

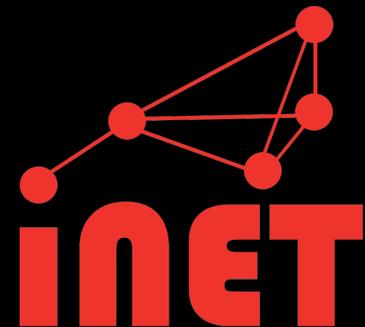
# Leistungsmessung eines modularen Netzwerk-Stacks für das IoT-Betriebssystem



Peter Kietzmann, Martin Landsmann, Thomas C. Schmidt,  
Hauke Petersen, Martine Lenders, Matthias Wählich  
iNET AG, Department Informatik  
Hochschule für Angewandte Wissenschaften Hamburg  
[peter.kietzmann@haw-hamburg.de](mailto:peter.kietzmann@haw-hamburg.de)



Hochschule für Angewandte  
Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*



# AGENDA

- Motivation IoT
- RIOT Überblick
- RIOT Netzwerk-Stack
- Leistungsvermessung
  - Durchsatz
  - Energie
  - Speicher
- Zusammenfassung

# Motivation IoT

„Der Name ist wörtlich zu nehmen: Jedes Ding soll einen Netzanschluss und eine IP-Adresse bekommen. Ob Auto, Kaffeemaschine, Zahnbürste oder Telefon, alle sollen miteinander vernetzt und übers Internet erreichbar sein.“

*<http://www.heise.de/thema/Internet-der-Dinge>, Sept. 2015*

„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.“

*<http://www.cebitt.de/de/news-trends/trends/internet-der-dinge/>, Sept. 2015*

Wir wollen das IoT mit RIOT ermöglichen

# IoT-Hardware

- Heterogene Hardware-Landschaft mit begrenzten Ressourcen

Name	RAM	Flash
Klasse 0	< 10 kB	< 100 kB
Klasse 1	~ 10 kB	~ 100 kB
Klasse 2	~ 50 kB	~ 250 kB



Arduino Uno board  
8 Bit Atmel AVR

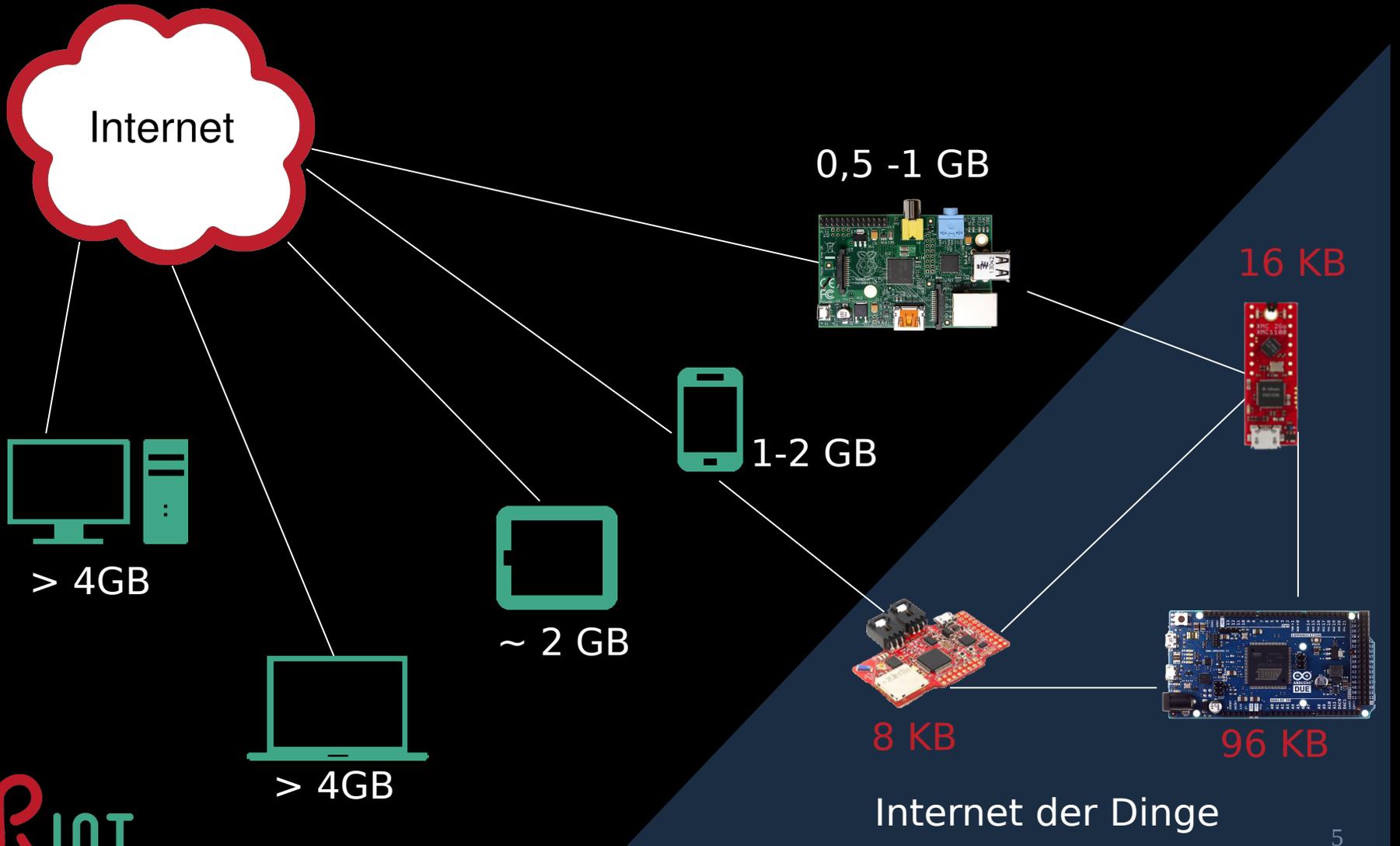


TI eZ430 Chronos watch 16 Bit  
MSP 430 sub-GHz Transceiver

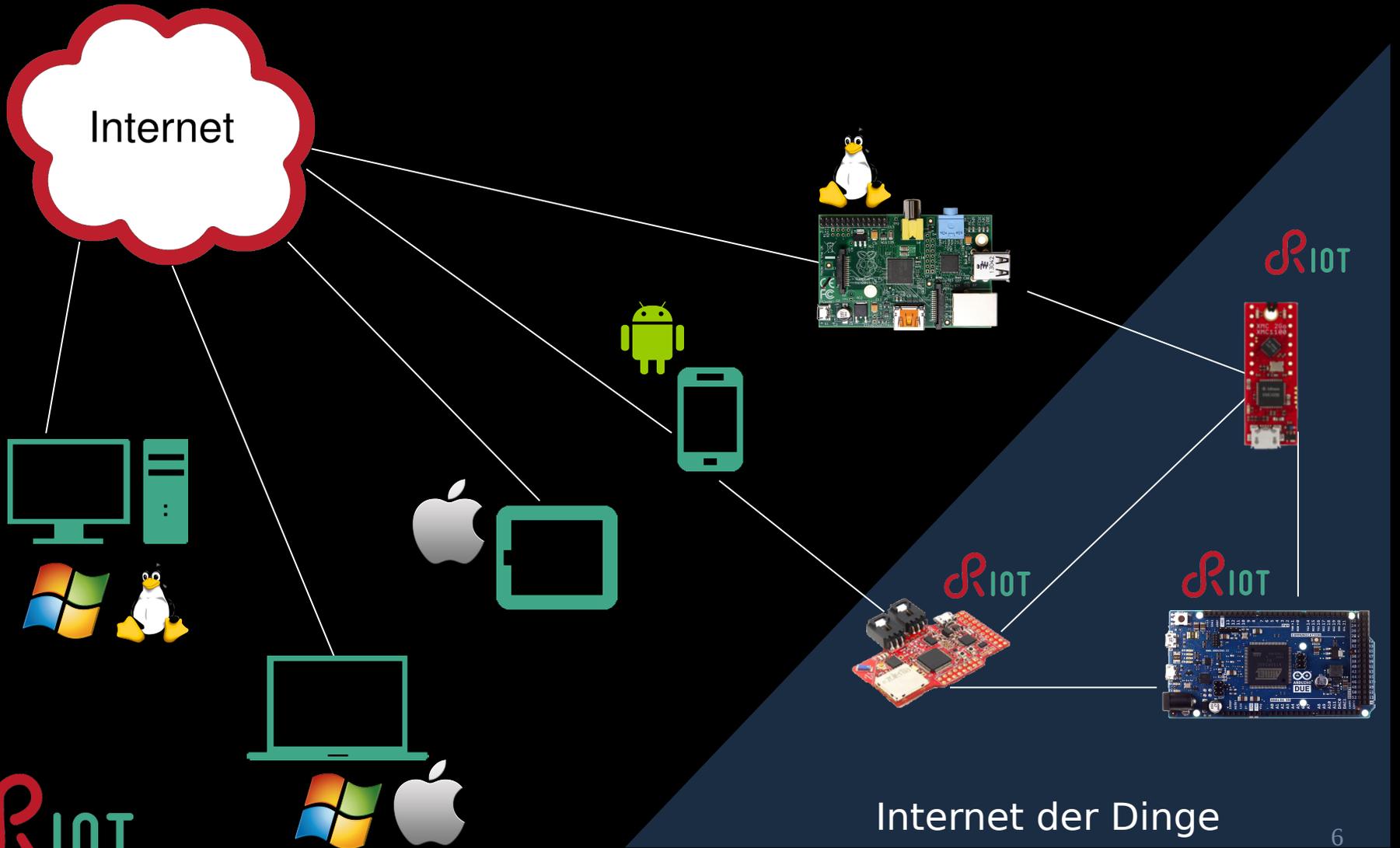


SAM R21 Xplained Pro  
32 Bit ARM Cortex-M0+  
2.4 GHz Transceiver

# Internet-Hardware



# „If your IoT device cannot run Linux, then run RIOT“



# AGENDA

- Motivation IoT
- RIOT Überblick
- RIOT Netzwerk-Stack
- Leistungsvermessung
  - Durchsatz
  - Energie
  - Speicher
- Zusammenfassung

# RIOT: „Das freundliche Betriebssystem für das Internet der Dinge“

- 2008: Wurzeln in *FeuerWare*, OS für WSN
- 2010: *μkleos*, erste Adaption von Internetprotokollen
- 2013: **RIOT** „re-branding“ und Veröffentlichung auf INFOCOM

Gründunsinstitutionen:



# RIOT: Überblick

- RIOT erfordert nur wenige kB RAM/ROM
- Programmierung in **ANSI-C oder C++**
- Gleicher Code auf heterogener Hardware
  - 8Bit Hardware (z.B. Arduino)
  - 16Bit Hardware (z.B. MSP430)
  - 32Bit Hardware (z.B. ARM Cortex-M)
- Unterstützung von Standard-Werkzeugen

Valgrind



**GDB**  
The GNU Project  
Debugger

# RIOT Fakten

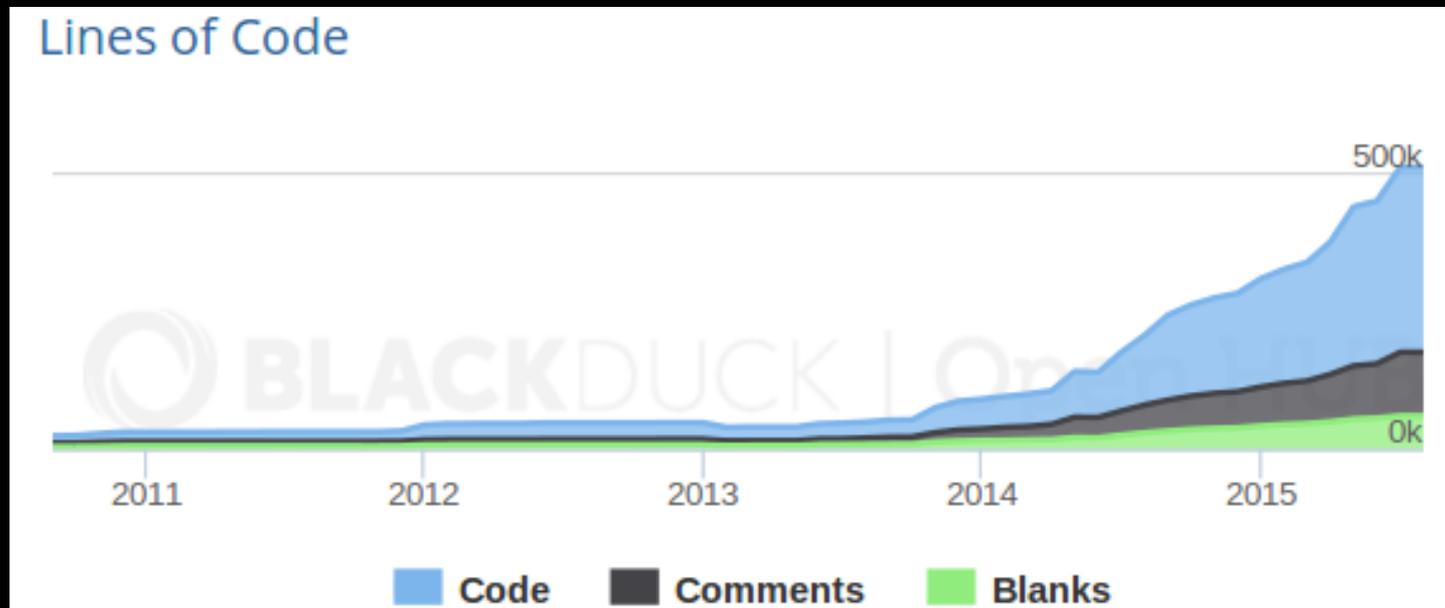
- Mikrokern Architektur (**robust**)
  - Nur der Kernel ~2.5kB RAM/10kB ROM @ ARM Cortex-M 32-bit
- Modular (**erweiterbar**)
- Kernel ohne zyklische Ticks (**energieeffizient**)
- Deterministisches Scheduling mit konstanter Zeit für Prozesswechsel (**echtzeitfähig**)
- Präemptives Multi-Threading & leichtgewichtige IPC

E. Baccelli, O. Hahm, M. Günes, M. Wählisch, T. Schmidt. RIOT OS: Towards an OS for the Internet of Things. In *The 32nd IEEE International Conference on Computer Communications (INFOCOM 2013)*.

H. Will, K. Schleiser, J. Schiller. A Real-Time Kernel for Wireless Sensor Networks Employed in Rescue Scenarios. In *The 34th IEEE Conference on Local Computer Networks (LCN 2009)*.

# RIOT Community

- Open Source (LGPLv2.1)
- 77 Entwickler, 382 GitHub Forks, 950 Twitter followers



„RIOT is one of the largest open-source teams in the world“

[www.openhub.net/p/RIOT-OS](http://www.openhub.net/p/RIOT-OS), Sept. 2015

# AGENDA

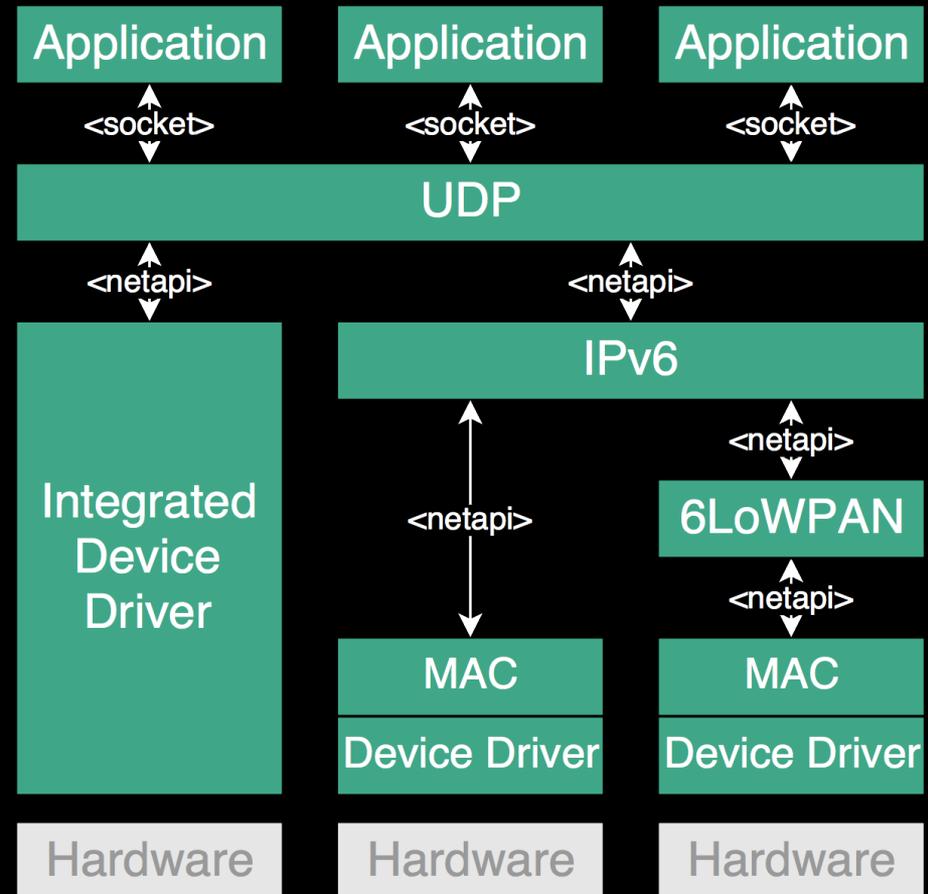
- Motivation IoT
- RIOT Überblick
- RIOT Netzwerk-Stack
- Leistungsvermessung
  - Durchsatz
  - Energie
  - Speicher
- Zusammenfassung

# Netzwerk-Stack Ziele

- Bereitstellen von standardisierten Internetprotokollen
- Ein Netzwerk-Stack für heterogene Hardware
- Modularität für verschiedene IoT-Szenarien
- Nicht die effizienteste, aber eine klar gegliederte und universelle Implementierung
- Anforderungen von Klasse 1 Geräten und größer

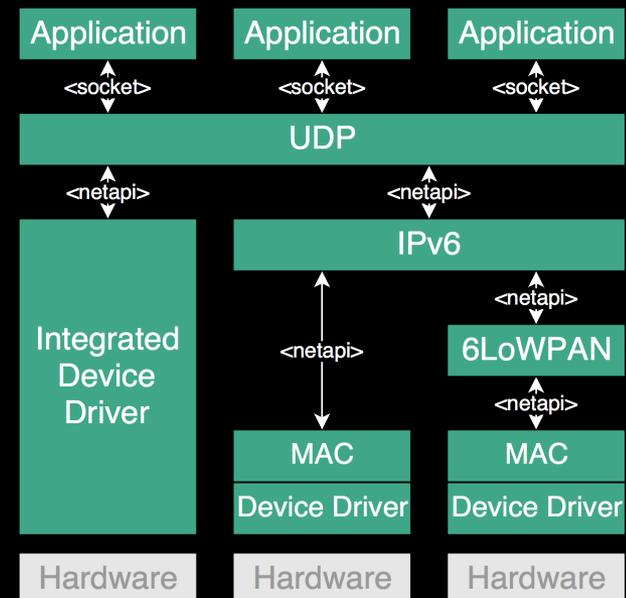
# Netzwerk-Stack Konzepte

- Drei Wesentliche Konzepte :
  - Multithreading Architektur
  - Vereinheitlichte Schnittstellen
  - Deduplizierende Datenhaltung



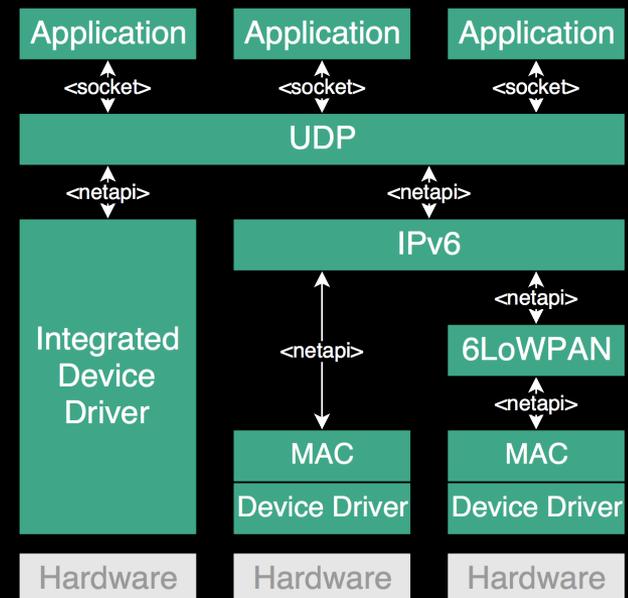
# Netzwerk-Stack Multi-Threading

- Modularität durch Aufteilung in Threads
- Typ. ein Protokoll (z.B. IPv6, UDP, ..) in einem Thread
- Unabhängig voneinander entwickelbar
- Konfigurierbar und universell
- Potenziell erhöhter RAM



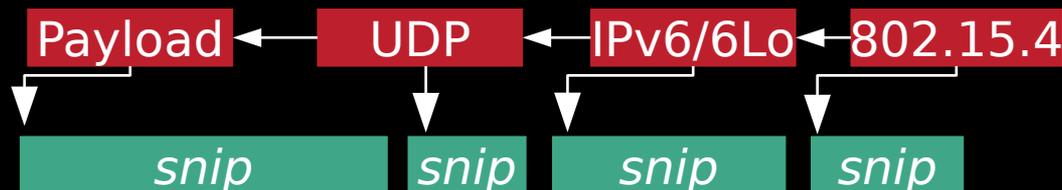
# Netzwerk-Stack Inter-Modul Schnittstelle

- Vereinheitlichte Kommunikationsschnittstelle
- *netapi* – Schnittstelle basiert auf IPC
- Kommunikation zwischen Modulen/Threads
- *send/receive* asynchron, *get/set* synchron
- Gleiche API für jede Implementierung



# Netzwerk-Stack Paketspeicher

- *pktbuf* – zentraler Paketspeicher
- Deduplizierende Datenhaltung
- Minimierung der Kopiervorgänge
- Statisch allozierter Buffer von 4kB  
(drei IPv6-Pakete + x)
- Verlinkte Listen von Fragmenten – *snips*  
→ einzelne Header bearbeiten und Größe ändern



# AGENDA

- Motivation IoT
- RIOT Überblick
- RIOT Netzwerk-Stack
- **Leistungsvermessung**
  - Durchsatz
  - Energie
  - Speicher
- Zusammenfassung

# Leistungsvermessung Hardware

- Sensorknoten des IoT-lab ([www.iot-lab.info](http://www.iot-lab.info))
- ARM Cortex-M3, 32Bit
- 72 MHz CPU
- 64 kB RAM, 512 kB ROM
- IEEE802.15.4 Radio
- Vier diskrete Sensoren, 3 LEDs  
(Klasse 2 Gerät)

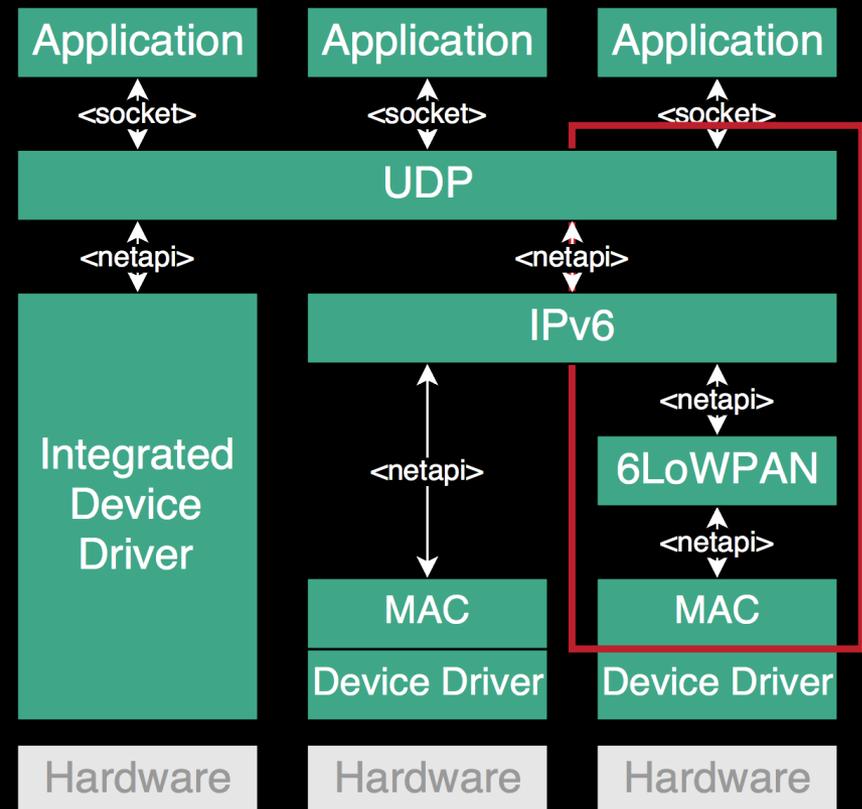


*Inria*



# Leistungsvermessung Vorgehen

- Ermittlung der Performance des Netzwerk-Stacks unter Vollast, für die Paketprozessierung
- Parameter: Durchsatz, Energieverbrauch und Speicherbedarf
- Prozessieren von UDP-Paketen verschiedener Payloads
- UDP, IPv6, 6LoWPAN (mit IPHC)
  - Link-lokale Unicast-IPv6-Adressen
- Reine Software-Performance → Ausgrenzung des Physical Layers

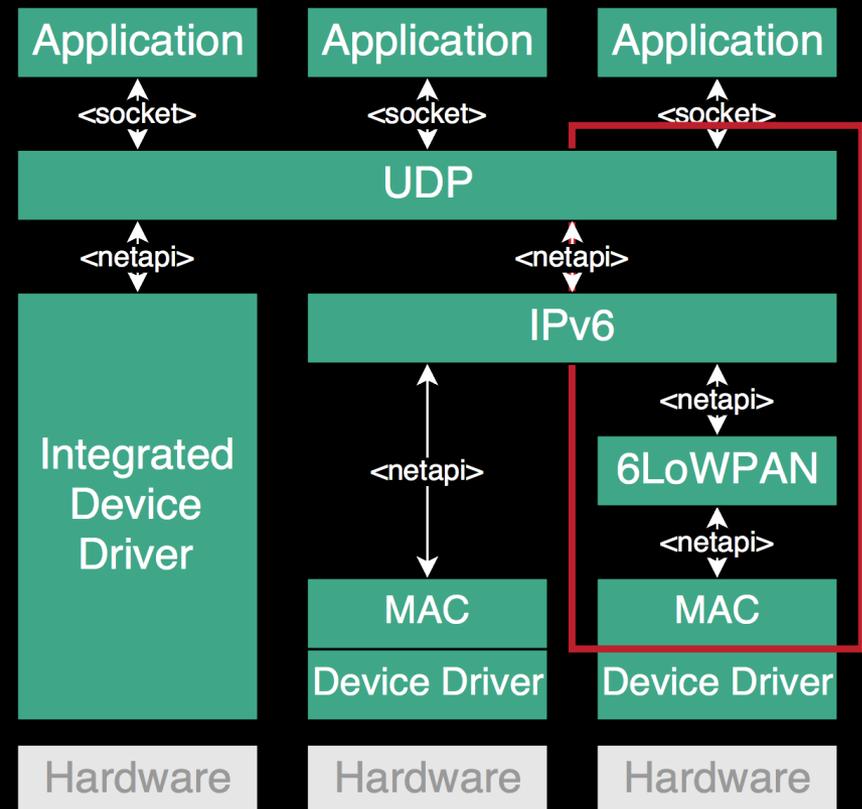


# AGENDA

- Motivation IoT
- RIOT Überblick
- RIOT Netzwerk-Stack
- Leistungsvermessung
  - Durchsatz
  - Energie
  - Speicher
- Zusammenfassung

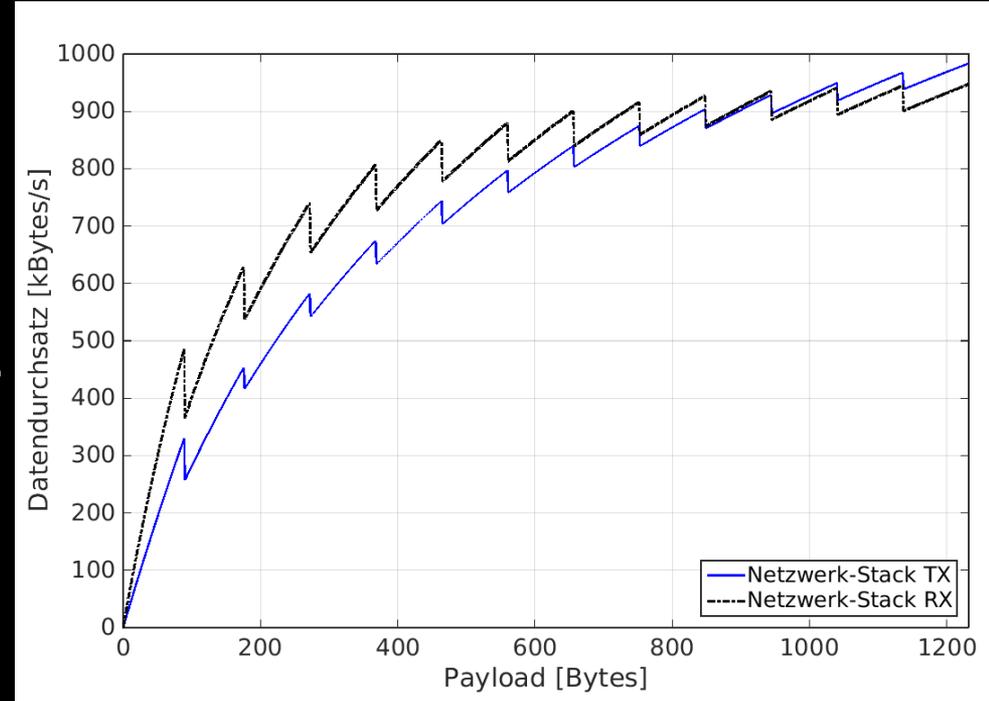
# Leistungsvermessung Datendurchsatz

- Payloads bis 0...1232B
- 1000 Pakete pro Messpunkt
- Zeitmessung für die Prozessierung im Sender und Empfänger

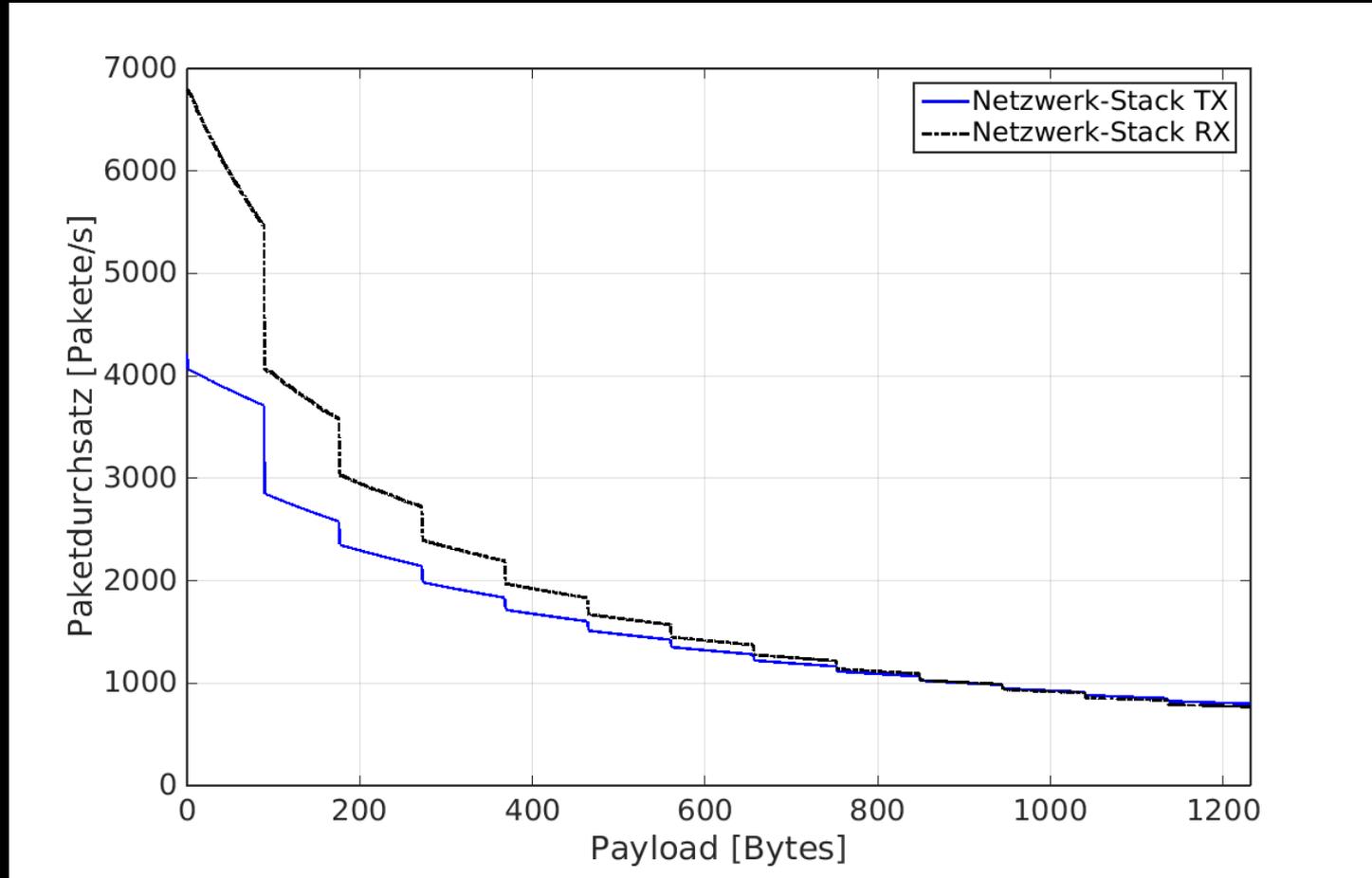


# Ergebnisse Datendurchsatz

- 6LoWPAN  
Paketfragmentierung
- Empfänger effizienter als Sender bei  $< 900\text{B}$
- Sender schreibt mehrfach auf Speicher im *pktbuf*
- Empfänger schreibt nur 1x auf Speicher im *pktbuf*  
→ Optimierungsbedarf



# Netzwerk-Stack Paketdurchsatz

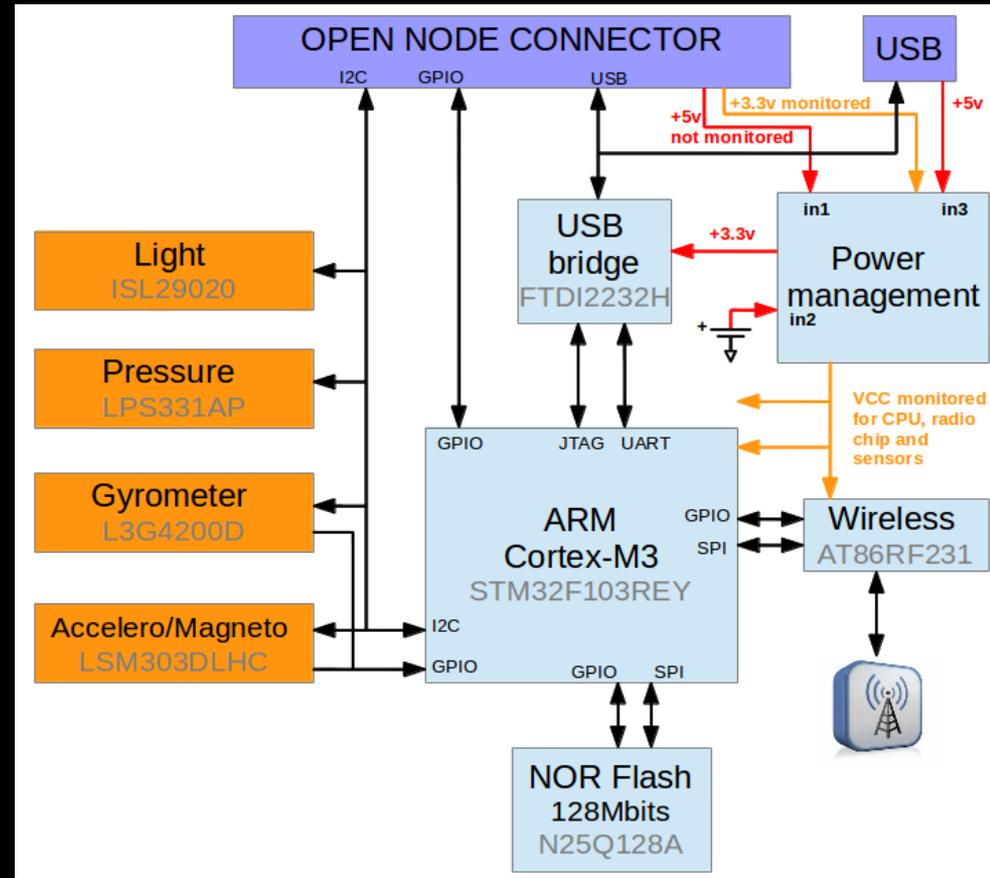


# AGENDA

- Das Internet der Dinge
- RIOT Überblick
- RIOT Netzwerk-Stack
- Netzwerk-Stack Leistungsmessung
  - Durchsatz
  - Energie
  - Speicher
- Zusammenfassung

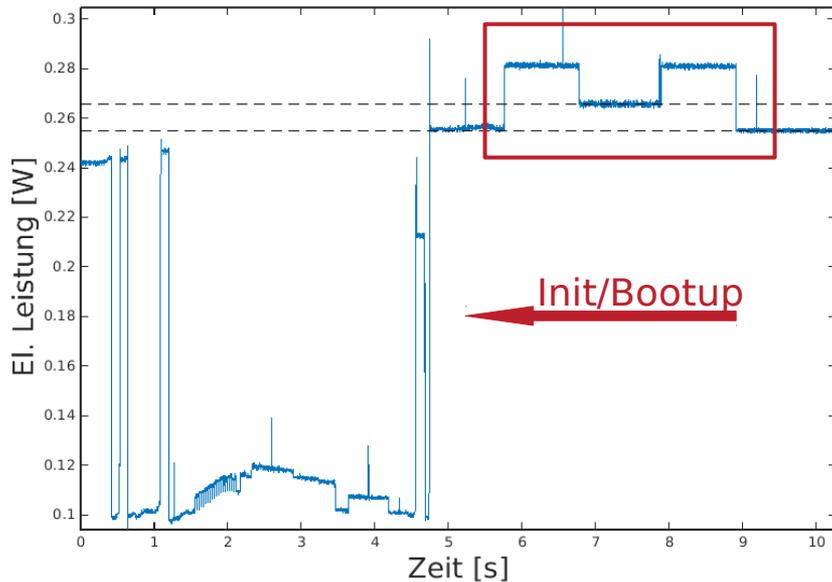
# Leistungsvermessung Energiebedarf

- 1000 Pakete
  - à 20 Bytes
  - à 1000 Bytes
- Transceiver-Verbrauch
- UART- Verbrauch
- IoT-lab ist fehlerbehaftet



[https://github.com/iot-lab/iot-lab/wiki/Hardware\\_M3-node](https://github.com/iot-lab/iot-lab/wiki/Hardware_M3-node)

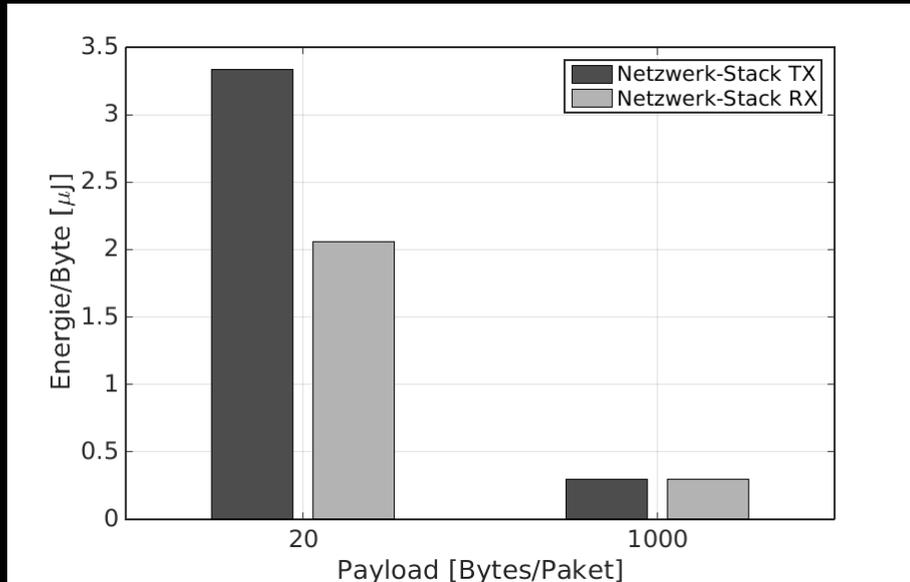
# Ergebnisse Energiebedarf



*Period: 140us, Average: 4 samples,  
Payload: 1000B, Packets: 1000*

- Annahme: Systemauslastung 100% bis Programmende
- Mittlere Leistung
  - Betrieb:  $0,266 \text{ W} \pm 0,31\%$
  - Idle :  $0,255 \text{ W} \pm 0,19\%$

# Ergebnisse Energiebedarf



*(Funk- und UART Modul eingeschaltet im Wartezustand)*

- Annahme: Systemauslastung 100% bis Programmende
- Mittlere Leistung
  - Betrieb:  $0,266 \text{ W} \pm 0,31\%$
  - Idle :  $0,255 \text{ W} \pm 0,19\%$
- Integrieren über Rechenzeit ergibt Energiebedarf
- Hier normiert auf ein Payload Byte

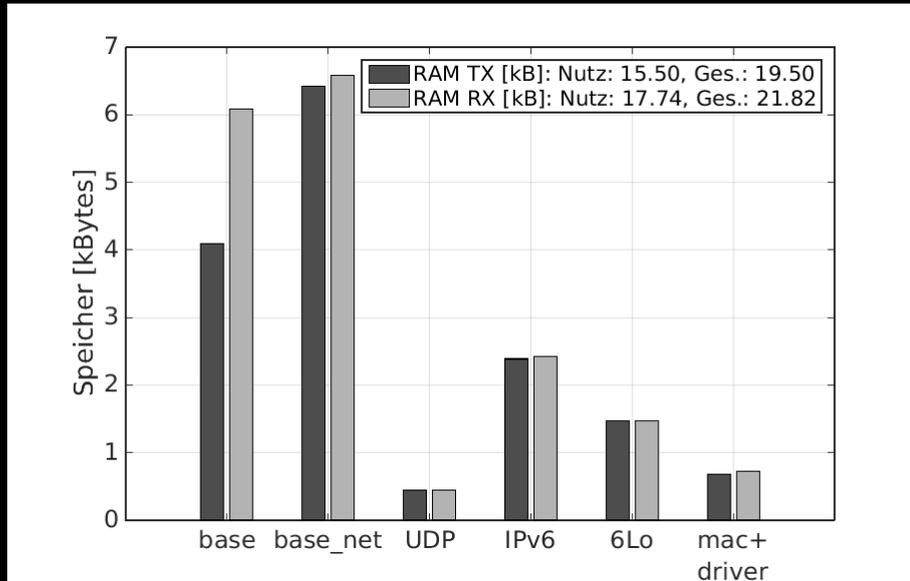
# AGENDA

- Motivation IoT
- RIOT Überblick
- RIOT Netzwerk-Stack
- Leistungsvermessung
  - Durchsatz
  - Energie
  - Speicher
- Zusammenfassung

# Leistungsvermessung Speicherbedarf

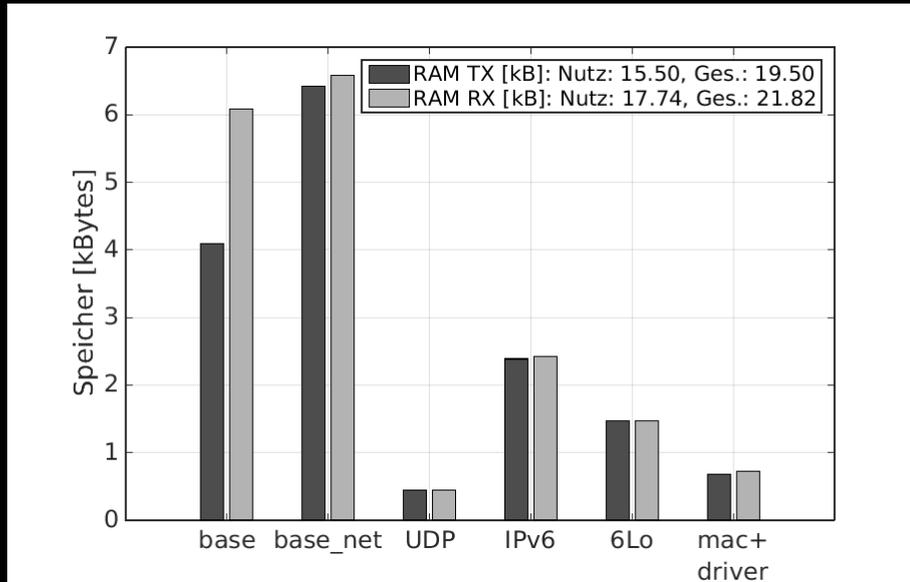
- Paketgröße so, dass 6LoWPAN **Fragmentierung** verwendet wird
- ROM (Programmspeicher):
  - Nur zur Compile-Zeit
  - Sender ~53KB ROM
  - Empfänger ~55KB ROM
- RAM (Arbeitsspeicher):
  - Compile-Zeit & Betrieb
  - Sortiert nach Modulen  
(Sortierung gelinkter Object-files)
  - Der nicht genutzte Stack-Speicher zur Laufzeit wurde zur Darstellung der Performance abgezogen

# Ergebnisse Arbeitsspeicher



- *base*:  
Basisfunktionen,  
Anwendungsprogramm  
(für Empfänger ein  
zusätzlicher Thread)
- *base\_net*:  
Protokollunabhängige  
Module des Netzwerk-  
Stacks (z.B. *pktbuf*,  
*netapi*)

# Ergebnisse Arbeitsspeicher



- Ähnlicher Speicherbedarf in Sender und Empfänger
- Problemlos für Hardware wie z.B. IoT-lab
- Erfüllt problemlos Klasse 2 Geräte-Anforderung
- Optimierungspotential

# AGENDA

- Motivation IoT
- RIOT Überblick
- RIOT Netzwerk-Stack
- Leistungsvermessung
  - Durchsatz
  - Energie
  - Speicher
- Zusammenfassung

# Zusammenfassung

- RIOT als leichtgewichtiges IoT-Betriebssystem
- Netzwerk-Stack
  - Modulare, thread-basierte Architektur
  - Deduplizierender, statischer Paketspeicher
- Datendurchsatz der **Software**: ~900 kB/s @1000B Payload
- Energieverbrauch **ohne** PHY: 0,3µJ pro Payload Byte @1000B Payload
- Speicher RAM und ROM adressiert derzeit Klasse 2 Geräte, liegt noch über Klasse 1-Anforderung
- Arbeiten sind WIP, Optimierungspotential besteht

# RIOT Community

- Web:  
<http://www.riot-os.org/>
- GitHub:  
<https://github.com/RIOT-OS/RIOT>
- Developer mailing list:  
[devel@riot-os.org](mailto:devel@riot-os.org)
- Internet Relay Chat:  
[#riot-os](irc.freenode.org)
- Twitter:  
[@RIOT\\_OS](https://twitter.com/RIOT_OS)



## Gründunsinstitutionen:





[www.riot-os.org](http://www.riot-os.org)