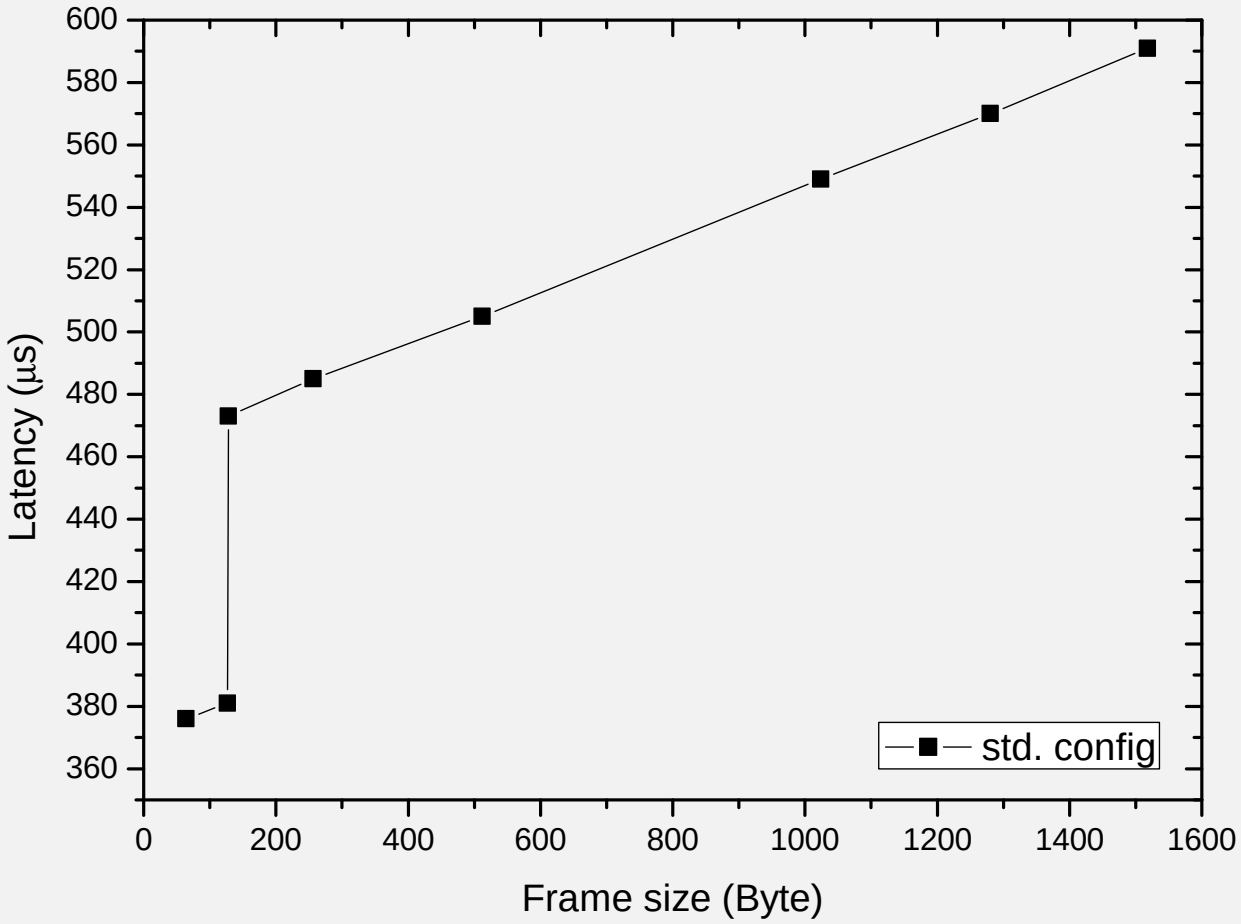
4 Validation & Measurements
- Validating the Approach
- Measurement Results

5 Conclusion & Outlook

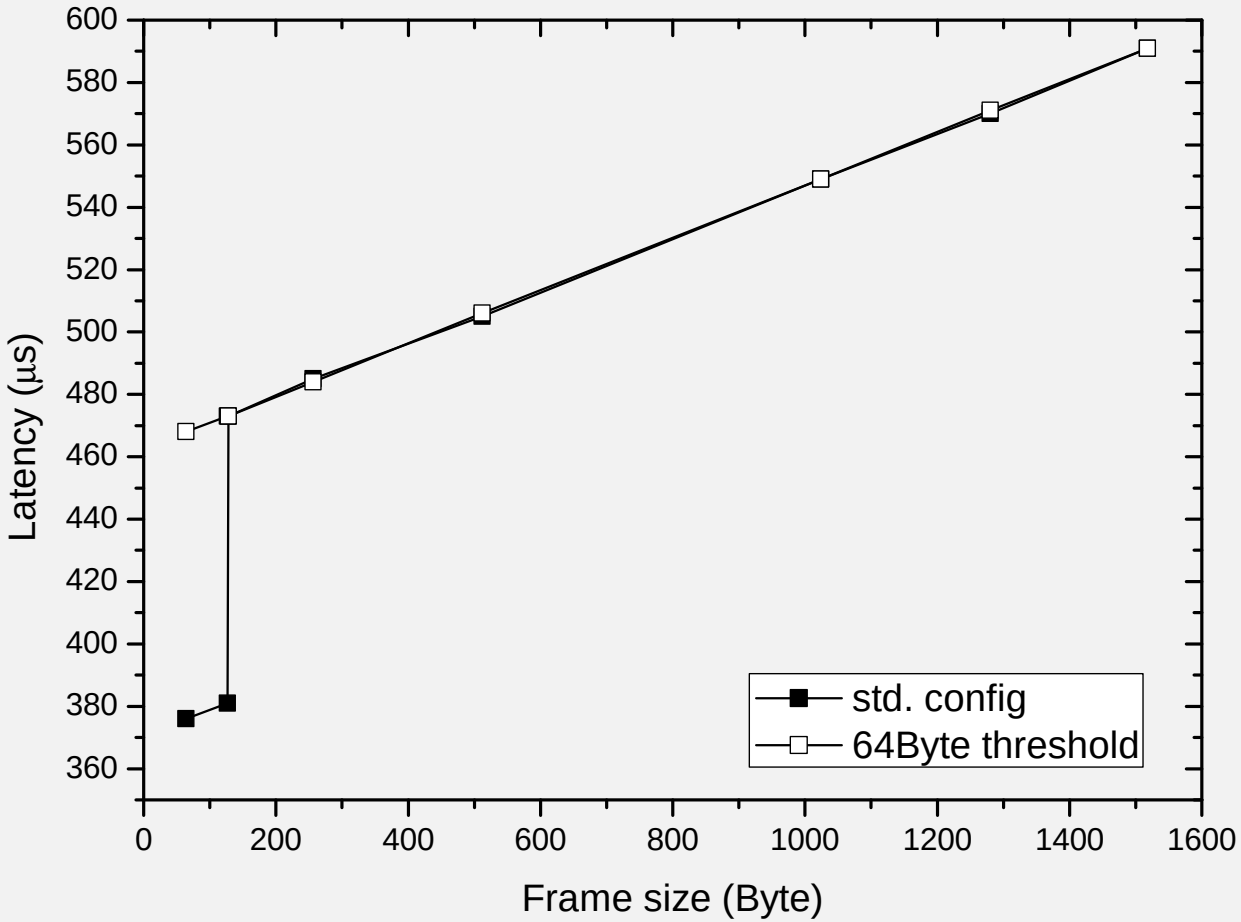
- 3ms System Schedule for each TT-Frame
- 350μs static switch delay



Validation
Sample Measurement

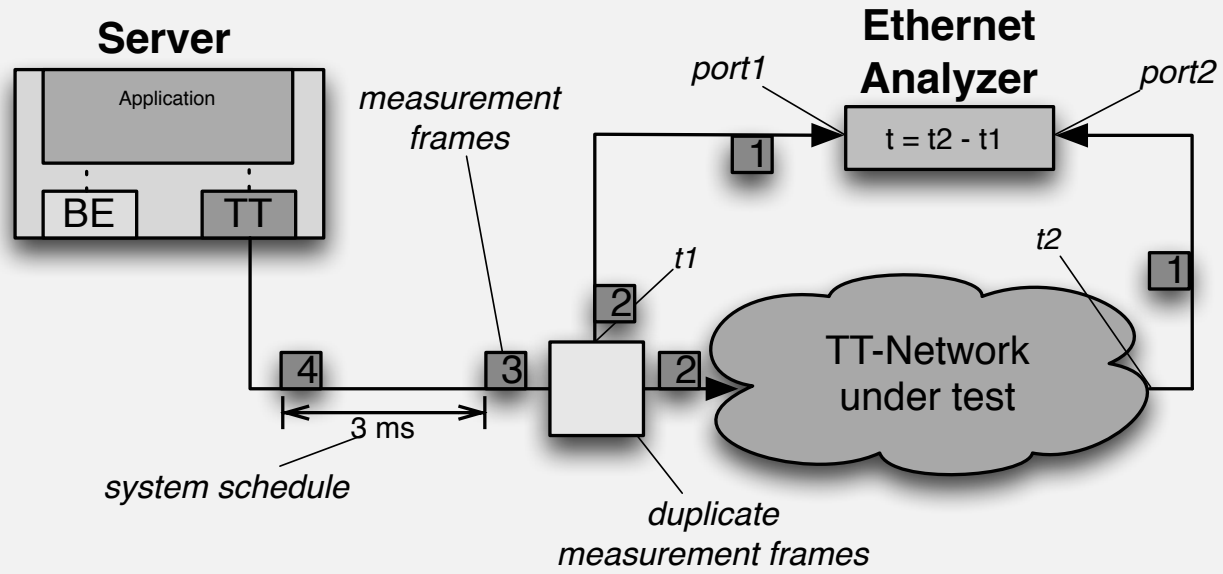


Validation
Sample Measurement

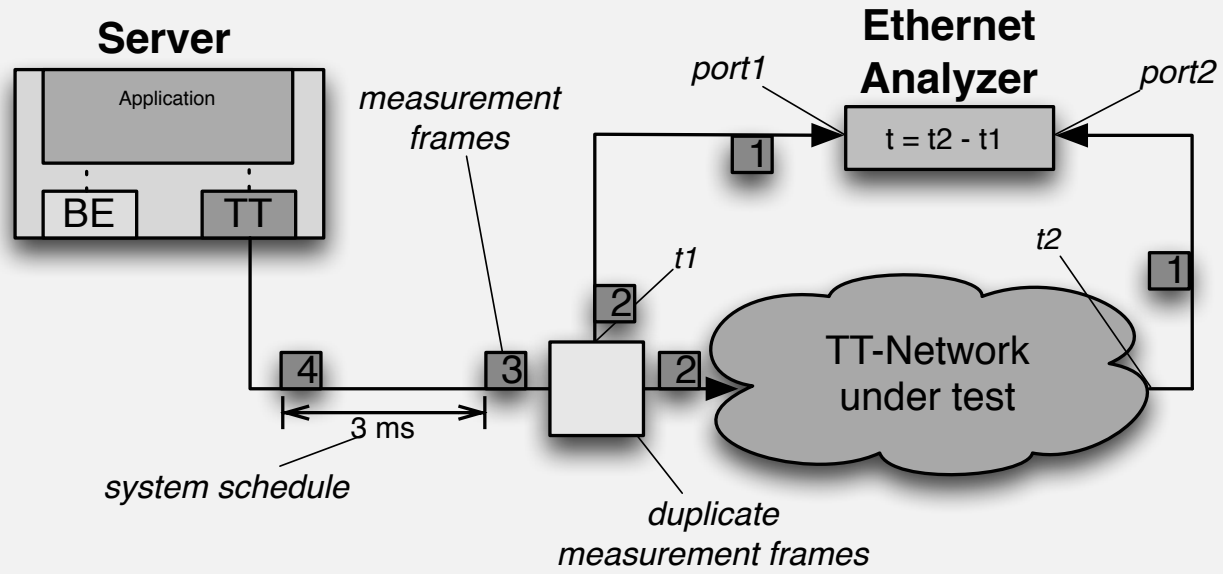


- Observation: A specific hardware and driver delay has to be substracted for each measurment

1 High precision Ethernet analyzer tool



1. High precision Ethernet analyzer tool



2. Mathematical framework
   - Analytical method to calculate latency, bandwidth and jitter

Validation
Results in Comparison

	measurement approach	hardware sniffing	mathematical model
min. framelength	355μs	364μs	355.125μs
max. framelength	471μs	479μs	471.445μs

Validation
Results in Comparison

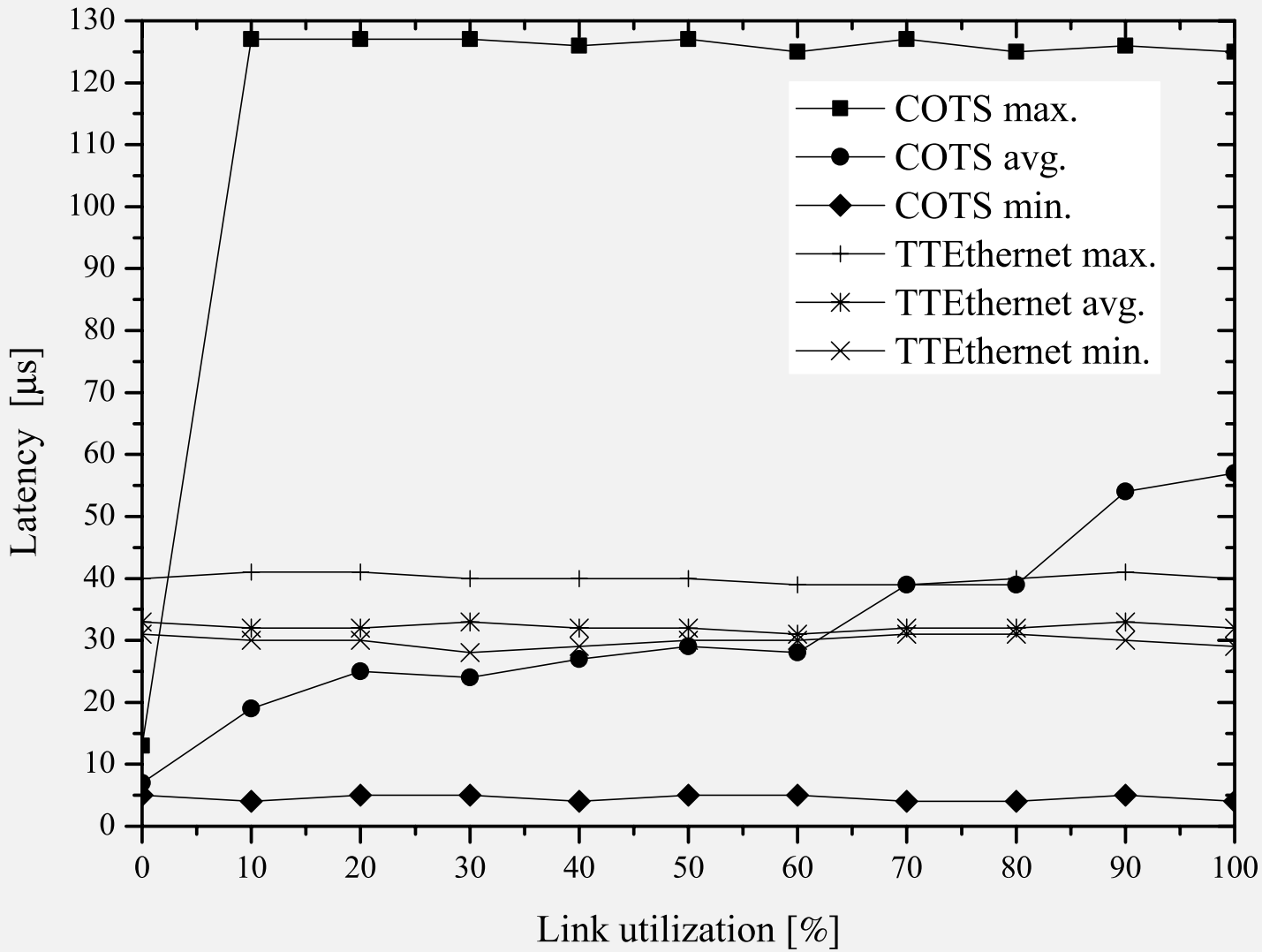
	measurement approach	hardware sniffing	mathematical model
min. framelength	355μs	364μs	355.125μs
max. framelength	471μs	479μs	471.445μs

Difference of 9μs is due to the used duplication switch

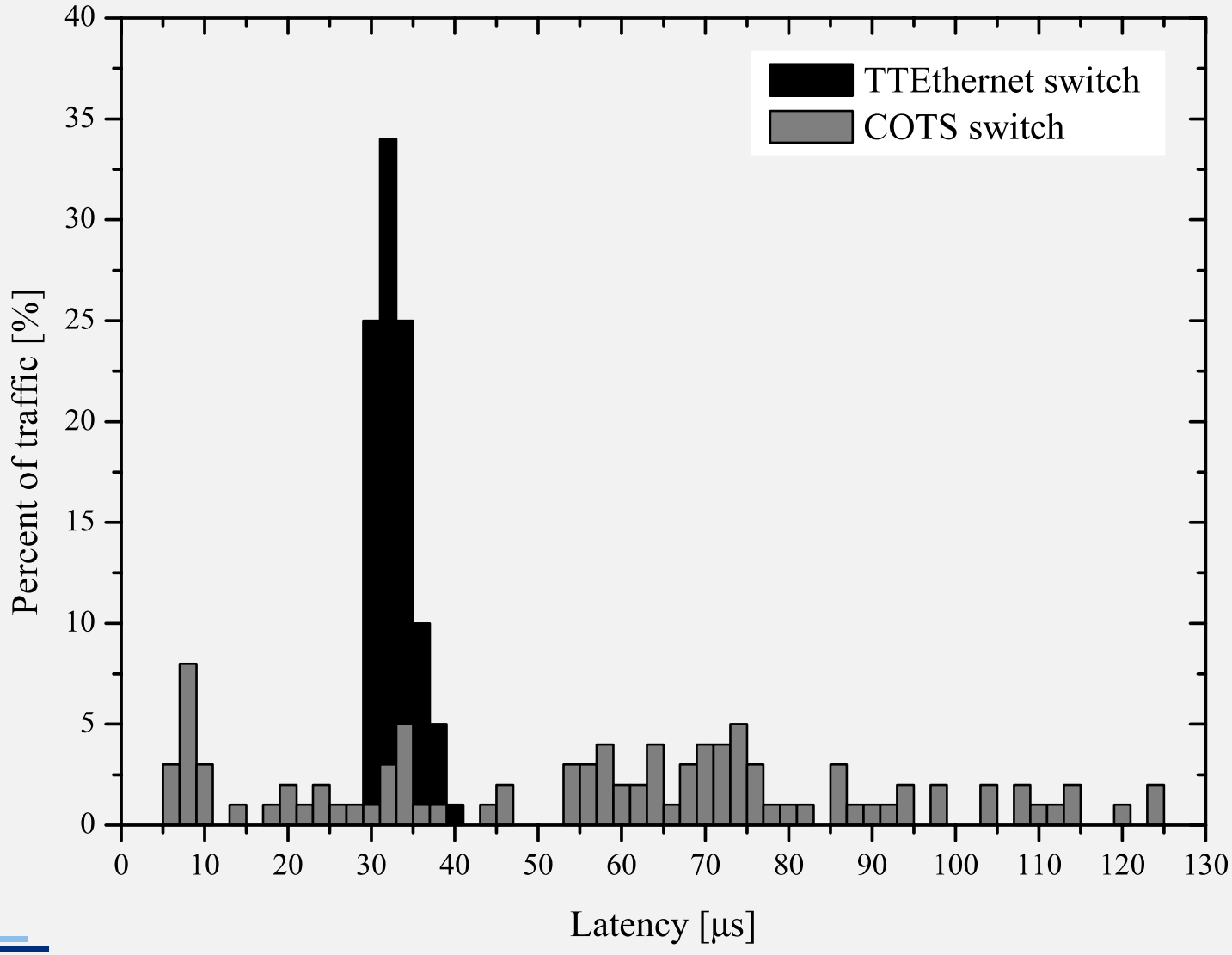
	measurement approach	hardware sniffing	mathematical model
min. framelength	355μs	355μs	355.125μs
max. framelength	471μs	470μs	471.445μs

Measurements

Latency Comparison COTS switch and TTEthernet Switch



Measurements
Latency Distribution Graph



Agenda

Conclusion & Outlook

Conclusion:

- Performance analysis with COTS in single-digit microseconds precision
- Synchronized packet generation
- Utilization of RT-Linux, modified device drivers and TTEthernetstack

Conclusion & Outlook

Conclusion:

- Performance analysis with COTS in single-digit microseconds precision
- Synchronized packet generation
- Utilization of RT-Linux, modified device drivers and TTEthernetstack

Outlook:

- Validation using hardware with specifically low receive and copy delay
- Adjustment of this approach on an ARM-based micro controller to increase resolution

Thank you!

Thank you for your attention!

- Website of research group:
  http://www.informatik.haw-hamburg.de/core.html