LARN
Latency- and Resilience-Aware Networking

Latency- and Resilience-Aware Networking

SPP 1914: "Cyber-Physical Networking"

http://larn.systems

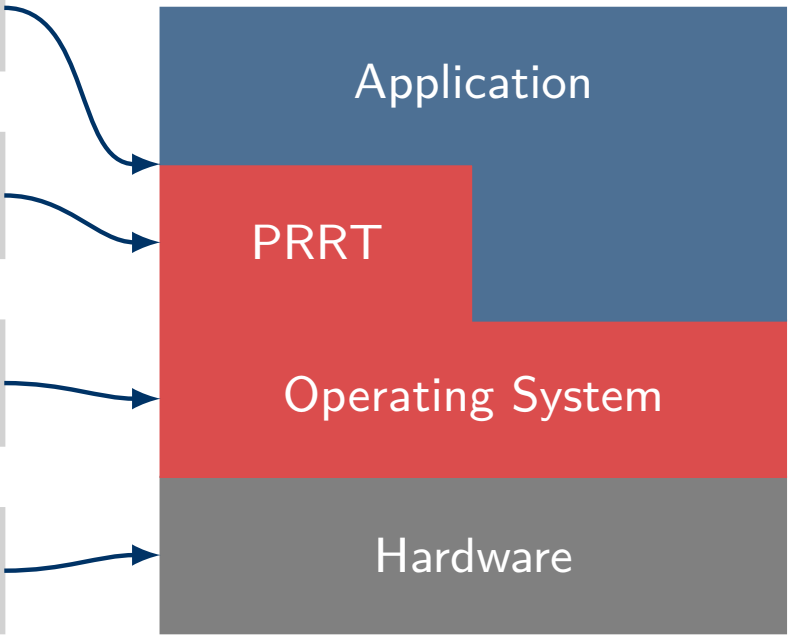
Andreas Schmidt, Thorsten Herfet
Telecommunications Lab
Saarland Informatics Campus - Saarbrücken

Stefan Reif, Timo Hönig, Wolfgang Schröder-Preikschat
Department of Computer Science 4 (Distributed Systems and Operating Systems)
Friedrich-Alexander-Universität Erlangen-Nürnberg

October 19th, 2017

Recapitulation: Goals

- Tailor PRRT to control applications
- Improve PRRT's performance
- OS for predictable communication
- Tailor network to control applications



LARN

Software, Hardware & Algorithms

X-Lap Cross-Layer Timing Analysis

LARN

Software, Hardware & Algorithms

X-Lap Cross-Layer Timing Analysis

PRRT Predictable Reliable Real-time Transport protocol

- Latency & Jitter Analysis (with X-Lap)
- APIs / Integrations: Python, Gstreamer
- Hardware Timestamping Support

LARN

Software, Hardware & Algorithms

X-Lap	Cross-Layer Timing Analysis
PRRT	Predictable Reliable Real-time Transport protocol
	- Latency & Jitter Analysis (with X-Lap)
	- APIs / Integrations: Python, Gstreamer
	- Hardware Timestamping Support
TTS	Transparent Transmission Segmentation
	- TCP Relay, RTP Relay

LARN

Software, Hardware & Algorithms

X-Lap	Cross-Layer Timing Analysis
PRRT	Predictable Reliable Real-time Transport protocol
	- Latency & Jitter Analysis (with X-Lap)
	- APIs / Integrations: Python, Gstreamer
	- Hardware Timestamping Support
TTS	Transparent Transmission Segmentation
	- TCP Relay, RTP Relay
RNA	Reliable Networking Atom
	- Linux Image & Orchestration Code (including PRRT, ...)
	- Showcase Applications (Video-Stream, Sine-Controller)

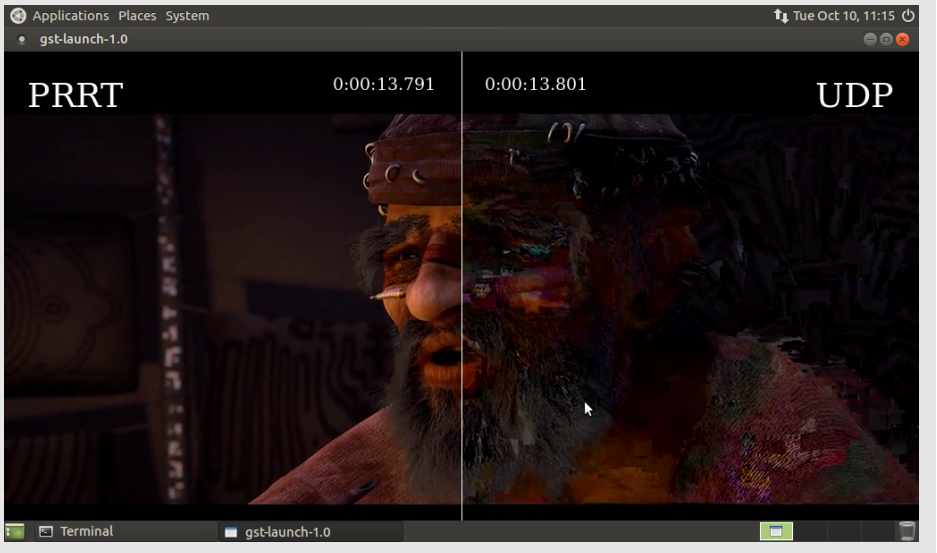
LARN

Software, Hardware & Algorithms

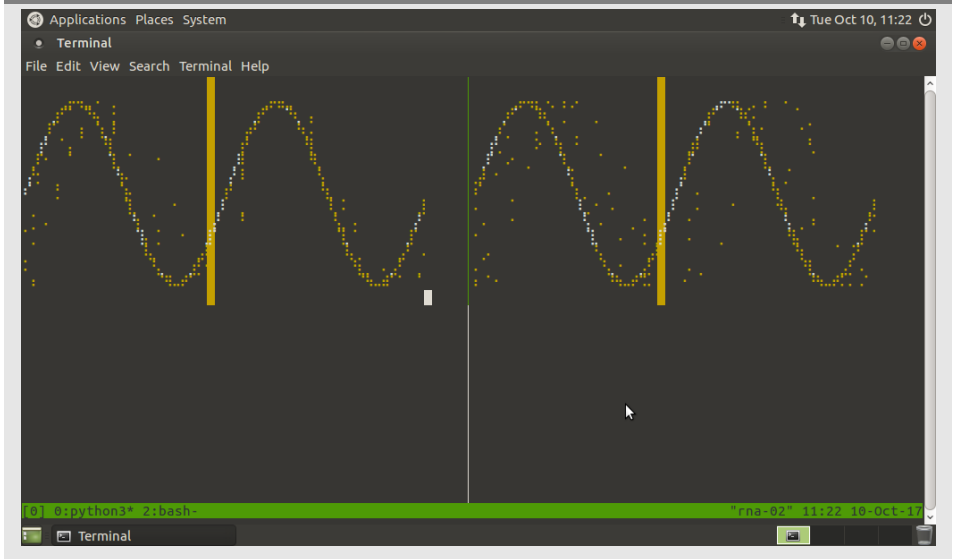
X-Lap	Cross-Layer Timing Analysis
PRRT	Predictable Reliable Real-time Transport protocol
	- Latency & Jitter Analysis (with X-Lap)
	- APIs / Integrations: Python, Gstreamer
	- Hardware Timestamping Support
TTS	Transparent Transmission Segmentation
	- TCP Relay, RTP Relay
RNA	Reliable Networking Atom
	- Linux Image & Orchestration Code (including PRRT, ...)
	- Showcase Applications (Video-Stream, Sine-Controller)
NEAT	Network Experiment Automation Tool

Outline

Video-Stream



Control Application



Hardware

- **Odroid XU4**
  - Samsung Exynos5422 Cortex-A15 2Ghz and Cortex-A7 **Octa core CPUs**
  - Mali-T628 MP6 (**OpenGL** ES 3.1/2.0/1.1 and **OpenCL** 1.2 Full profile)
  - **2Gbyte** LPDDR3 RAM PoP stacked
  - **Gigabit Ethernet** port
- **VU7 Plus Display** (7inches, 1024x600 pixels, HDMI, 5-finger touch)

Software

- **Ubuntu 16.04.3 LTS**, Kernel 4.9
- Mate Desktop (for receiver/display unit)
- **PRRT** (Git: `http://prrt.larn.systems`)
- **RNA-Master VM** (DNS, DHCP, Orchestration, PTP, IP gateway, ...)

LARN

Sender

```
gst-launch-1.0 filesrc location=~/Video.mp4 \
    ! typefind \
    ! queue \
    ! prrtsink host=10.8.0.102 port=5000
```

Receiver

```
gst-launch-1.0 prrtsource port=5000 \
    ! queue ! decodebin \
    ! queue ! videoconvert \
    ! queue ! ximagesink
```

LARN

Sender (Controller)

```
import time
import prrt
import controller

s = prrt.PrrtSocket(port=6000,
     isSender=True)
s.target_delay = 50 * 1000 # us
s.connect("10.8.0.102", 5000)

while True:
  x = time.time()
  y = controller.control(x)
  s.send((x,y))
```

Receiver (Plant)

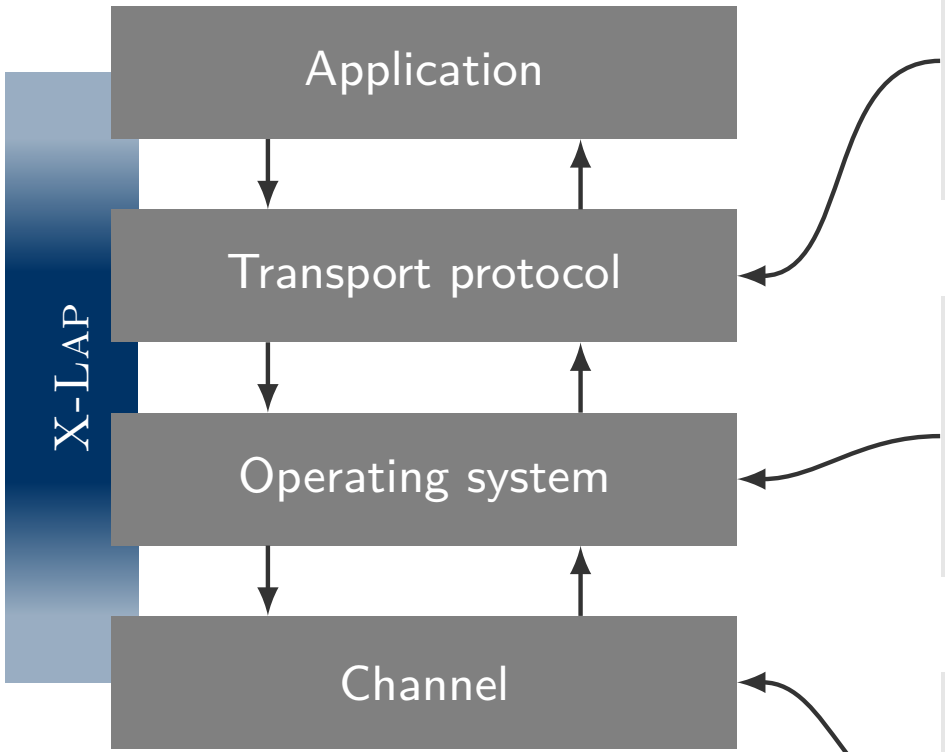
```
import prrt
import plant

s = prrt.PrrtSocket(port=5000,
     isSender=False)

while True:
    x,y = s.recv()
    plant.act(x,y)
```

LARN

Idea: Identify root causes for latency and jitter in PRRT.

LARN

Idea: Identify root causes for latency and jitter in PRRT.



- Packet creation
- Error correction
- ...

- Inter-process communication (IPC)
- Resource management
- ...

- Transmission delay
- Propagation delay
- ...

LARN

Run-time evaluation

- Timestamping functions
- $\Rightarrow$ Timestamps
- C code

Off-line analysis

- Data analysis
- $\Rightarrow$ Latency and jitter
- Python code

LARN

Run-time evaluation

- Timestamping functions
- $\Rightarrow$ Timestamps
- C code

- Trace every packet
- Minimize run-time interference
- Embedded into protocol source code

Off-line analysis

- Data analysis
- $\Rightarrow$ Latency and jitter
- Python code

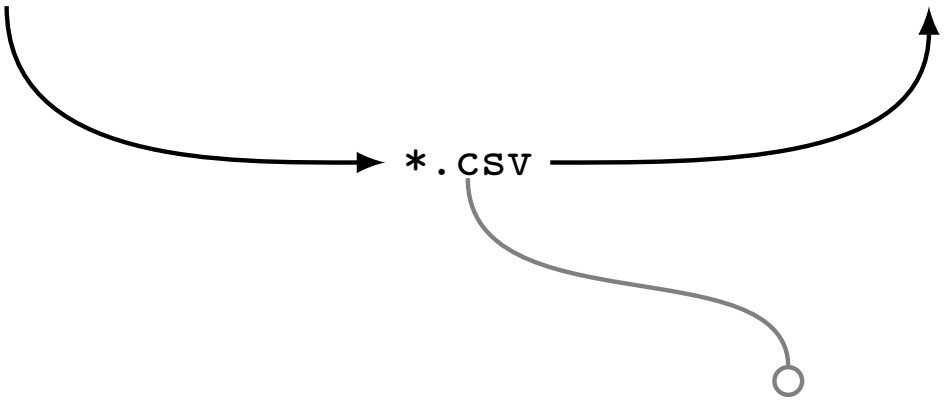
LARN

Run-time evaluation

- Timestamping functions
- ⇒ Timestamps
- C code

Off-line analysis

- Data analysis
- ⇒ Latency and jitter
- Python code



- Collect traces
- Combine trace data

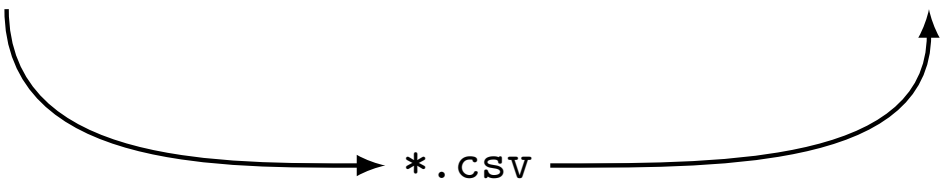
LARN

Run-time evaluation

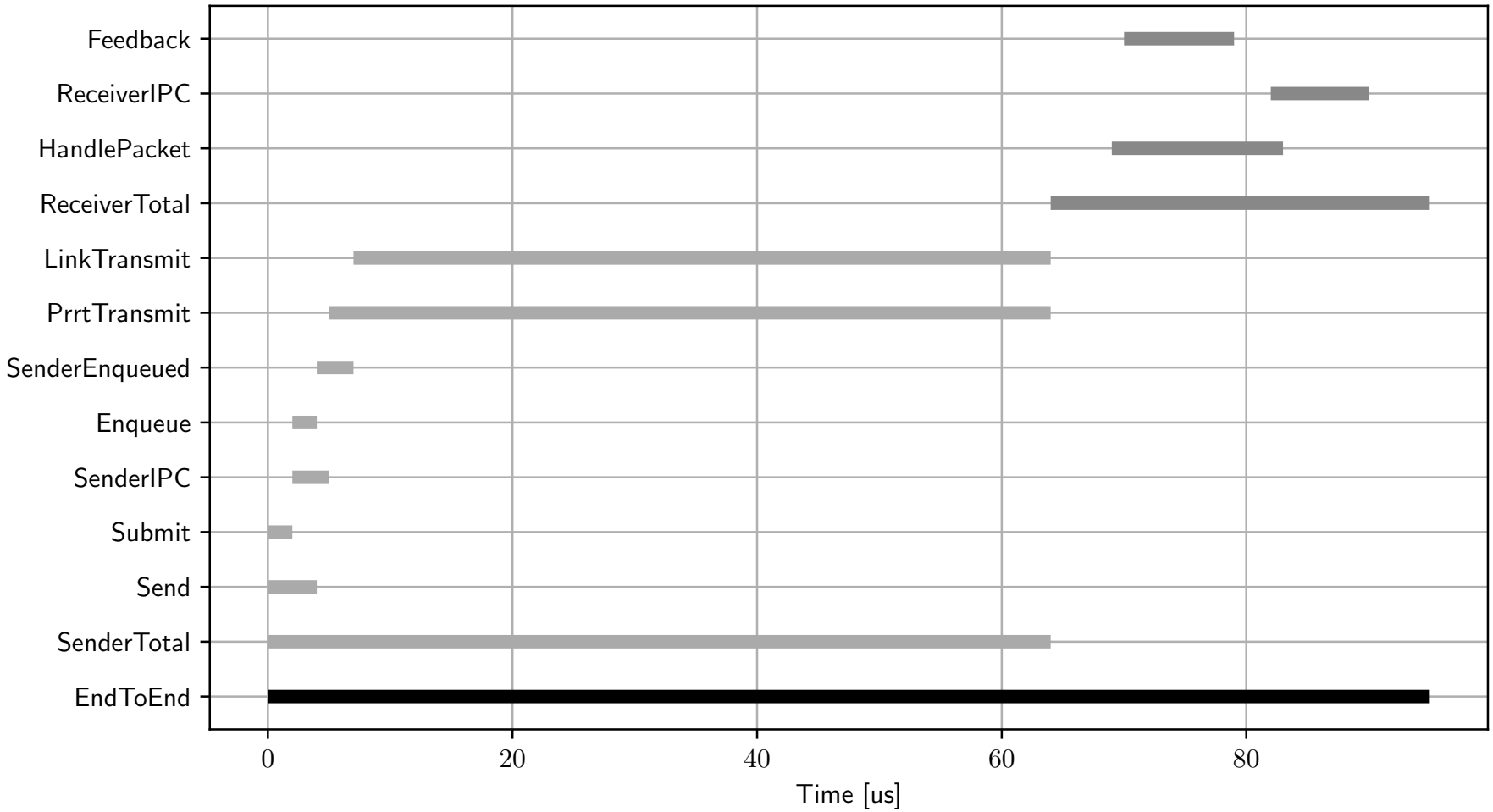
- Timestamping functions
- ⇒ Timestamps
- C code

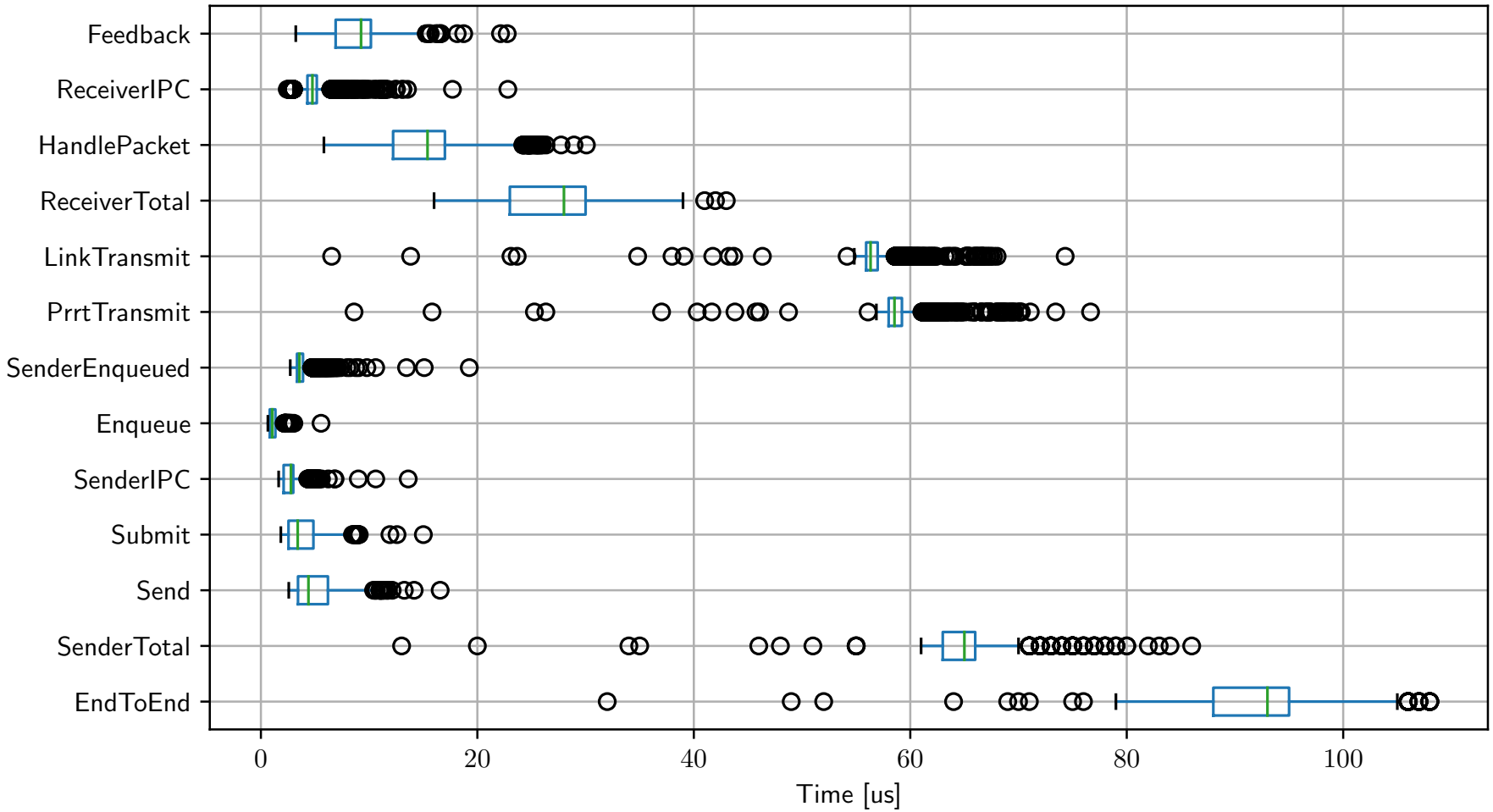
Off-line analysis

- Data analysis
- ⇒ Latency and jitter
- Python code



- Single-packet traces
- Jitter amongst packets
- Outlier analysis
- Correlation analysis


Feedback
ReceiverIPC
HandlePacket
ReceiverTotal
LinkTransmit
PrrtTransmit
SenderEnqueued
Enqueue
SenderIPC
Submit
Send
SenderTotal
EndToEnd
0
20
40
60
80
Time [us]


Feedback
ReceiverIPC
HandlePacket
ReceiverTotal
LinkTransmit
PrrtTransmit
SenderEnqueued
Enqueue
SenderIPC
Submit
Send
SenderTotal
EndToEnd
0
20
40
60
80
100
Time [us]

Publications

- Reif, Stefan; Hönig, Timo; Schröder-Preikschat, Wolfgang: **"In the Heat of Conflict: On the Synchronisation of Critical Sections"**, IEEE International Symposium on Real-Time Distributed Computing (ISORC), Toronto, Canada, May 2017

Publications

- Reif, Stefan; Hönig, Timo; Schröder-Preikschat, Wolfgang: **"In the Heat of Conflict: On the Synchronisation of Critical Sections"**, IEEE International Symposium on Real-Time Distributed Computing (ISORC), Toronto, Canada, May 2017
- Reif, Stefan; Schmidt, Andreas; Hönig, Timo; Herfet, Thorsten; Schröder-Preikschat, Wolfgang: **"X-Lap: A Systems Approach for Cross-Layer Profiling and Latency Analysis for Cyber-Physical Networks"**, 15th International Workshop on Real-Time Networks (ECRTS RTN), Dubrovnic, Croatia, June 2017

Publications

- Reif, Stefan; Hönig, Timo; Schröder-Preikschat, Wolfgang: **"In the Heat of Conflict: On the Synchronisation of Critical Sections"**, IEEE International Symposium on Real-Time Distributed Computing (ISORC), Toronto, Canada, May 2017
- Reif, Stefan; Schmidt, Andreas; Hönig, Timo; Herfet, Thorsten; Schröder-Preikschat, Wolfgang: **"X-Lap: A Systems Approach for Cross-Layer Profiling and Latency Analysis for Cyber-Physical Networks"**, 15th International Workshop on Real-Time Networks (ECRTS RTN), Dubrovnic, Croatia, June 2017
- Schmidt, Andreas; Herfet, Thorsten: **"Transparent Transmission Segmentation in Software-Defined Networks"**, IEEE Conference on Network Softwarization (NetSoft), Bologna, Italy, July 2017

Publications

- Reif, Stefan; Hönig, Timo; Schröder-Preikschat, Wolfgang: **"In the Heat of Conflict: On the Synchronisation of Critical Sections"**, IEEE International Symposium on Real-Time Distributed Computing (ISORC), Toronto, Canada, May 2017
- Reif, Stefan; Schmidt, Andreas; Hönig, Timo; Herfet, Thorsten; Schröder-Preikschat, Wolfgang: **"X-Lap: A Systems Approach for Cross-Layer Profiling and Latency Analysis for Cyber-Physical Networks"**, 15th International Workshop on Real-Time Networks (ECRTS RTN), Dubrovnic, Croatia, June 2017
- Schmidt, Andreas; Herfet, Thorsten: **"Transparent Transmission Segmentation in Software-Defined Networks"**, IEEE Conference on Network Softwarization (NetSoft), Bologna, Italy, July 2017
- Schmidt, Andreas; Herfet, Thorsten: **"NEAT: Network Experiment Automation Tool"**, 1. KuVS Fachgespräch "Network Softwarization"(KuVS-FG-NetSoft), Tübingen, Germany, October 2017

LARN

RNA

- Advanced control applications
- Wireless communications

LARN

RNA

- Advanced control applications
- Wireless communications

X-Lap

- Hardware timestamping

LARN

RNA

- Advanced control applications
- Wireless communications

X-Lap

- Hardware timestamping

PRRT

- Congestion control and delivery rate estimation
- Optimized error control for embedded platforms
- Python API: Specify more application parameters

LARN

RNA

- Advanced control applications
- Wireless communications

X-Lap

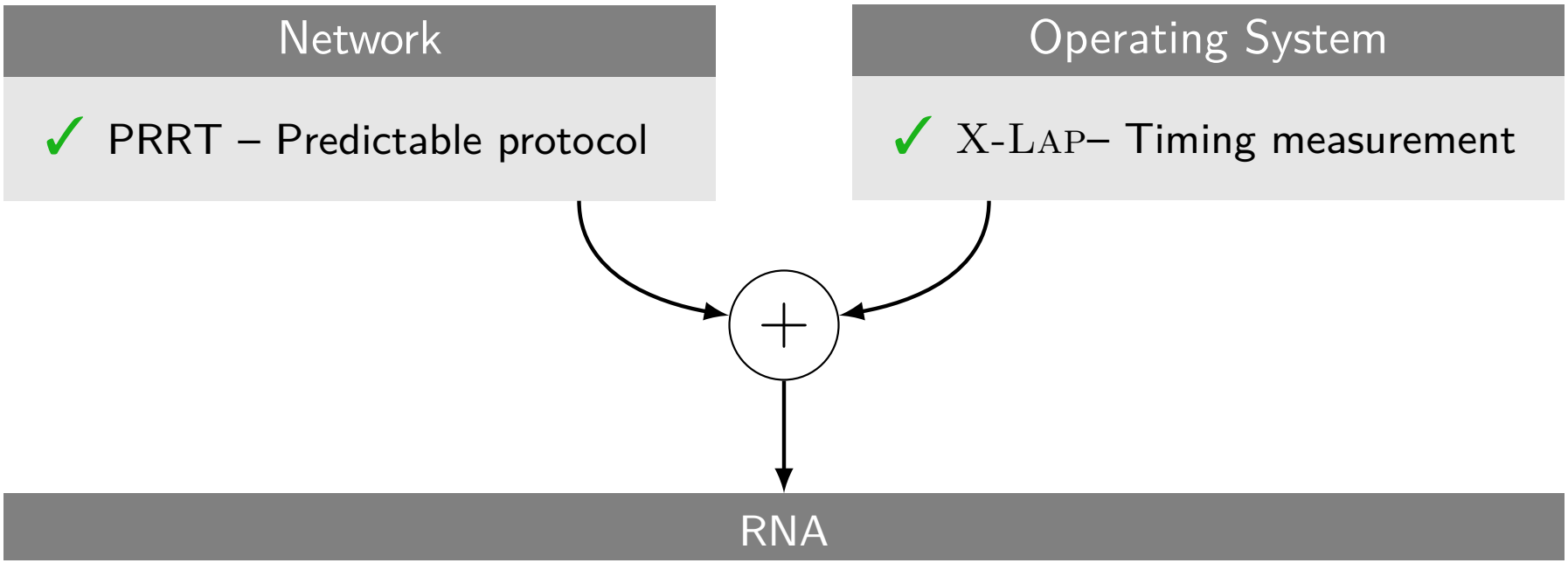
- Hardware timestamping

PRRT

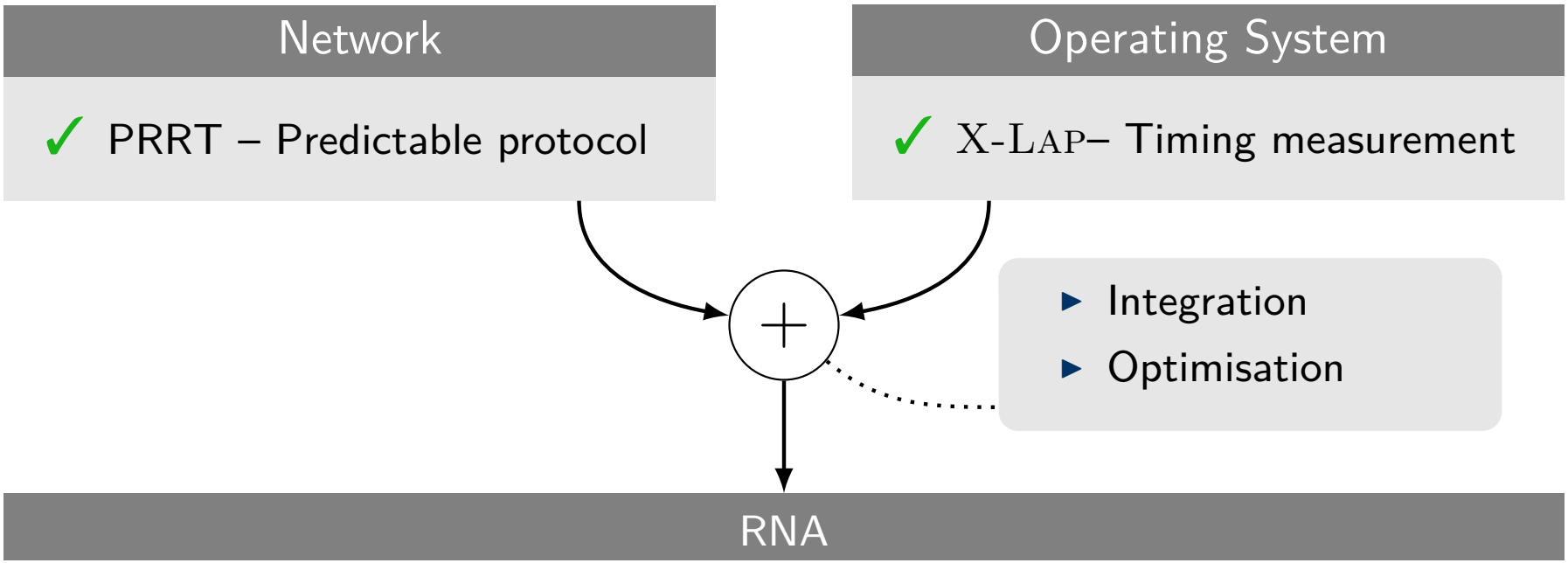
- Congestion control and delivery rate estimation
- Optimized error control for embedded platforms
- Python API: Specify more application parameters

We need **control applications** that benefit
from **predictable delay** and **predictably high reliability**!

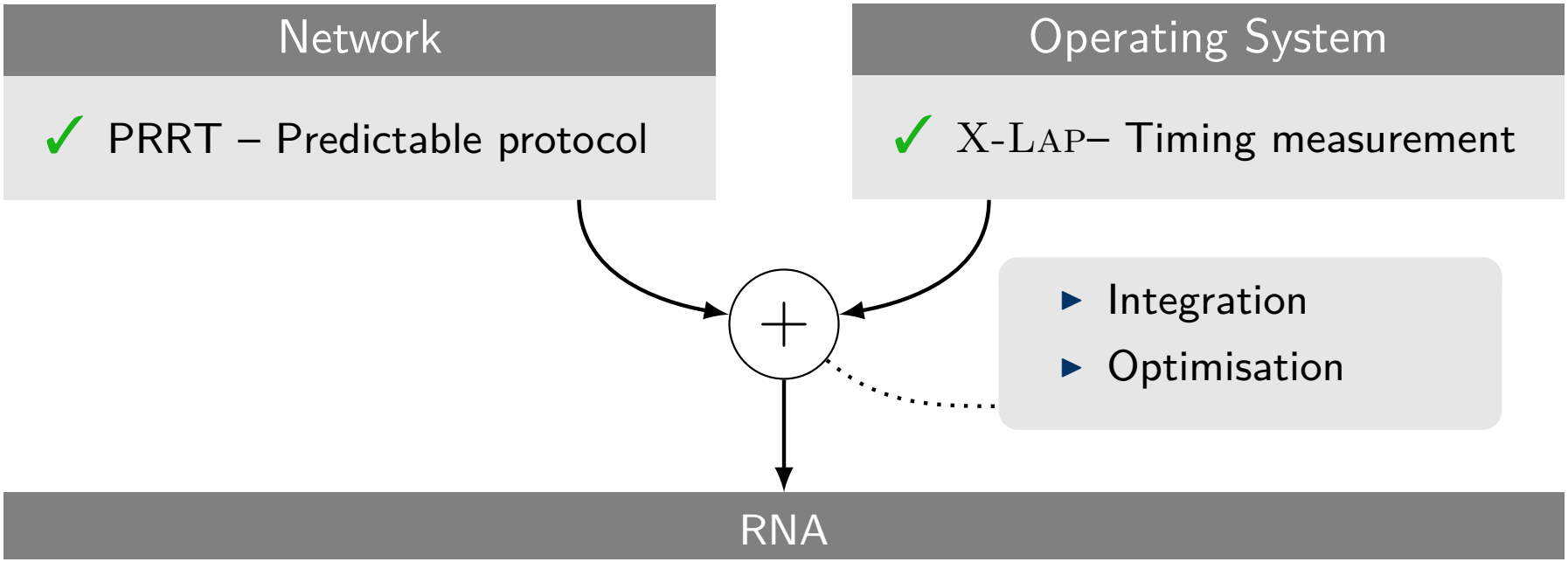
LARN


Network
✓ PRRT – Predictable protocol
Operating System
✓ X-LAP– Timing measurement
RNA

LARN


Network
✓ PRRT – Predictable protocol
Operating System
✓ X-LAP– Timing measurement
+
Integration
Optimisation
RNA

LARN



Thank you for your attention. Questions?

Backup

LARN

Server (~134EUR)

- Odroid XU4 (with fan) + Power Plug: ~72EUR
- Case (Black/Clear/Blue): ~6EUR
- eMMC Module (32GB): ~46EUR

Desktop (~238EUR)

- RNA Server (without case): ~128EUR
- VU7 Plus Display: ~77EUR
- VU7 Case: ~33EUR