- The Friendly Real-Time Operating System for the IoT

Peter Kietzmann, Thomas C. Schmidt and Matthias Wählisch
{peter.kietzmann | t.schmidt}@haw-hamburg.de, m.waehlisch@fu-berlin.de

iNET RG, Department of Computer Science
Hamburg University of Applied Sciences

November 17, 2016

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

In Zukunft werden immer mehr Maschinen, Waren und Alltagsgegenstände mit Sensoren und Funkchips ausgestattet, damit sie selbstständig miteinander kommunizieren können. Das "Internet der Dinge" kann aber nur Realität werden, wenn es einheitliche Vernetzungsstandards gibt. [1]

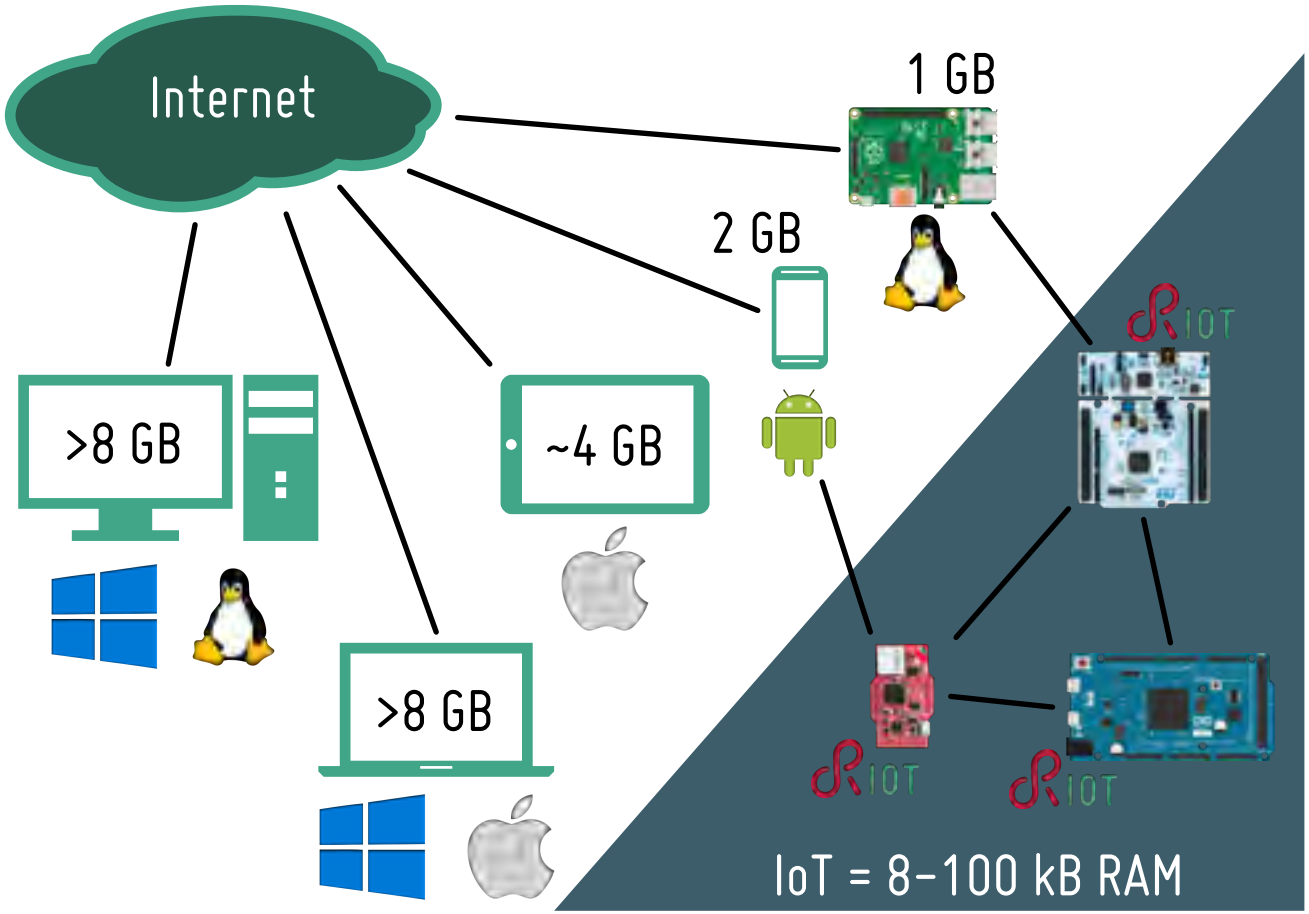
[1] `http://www.cebit.de/de/news-trends/trends/internet-der-dinge`, 2015

Agenda

Introduction

IoT Software Platform

- Which software platform to use on IoT devices?



If your device cannot run Linux, use RIOT!

Desing Goals for an IoT OS

Requirements I

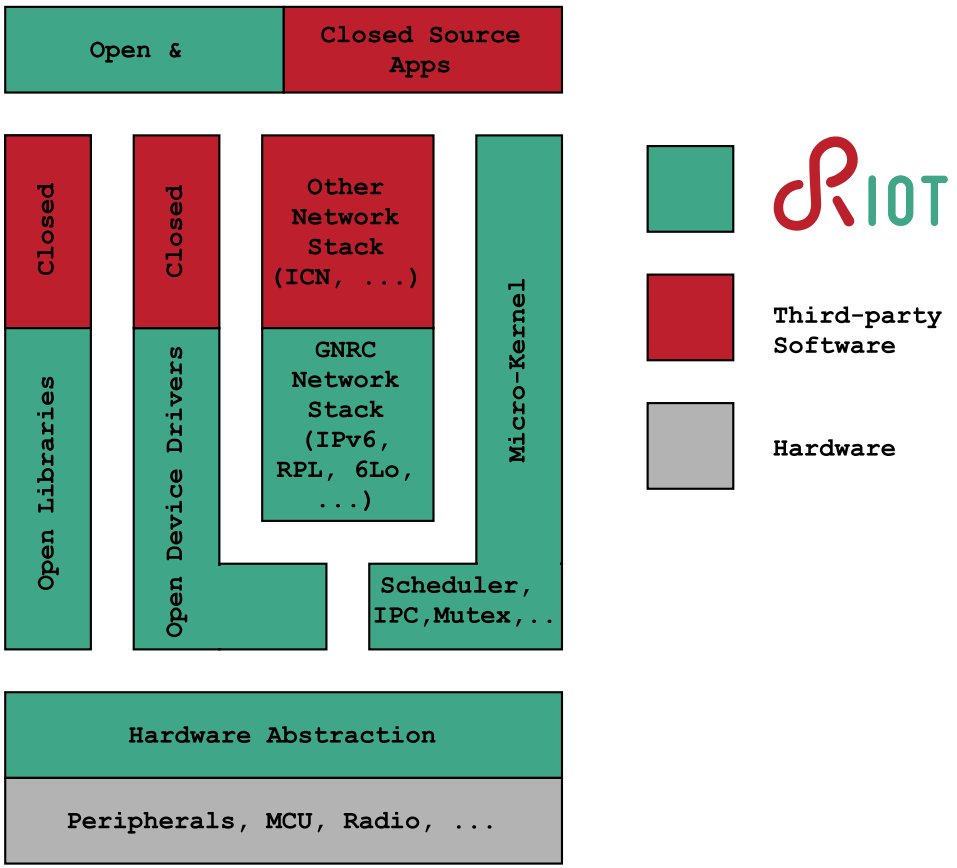
- Hardware abstraction
  - Re-usability of complex software (device drivers, network stacks, ...)
  - Heterogeneous hardware (8-bit, 16-bit, 32-bit MCUs)
- Resource efficiency
  - Memory
  - Energy
- Network support
  - Standard (Internet) protocols
  - Adaption layers needed (e.g. 6LoWPAN)
  - Different stacks (IPv6, ICN, ...)

Requirements II

- Real-time capabilities
  - Deterministic latencies
  - Dependability
- Standard programming interfaces
- Security and modifiability
  - Mechanism for software updates
  - Open source for transparency, control, trust

RIOT in One Slide

- Open Source (LGPLv2.1)
- Open-access protocol specs (IETF/IRTF)
- Community-driven development (GitHub)



RIOT Real-Time

Micro-Kernel

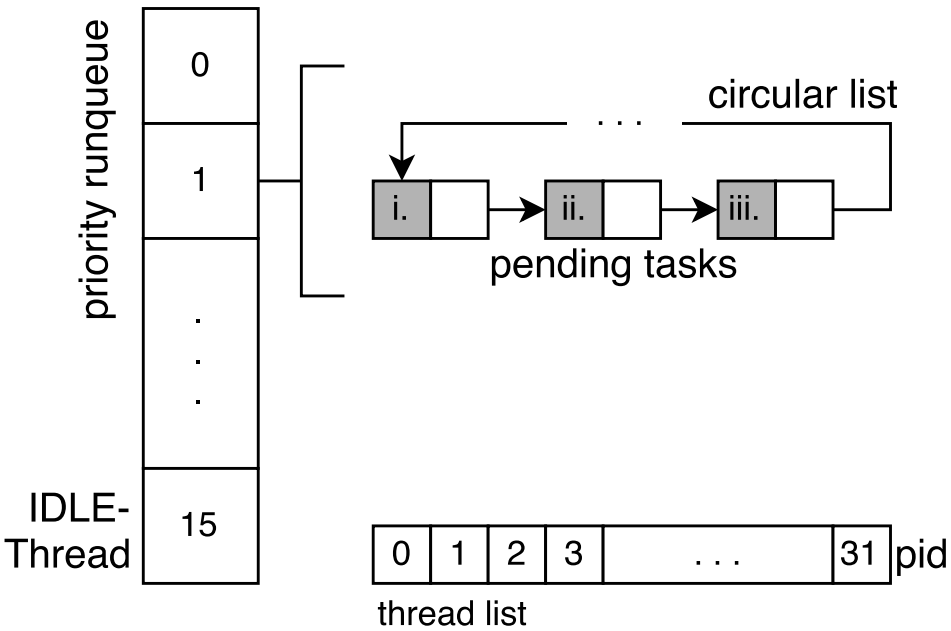
- Micro-Kernel offers basic functionalities for:
  - Multi-threading / context switching
  - Scheduling
  - Inter-process messaging (IPC)
  - Synchronization primitives
- Other features via MODULE through slim API
- Memory efficiency, configurability, stability

Scheduler Concept

- Tickless scheduling
  - No cyclic wake-up
  - Reduce system load
- Wake-up only from *event* (external interrupt, timer, ...)
- Preemptive, priority scheduling
  - Static priorities set a priori
- Highest priority active thread is always run, only interrupted by ISR

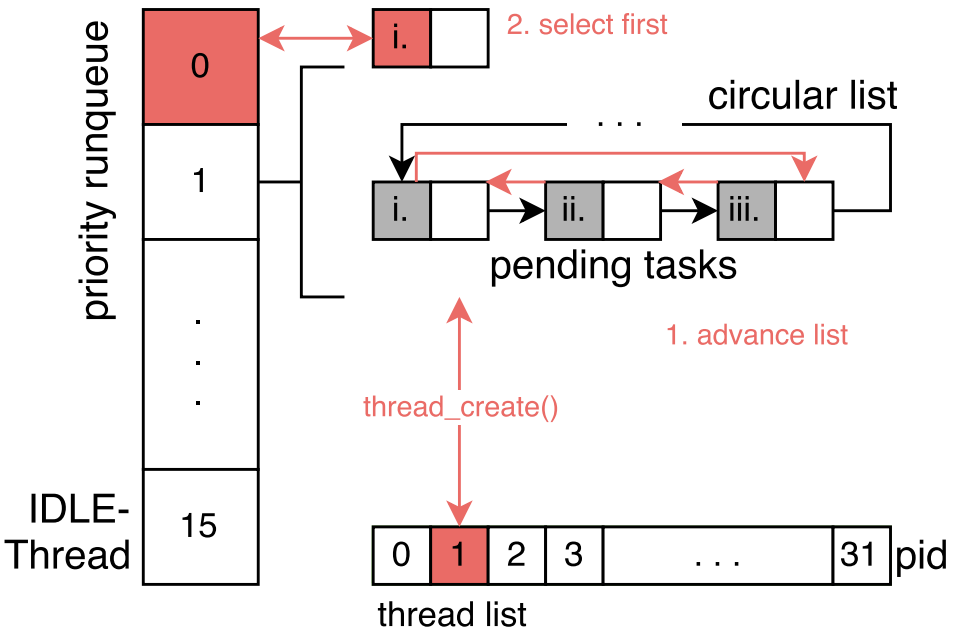
Scheduler Components

- Thread list contains all threads (active or sleep)
- Runqueue contains pending tasks
- 16 prios: each with circular list of pendig tasks



Preemptive Context Switching

- High prio thread created or mutex locked
  - From (low prio) thread
  - From ISR
- Context switch procedure
  - Advancing the circular list of the issuing task
  - Find highest priority that has pending tasks and select first

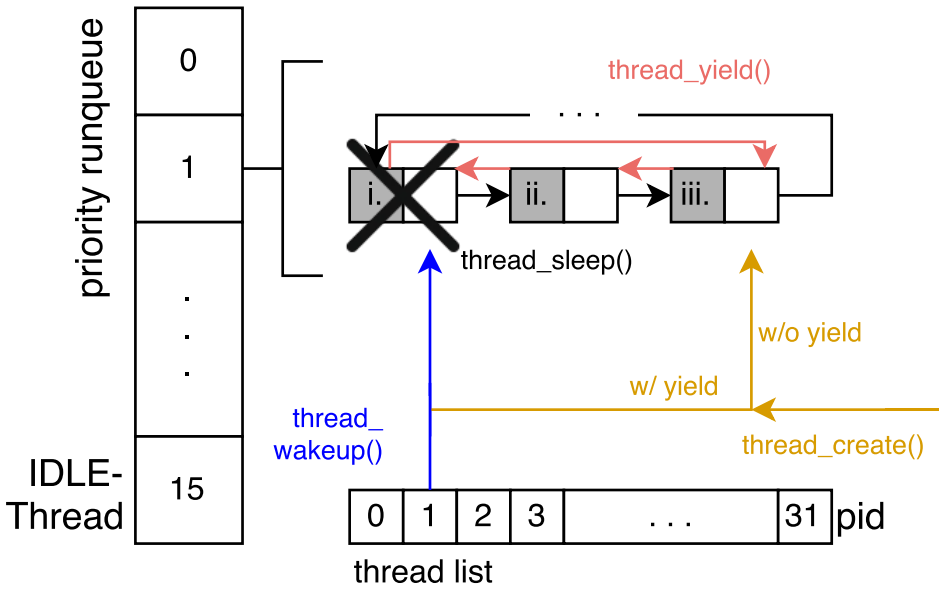


Real-Time Scheduling

- Two aspects enabling (soft) real-time
  - (1) Advancing circular list
    - Independent from number of threads in a list
    - There is just one list entry to adjust
  - (2) Finding thread with highest priority
    - Maximum through limited number of priorities
    - Always taking first entry
- Scheduling complexity of O(1)

Non-Preemptive Context Switching

- Wakeup/create thread with same prio as active thread
- Thread yields CPU access
  - Cooperative scheduling
- Thread goes sleep

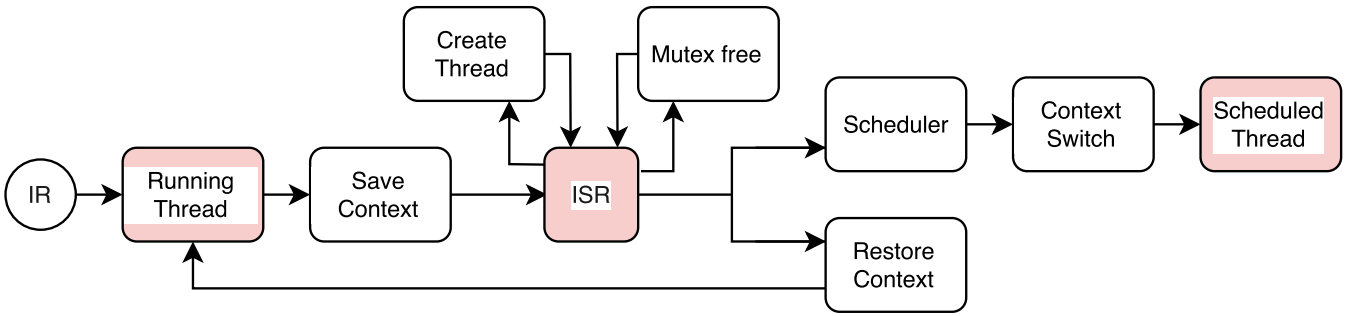


Action	Time[μs]	Cycles
`thread_yield()`	4,1	295
`thread_wakeup()`	5,5	390
`mutex_unlock()`	5,8	415
`msg_send()`	6,4	460

On ARM Cortex M3, 72 MHz

Interrupt Handling

- In a preemptive system, interrupts must go through kernel
  1. Interrupt occurs
  2. MCU jumps into kernel and saves context
  3. ISR is called in interrupt context
  4. Return to kernel
  5. Check if context switch is needed
- No need for scheduler-call if no context switch is needed



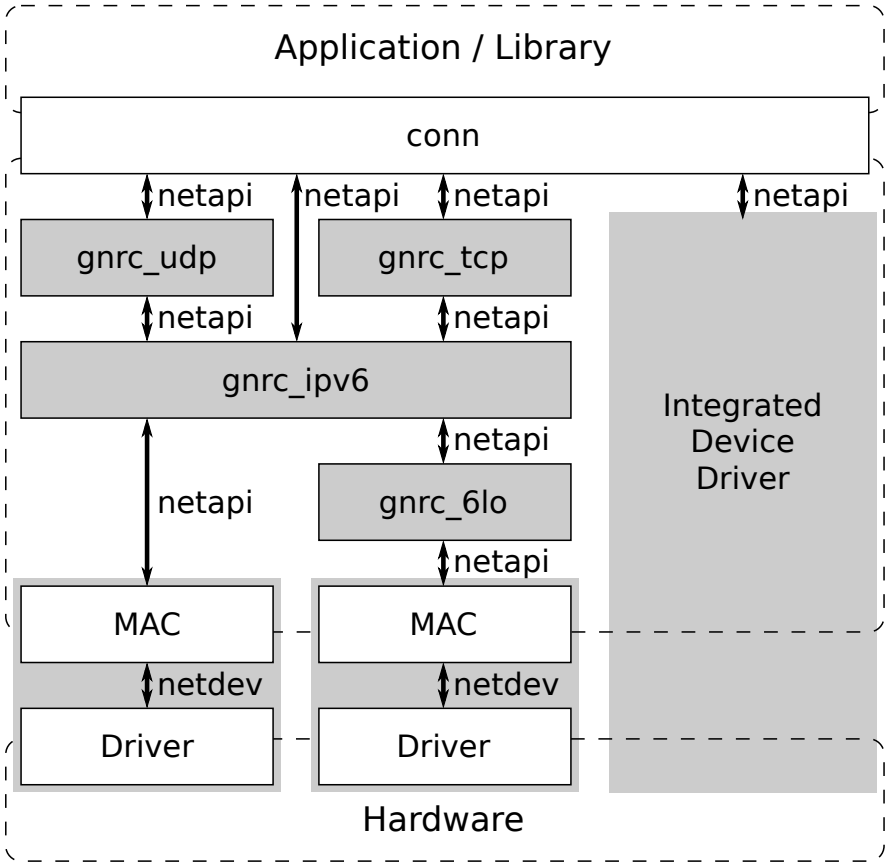
Event to ISR latency hardware[2]: 12 cycles, with RIOT: 31 cycles

[2] ARM Cortex-M3. Technical Reference Manual. Revision: r1p1

RIOT Network Stacks

GNRC Network Stack

- Generic network API *conn*
- Modules in separate threads
- Uniform interface *netapi*
- Uniform device API *netdev*
- Central coordinator *netreg*
- Central packet buffer *pktbuf*



Recursive architecture of GNRC

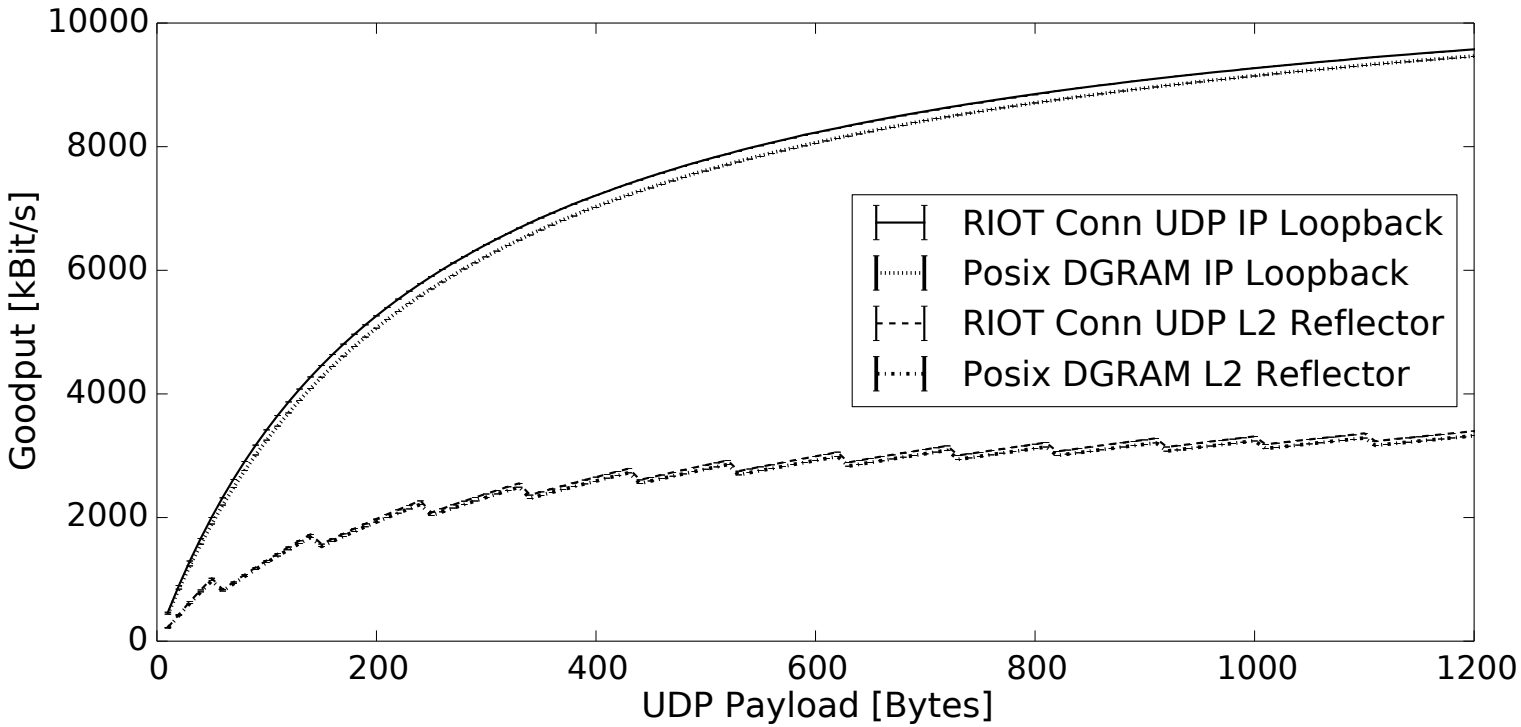
IoT-lab Testbed

- Sensor node in IoT-lab testbed `www.iot-lab.info`
- ARM Cortex-M3, 32-bit
- 72 MHz MCU
- 64 kB RAM / 512 kB ROM
- Atmel IEEE802.15.4 radio
- A couple of sensors and LEDs

Sensor node iotlab-m3

Single Layer Performance

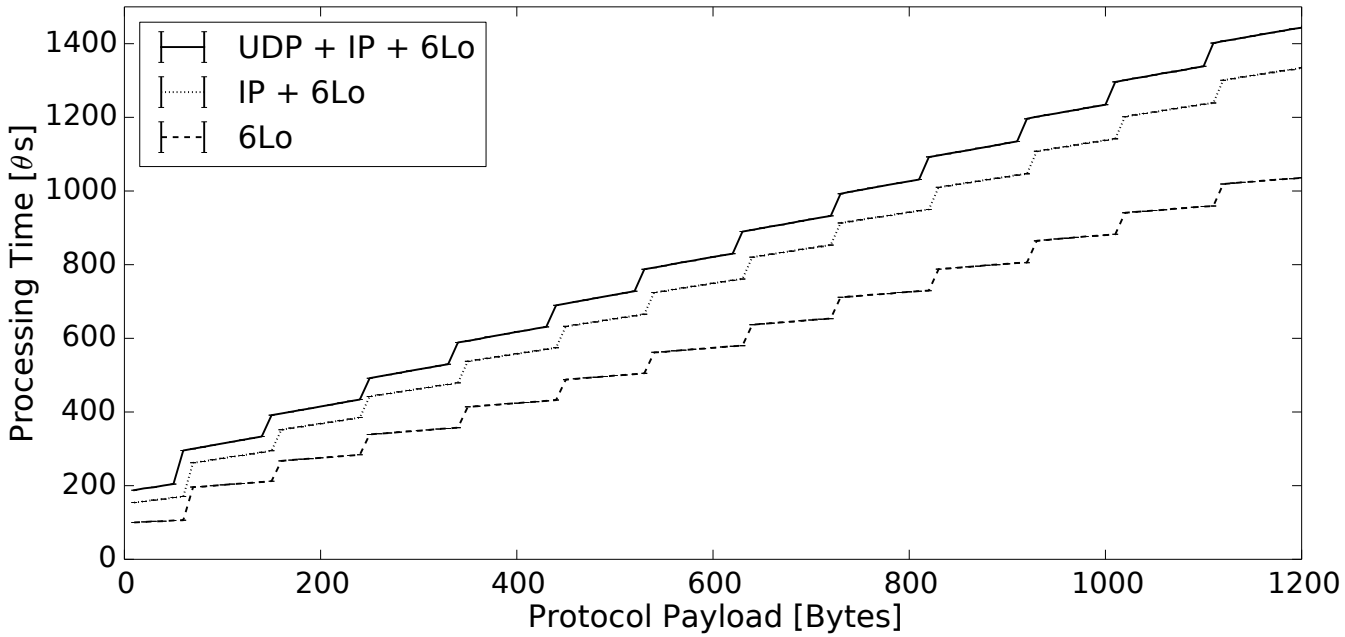
- Pure software, no drivers
- Ability to serve 802.11 a/b based network
- 6LoWPAN is costly



UDP goodput w/, w/o 6LoWPAN.

Single Layer Performance

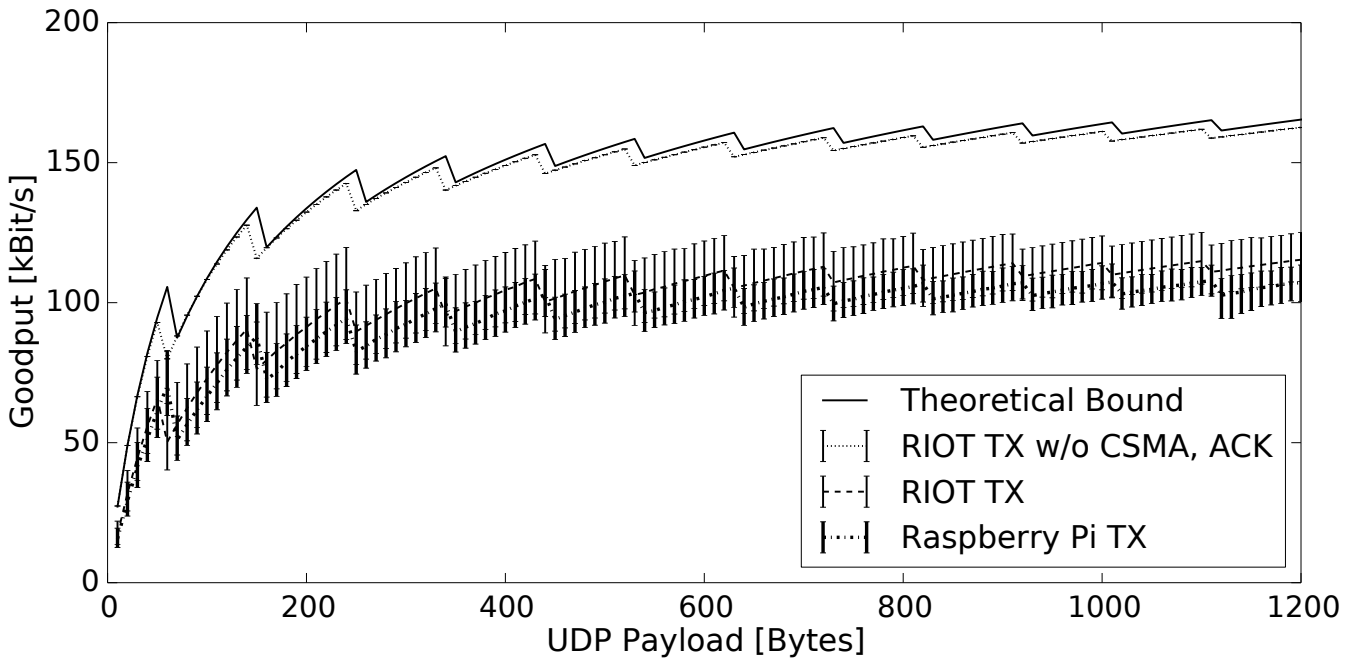
- 6LoWPAN takes costly due to fragmentation/reassembly
- IP complexity strongly depends on direction
  - Next hop determination (FIB lookup, cache management)
- UDP checksum calculation
- Largely varying rates on different layers



Average processing times per packet.

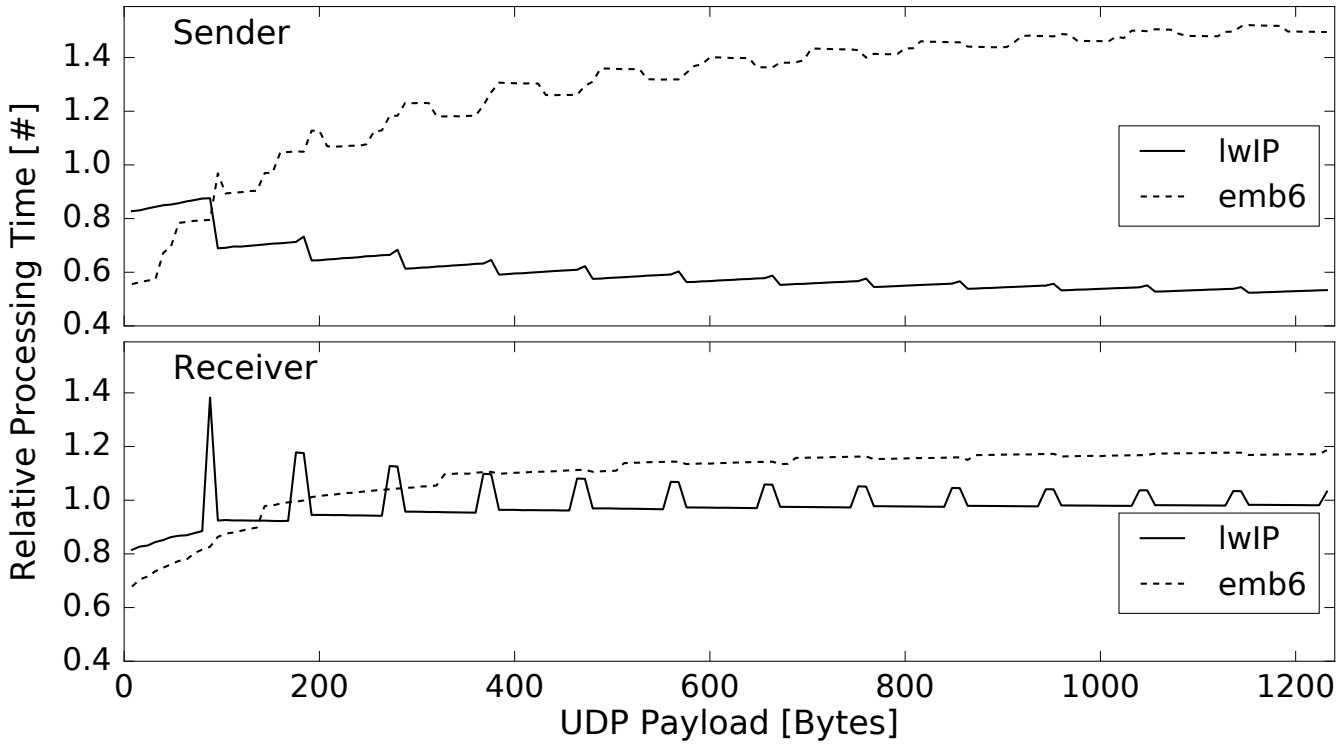
Goodput w/ Transmission

- Full stack measurement w/ driver and 802.15.4 transmission
- Results remain below theoretical curve because MAC coordination
- Plain stack serves driver at full speed
- Overall goodput limited by 802.15.4 media access



Processing Time Comparsion

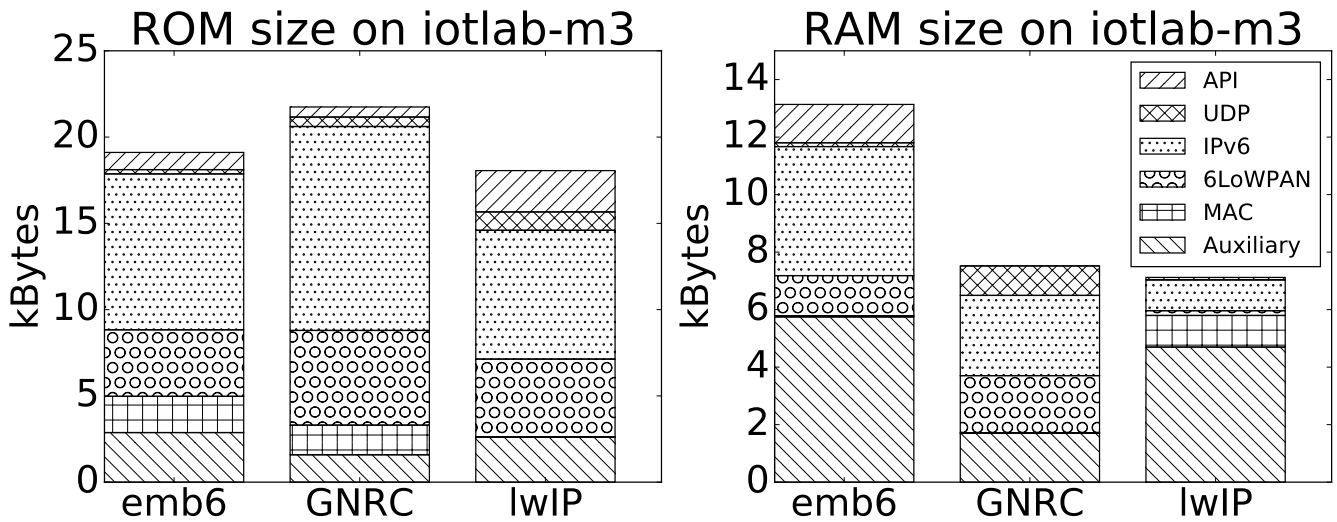
- emb6 slow packet queue, fast tinygrams
- lwIP send fragments in one go
- Peaks from sub-optimal fragmentation payloads (known issue)
- mbox-based IPC vs standard IPC



Runtime of networks stacks relative to GNRC.

Memory Consumption Comparsion

- Buffer size for IPv6 packet incl. metadata
- Maturity of code might affect ROM
- GNRC overhead by threaded model?
  - Central universal packet buffer
  - Allowing variably-sized chunks



Conclusion and Outlook

Conclusion

- Portable OS that fits constraints of IoT devices
- Functionalities of a full-fledged operating system
  - Real-time
  - Multi-threading
  - Hardware abstraction
  - Networking
- Modular and expendable stack without introducing significant performance penalties
- Overall throughput determined by 802.15.4 access

Outlook

- Kernel real-time measurements
- Finer comparison of RIOT and other OS
- Finer energy measurements
- Over the air updates
- Bluetooth Low Energy
- Low Power Wide Area Network
- Information-Centric Networking

Thank you for your attention. Questions?

iNET:	`http://inet.haw-hamburg.de`
RIOT:	`http://www.riot-os.org`
GitHub:	`http://github.com/RIOT-OS/RIOT`