MASTER: Long-Term Stable Routing and Scheduling in Low-Power Wireless Networks

Oliver Harms\textsuperscript{1,2}, Olaf Landsiedel\textsuperscript{1,2}

\textsuperscript{1}Kiel University, Germany
\textsuperscript{2}Chalmers University of Technology, Sweden
Industrial Internet of Things (IIoT)

• Requirements:
  • High reliability
  • Low latency
  • Guaranteed performance
• Central Scheduling
Contributions

• MASTER: centralized router and scheduler
  • Targets IEEE 802.15.4 and TSCH MAC
  • Provides easy extendibility

• Sliding Windows: transmission strategy
  • Enables flexible, stable, and reliable communication

• Extensive testbed evaluation
  • Long-term stable schedule (24 hours)
Outline

1. Motivation

2. Background
   a. Time-Slotted Channel Hopping (TSCH)
   b. Central Scheduling

3. MASTER
   a. Design
   b. Transmission strategies: Sliding Windows

4. Evaluation

5. Conclusion
Time-Slotted Channel Hopping (TSCH)

- TDMA
- FDMA (up to 16 channels)
- Control information
- Recurring Slotframes
Central Scheduling
Outline

1. Motivation
2. Background
   a. Time-Slotted Channel Hopping (TSCH)
   b. Central Scheduling
3. MASTER
   a. Design
   b. Transmission strategies: Sliding Windows
4. Evaluation
5. Conclusion
MASTER: Long-Term Stable Routing and Scheduling in Low-Power Wireless Networks – Oliver Harms
MASTER Overview

- Central Scheduler
- Routing Layer
MASTER building blocks
Routing

• Dijkstra’s shortest path algorithm
• Metric: ETX, \(\text{ETX}^2\), or \(\text{ETX}^3\)
Transmission Strategies

• No retransmissions
• Slot-based retransmissions
• Flow-based retransmissions
  • Sliding Windows
Traditional approaches

• No retransmissions
• Slot-based retransmissions
Sliding Windows

- Flow-based retransmissions
- Slot role: RX, TX, shared (RXTX)
- Number transmissions
  - Fixed
  - $ETX$-based
    - $n \times \sum ETX_{link}$
    - $n \times \sum [ETX_{link}]$
Sliding Windows in action

1 2 3 4 5 6

A  TX  RX  TX  RX  TX  RX
B  RX  TX  RX  TX  RX  RX
C  RX  TX  RX  TX  RX  RX
D  RX  RX  RX  RX  RX  RX
Sliding Windows in action

A  TX  TX  TX  
B  RX  RX  RX  TX  
C  RX  RX  RX  TX  TX  
D  RX  RX  RX  RX  

1  2  3  4  5  6

A → B  C → D
Scheduling

- R-LPF (Reverse Longest Path First)
- Best-effort
- Non deadline-based
Outline

1. Motivation
2. Background
   a. Time-Slotted Channel Hopping (TSCH)
   b. Central Scheduling
3. Master
   a. Design
   b. Transmission strategies: Sliding Windows
4. Evaluation
5. Conclusion
Evaluation Setup

• Testbed at Kiel University
  • 500 m²
  • 20 nodes
  • Platform: Zolertia Firefly

• Evaluation configuration
  • 6 flows, 2-4 hops
  • 2 hours
  • 1 packet per second
  • 64 bytes payload + headers
MASTER’s Transmission Strategies

- Baseline
- Slot-based
- $1 \times \sum ETX_{\text{link}}$
- $2 \times \sum ETX_{\text{link}}$
- $1 \times \sum ETX_{\text{link}}$
- $2 \times \sum ETX_{\text{link}}$
- $3 \times \sum ETX_{\text{link}}$

PDR [%] vs Latency [slots]
MASTER's Transmission Strategies

- Baseline
- Slot-based
- \(1 \times \sum ETX_{\text{link}}\)
- \(2 \times \sum ETX_{\text{link}}\)
- \(1 \times \sum ETX_{\text{link}}\)
- \(2 \times \sum ETX_{\text{link}}\)
- \(3 \times \sum ETX_{\text{link}}\)
**MASTER vs. Orchestra**

- **MASTER:** Long-Term Stable Routing and Scheduling in Low-Power Wireless Networks

**Graphs:**
- **PDR [%]** vs. **Latency [slots]**
- **Duty Cycle [%]**

- Orange: $3 \times \sum[ETX_{\text{link}}]$ at night
- Orange: $3 \times \sum[ETX_{\text{link}}]$ during daytime
- Green: Orchestra at night
- Green: Orchestra during daytime
MASTER’s Long-Term Stability

\[ 1 \times \sum [ETX_{link}] \]

\[ 2 \times \sum [ETX_{link}] \]

\[ 3 \times \sum [ETX_{link}] \]

**Time of day [hours]**

**PDR [%]**

**Latency [slots]**
Conclusion

• MASTER: centralized router and scheduler for TSCH
• Sliding Windows: flexible and stable transmission strategy
• Implementation available at https://github.com/ds-kiel/master-scheduler
• Long-term stable schedules
• Outperforms Orchestra latency wise
Thank you for your attention

https://github.com/ds-kiel/master-scheduler
oha@informatik.uni-kiel.de