Towards Delay-Minimal Scheduling through Reinforcement Learning in IEEE 802.15.4 DSME

Florian Meyer and Volker Turau

KuVS Fachgespräche: Machine Learning & Networking
February 20th, 2020

Institute of Telematics
Hamburg University of Technology
TUHH

Motivation

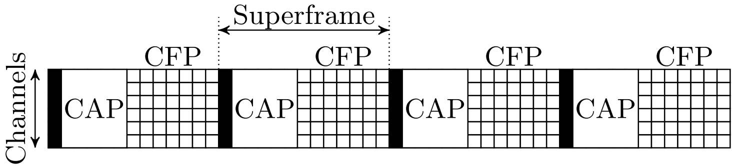
- Increased adoption of WSNs in industrial environments (IIoT)
- Many IIoT applications have tight delay constraints
- Existing protocols (CSMA, IEEE 802.15.4, ...) not suitable
- The Deterministic and Synchronous Multi-Channel Extension (DSME) increases robustness, reliability and scalability:
  - TDMA-based medium access
  - Channel diversity
  - Distributed slot negotiation at runtime

Motivation

- Increased adoption of WSNs in industrial environments (IIoT)
- Many IIoT applications have tight delay constraints
- Existing protocols (CSMA, IEEE 802.15.4, ...) not suitable
- The Deterministic and Synchronous Multi-Channel Extension (DSME) increases robustness, reliability and scalability:
  - TDMA-based medium access
  - Channel diversity
  - Distributed slot negotiation at runtime

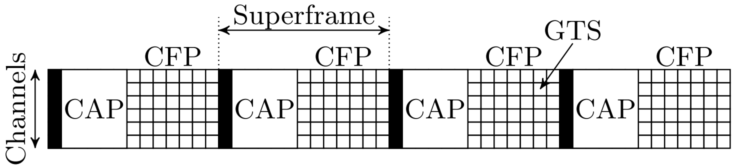
$\Rightarrow$ Find a scheduling strategy for DSME that minimizes global delay using Reinforcement Learning

DSME Frame Structure



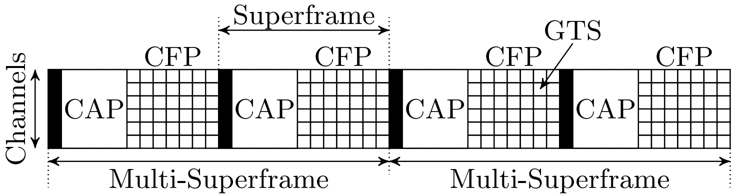
- CAP: Contention access period
- CFP: Contention free period
- GTS: Guaranteed time slot

DSME Frame Structure



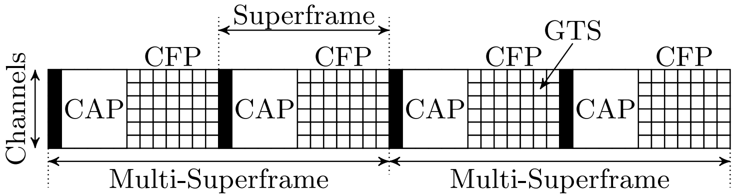
- CAP: Contention access period
- CFP: Contention free period
- GTS: Guaranteed time slot

DSME Frame Structure



- CAP: Contention access period
- CFP: Contention free period
- GTS: Guaranteed time slot

DSME Frame Structure



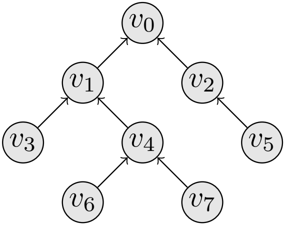
- CAP: Contention access period
- CFP: Contention free period
- GTS: Guaranteed time slot

K: number of GTS per Multi-Superframe

Schedule Definition

Schedule

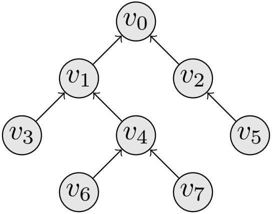
A schedule is an assignment of GTS with specific times and frequencies to pairs of communicating nodes.



Scheduling restrictions:

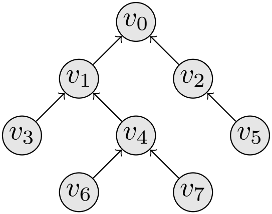
1. In each slot a node can either send a single packet to its parent or receive a single packet from a child, not both.
2. A node can only receive a packet from a single child in a slot. Several packets from different children collide and are corrupted.
3. If in a slot a node v_i sends a packet, its neighbors can only use different channels of the same time slot.

Scenario Description



- Sink v_0
- Every node v_i generates δ_i packets per second
- Generation time τ_j of every packet p_j is known
- Delay of packet p_j: $\phi_j - \tau_j$

Scenario Description

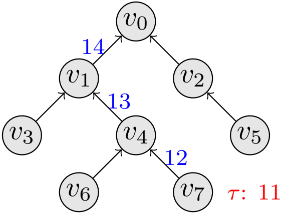


- Sink v_0
- Every node v_i generates δ_i packets per second
- Generation time τ_j of every packet p_j is known
- Delay of packet p_j: $\phi_j - \tau_j$

Find a schedule θ as

$$\arg\min_\theta \sum_{p_j} \phi_j - \tau_j$$

Scenario Description

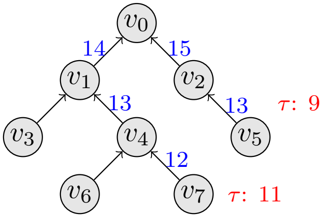


- Sink v_0
- Every node v_i generates δ_i packets per second
- Generation time τ_j of every packet p_j is known
- Delay of packet p_j: $\phi_j - \tau_j$

Find a schedule θ as

$$\arg\min_{\theta} \sum_{p_j} \phi_j - \tau_j$$

Scenario Description

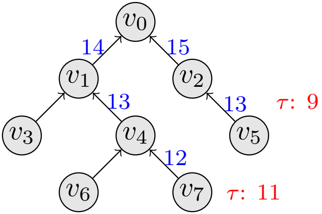


- Sink v_0
- Every node v_i generates δ_i packets per second
- Generation time τ_j of every packet p_j is known
- Delay of packet p_j: $\phi_j - \tau_j$

Find a schedule θ as

$$\arg\min_\theta \sum_{p_j} \phi_j - \tau_j$$

Scenario Description



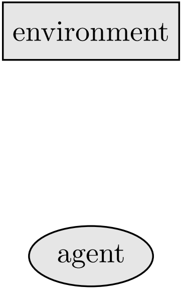
- Sink v_0
- Every node v_i generates δ_i packets per second
- Generation time τ_j of every packet p_j is known
- Delay of packet p_j: $\phi_j - \tau_j$

Find a schedule θ as

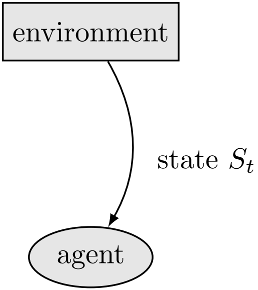
$$\arg\min_\theta \sum_{p_j} \phi_j - \tau_j$$

$\Rightarrow$ Solvable using LPs but too complex

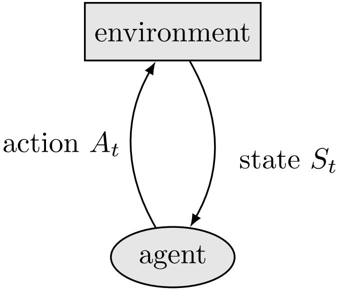
Reinforcement Learning



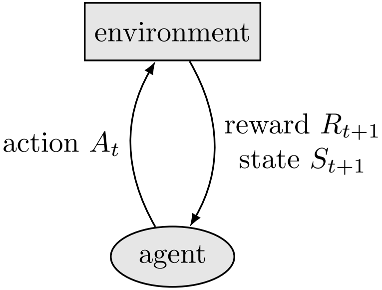
Reinforcement Learning



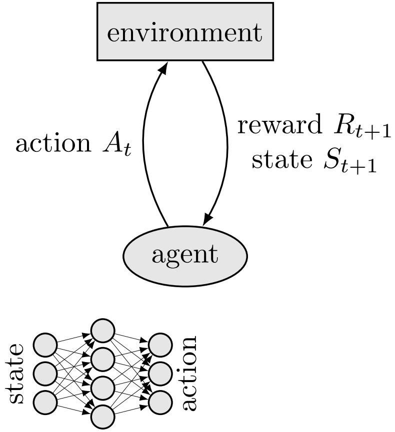
Reinforcement Learning



Reinforcement Learning



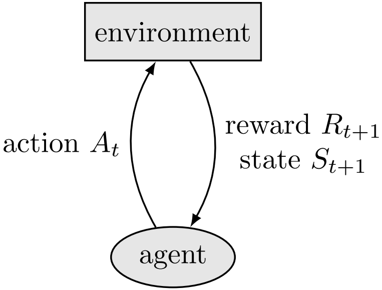
Reinforcement Learning

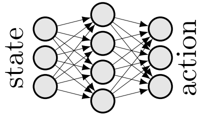


Agent:

- Deep Neural Network (DNN) as scheduler
- Distributed to every node

Reinforcement Learning





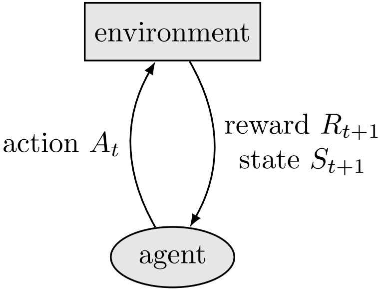
Environment:

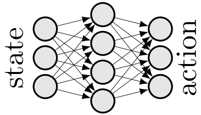
- Network simulation too slow

⇒ Simplified model

⇒ Time on slot basis

Reinforcement Learning



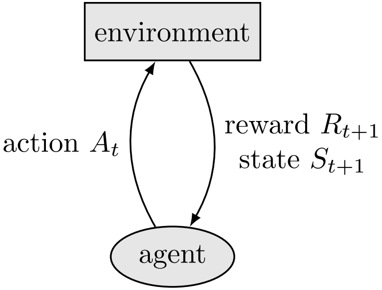


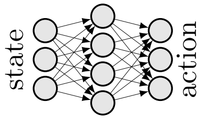
State:

- combined Tx-slots of children
- Tx/Rx-slots of parent
- Tx-slots of neighbors
- δ_i
- queue level
- own schedule
- current slot number

$\Rightarrow$ No communication overhead

Reinforcement Learning

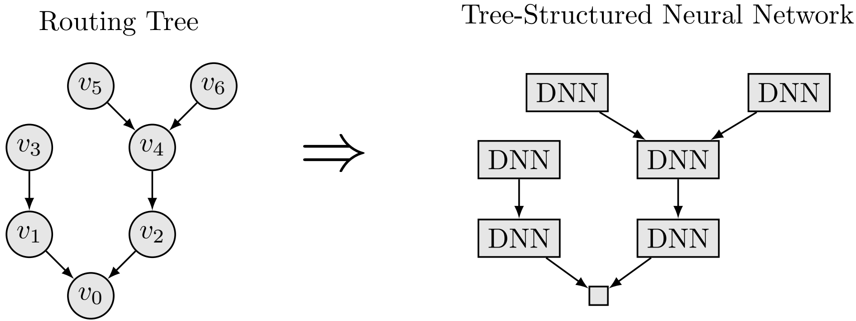




Actions:

- *K* actions for *allocation*
- *K* actions for *deallocation*
- 1 action for *do nothing*

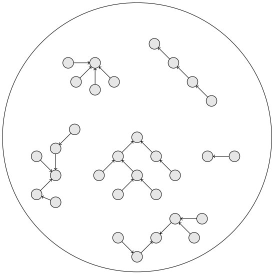
Network Architecture



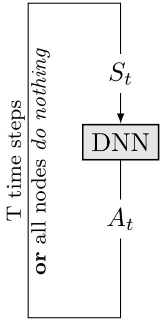
- Weight sharing between DNNs
- Deep Double Q-Learning with Prioritized Experience Replay

Training

Generate random routing trees:



Execute DNN at every node:



Calculate global reward:

if not fair:

$$R = -\infty$$

else:

$$R = -(\gamma_0 \cdot d_{avg} + \gamma_1 \cdot d_{max} + \gamma_2 \cdot N_{TX})$$

d_{avg} : average delay

d_{max} : maximum delay

N_{TX} : # Tx-slots

Discussion & Outlook

- Proposed algorithm always finds valid schedule ...
- ... but only achieves minimum delay for small networks ($N \leq 5$)
- Delay diverges from minimum delay for larger number of nodes
- Other RL-algorithms were tested (Monte-Carlo, Deep Deterministic Policy Gradients, Neuroevolution)
  - Neuroevolution yields better results for larger networks

- Outlook:
  - Verification in OMNeT++ and comparison with TPS
  - Optimization for other metrics (e.g. throughput)
  - Avoid retraining for different frame lengths

Towards Delay-Minimal Scheduling through Reinforcement Learning in IEEE 802.15.4 DSME

Florian

KuVS Fachgesprä

Florian Meyer

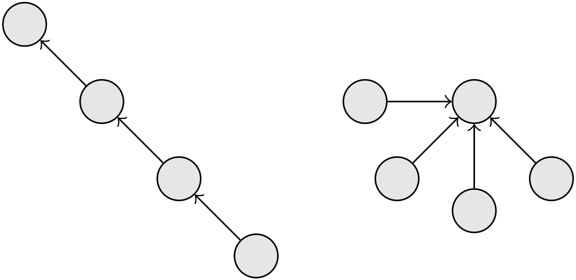
Research Assistant

Phone +49 / (0)40 42878 - 3746

e-Mail fl.meyer@tuhh.de

http://www.ti5.tu-harburg.de/staff/meyer

Training - Extreme Cases



- Extreme cases similar since bottleneck occurs at sink