Multi-objective Reinforcement Learning *for* Energy Harvesting Wireless Sensor Nodes

Shaswot Shresthamali
Keio University

Masaaki Kondo
Keio University

Hiroshi Nakamura
The University of Tokyo

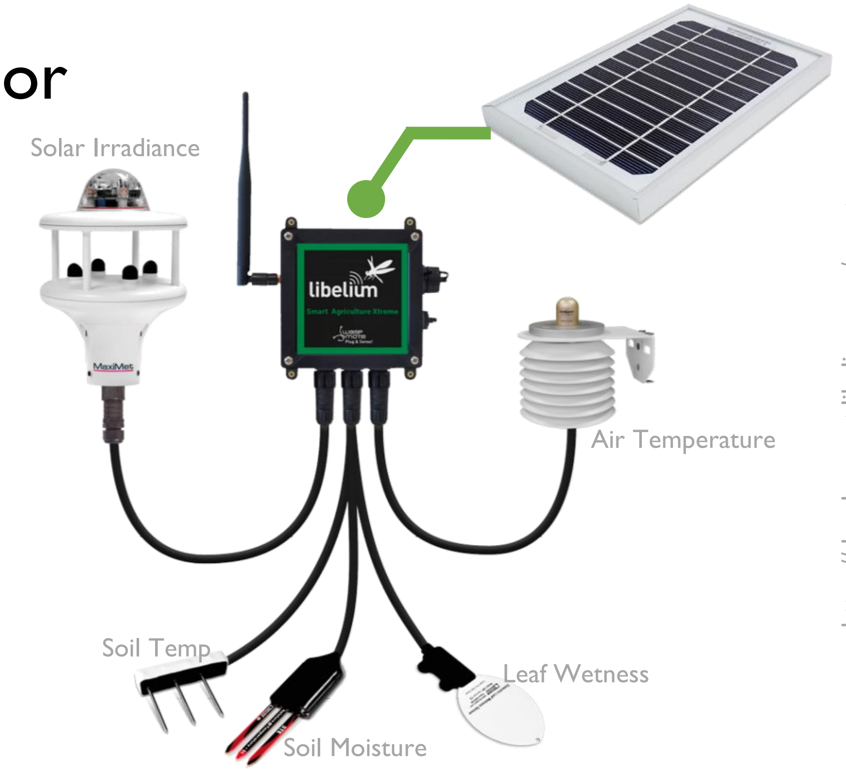
IEEE 14th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC-2021)

21-December-2021

Billions of Autonomous Sensor Nodes

- Energy Harvesting Wireless Sensor Nodes (EHWSNs)
  - sense, process and transmit data wirelessly
  - harvest energy from the ambient environment

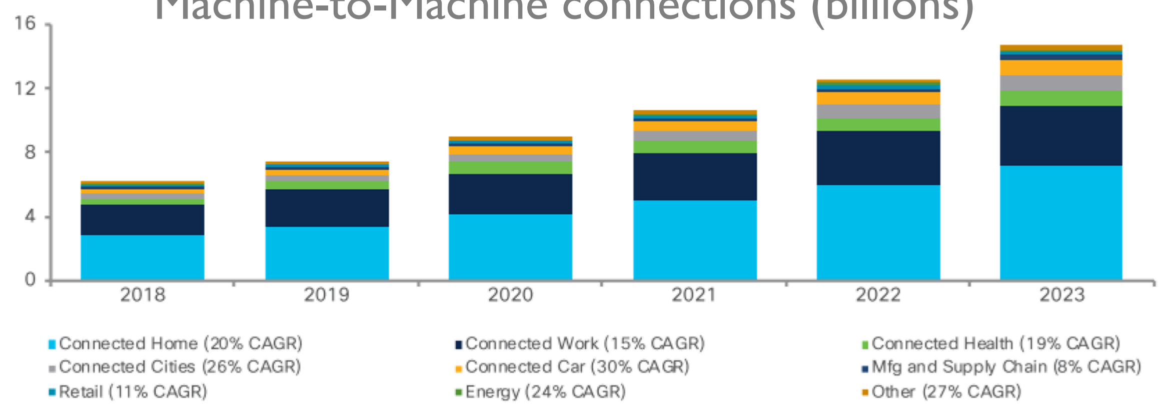
Autonomous
Perpetual
Operation



https://development.libelium.com/smart-agriculture-xtreme-sensor-guide/

- Millions of usage scenarios (deploy-and-forget)
- Massive scaling

Machine-to-Machine connections (billions)

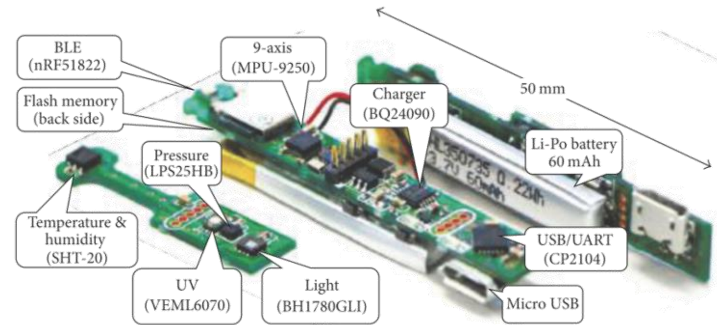


Source: Cisco Annual Internet Report, 2018-2023

Energy Neutrality and multiple objectives

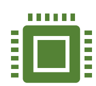
Energy Neutral Operation

- Balancing energy generation and consumption

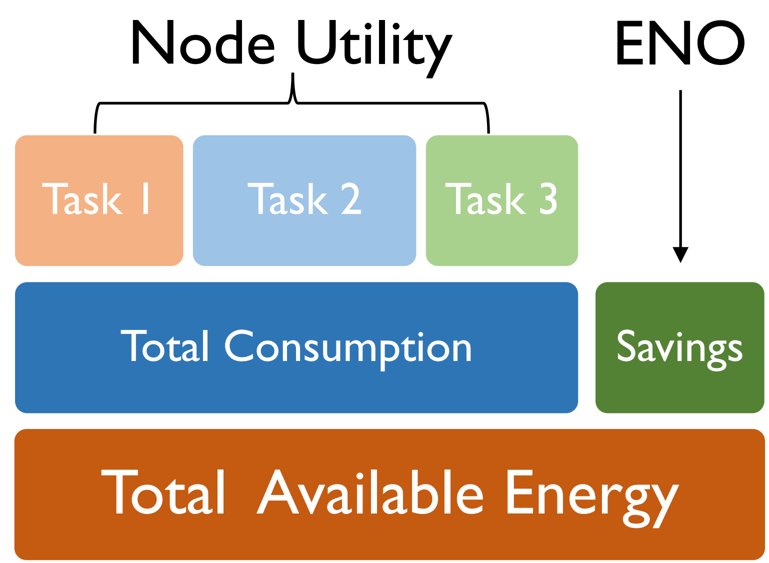


[Nakamura et al 2017, SenStick: Comprehensive Sensing Platform with an Ultra Tiny All-In-One Sensor Board for IoT Research]

Modern EHWSNs are **multi-task**

- Sensing 
- Processing 
- Communication 

- Allocate energy among tasks
  - To maximize utility
  - According to user priority
    - Only known during runtime (not a priori)



Optimizing over multiple objectives

- Use Multi-objective Reinforcement Learning (MORL) to learn the energy management policy

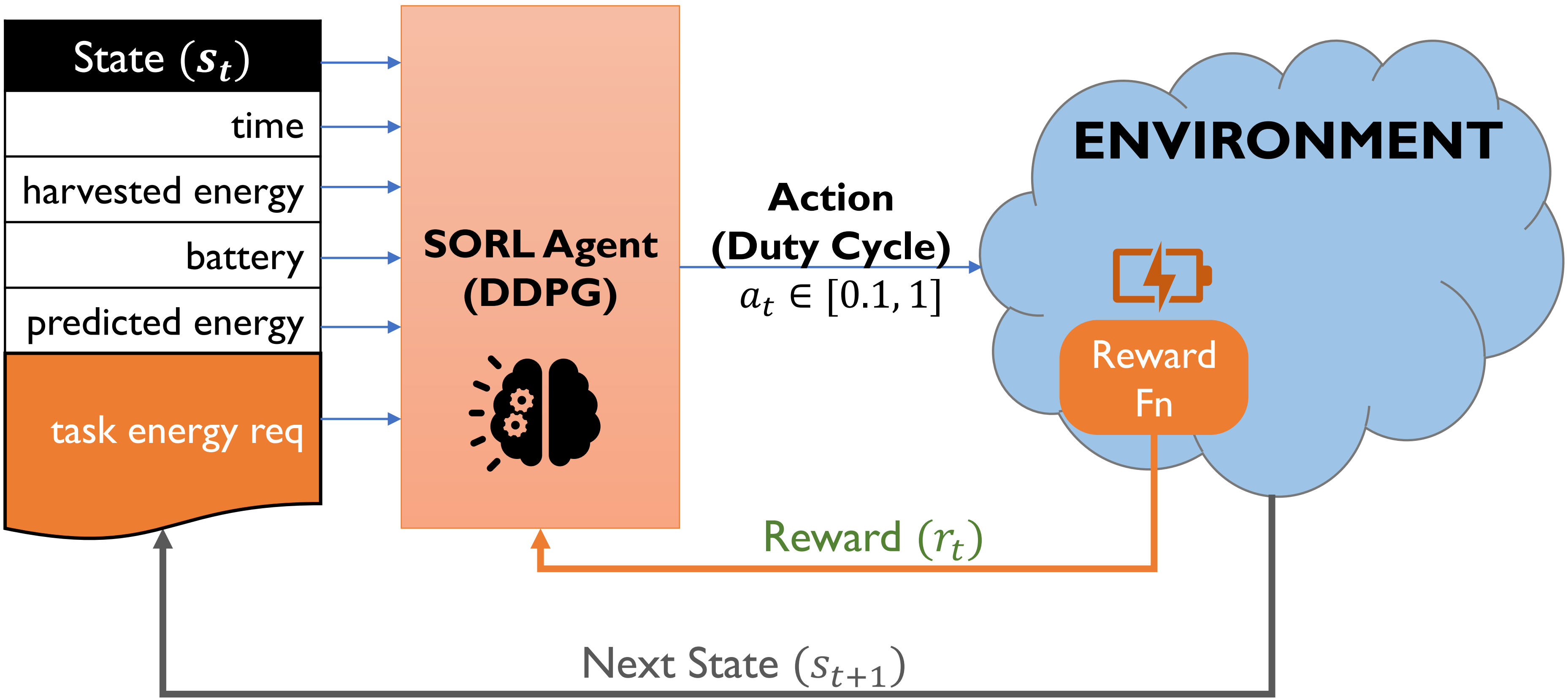
Proposal

- General Multi-Objective MDP Formulation
  - Continuous States and Actions
  - Single/Multiple rewards
- Novel low-compute MORL algorithms

Results

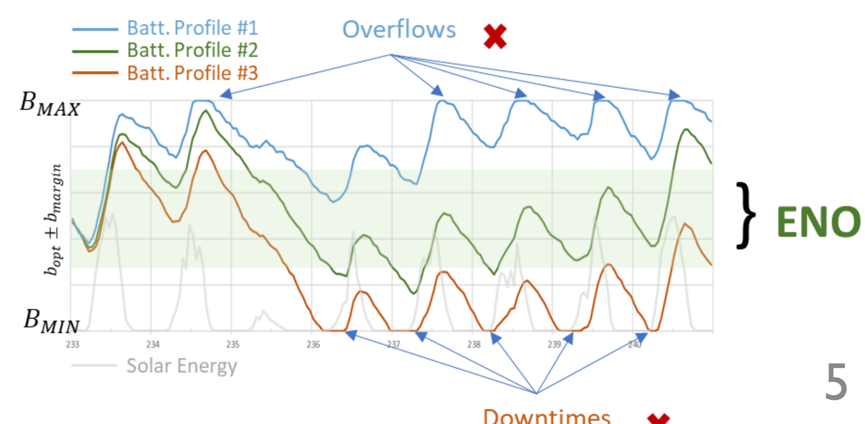
- Near-optimal policies
  - Better than scalarization methods
- Low learning costs
- Runtime Tradeoffs

Single-objective RL (SORL)

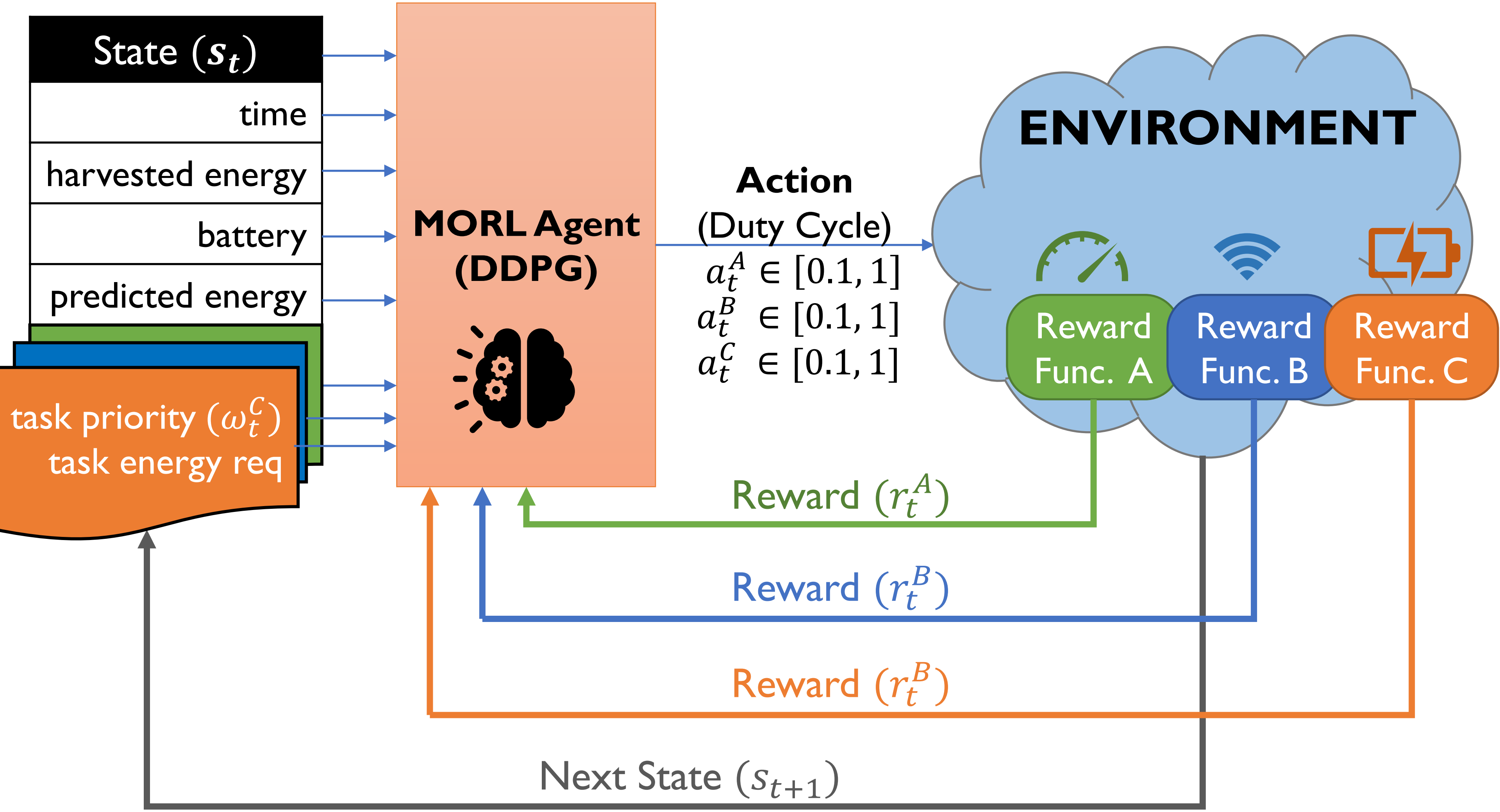


Shresthamali et. al., **Adaptive power management in solar energy harvesting sensor node using reinforcement learning**, EMSOFT 2017.

Shresthamali et. al., **Power Management of Wireless Sensor Nodes with Coordinated Distributed Reinforcement Learning**, ICCD 2019.



Multi-objective RL (MORL)



Node utility, $\mathrm{w_t} = \omega_t^A \times r_t^A + \omega_t^B \times r_t^B + \omega_t^C \times r_t^C$

3. Optimizing over Multiple Objectives

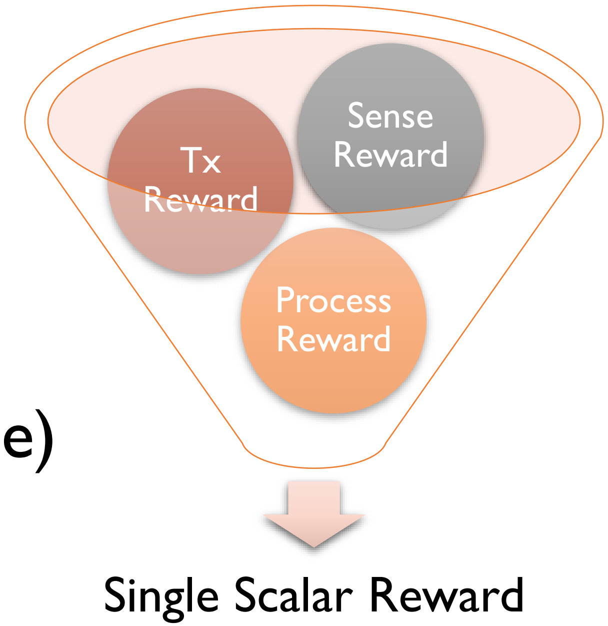
https://www.fm-magazine.com/news/2019/nov/resource-allocation-best-practices-201922378.html

Challenges of learning with many rewards

Scalarization

Mix rewards into one scalar

- *Aoudia et al., 2018*:
  - Multiply rewards together
  - Confounds rewards (noisy interference)
- *Ferreira et al., 2018*:
  - Add rewards together
  - Tradeoffs not possible
  - Prior knowledge of relative weights required
- *Hsu et. al., 2014:*
  - Complicated unintuitive reward function
  - Unexpected behavior (reward hacking)



Proposed MORL Framework

1. General Multi-Objective MDP Formulation
   - Rewards
     - Simpler, precise, intuitive rewards
     - Multiple sources of rewards (reward vector)
   - States
     - Continuous
     - Inclusion of temporal information (more Markov)
   - Actions
     - Continuous
     - Relative actions (Safe actions)

For efficient learning

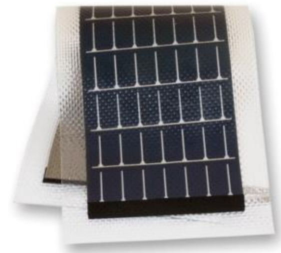
2. Low compute MORL algorithms (based on DDPG)

   [Lillicrap et. al., 2015]

   - Runtime MORL
     - Runtime tradeoffs using pre-learned greedy policies
   - Off-policy MORL
     - Learn tradeoff policies from scratch with low learning costs

Simulation Parameters

Component	Rating
Battery (Rechargeable Li-Ion)	**2000 mAh**
Solar Panel	**100 mA**



Foldable Solar panel
(100 mAh)

Li-Ion Battery
(2000mAh)

Device	Rating
Waspmote Sensor Platform ATmega1281@15 MHz)	17 mA
Comm. Device (Zigbee 3)	40 mA
Sensor (GPS)	32 mA
TOTAL	**89 ~ 100 mA**

Waspmote

Parameter	maximum value	minimum value
Battery, b_t	b_{max}	$b_{min} = 10\%$ of b_{max}
Harvester, h_t	$h_{max} = 5\%$ of b_{max}	$hmin = 0$
EHWSN, z_t	$z_{max} = 5\%$ of b_{max}	$z_{min} = 0.5\%$ of b_{max}
Requests, d_t	$z_{max} = 5\%$ of b_{max}	$z_{min} = 0.5\%$ of b_{max}

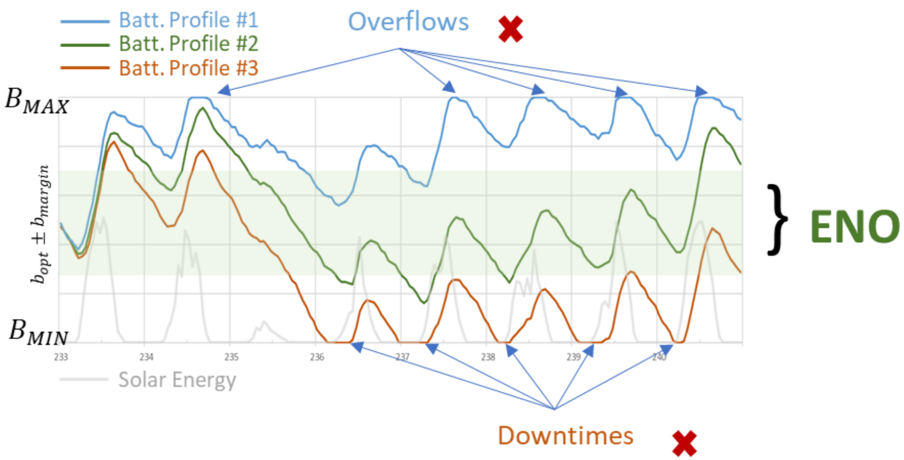
https://development.libelium.com/waspmote-technical-guide/
https://weffsas.com/shop/lithium-ion-battery-3-7v-2000mah/

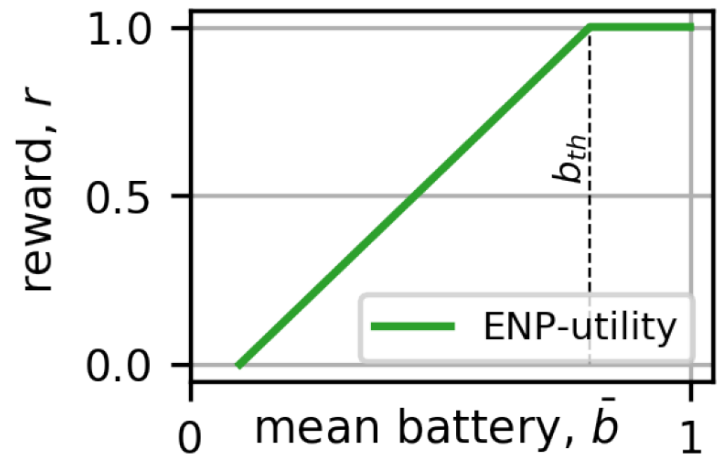
Experimental Results

- Near-optimal Performance
    - Superior to scalarization methods
- Dynamic Runtime Tradeoffs
- Low learning cost
    - Safer Exploration
    - Faster learning

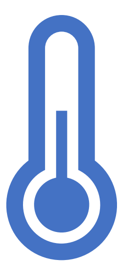
Multiple Objectives

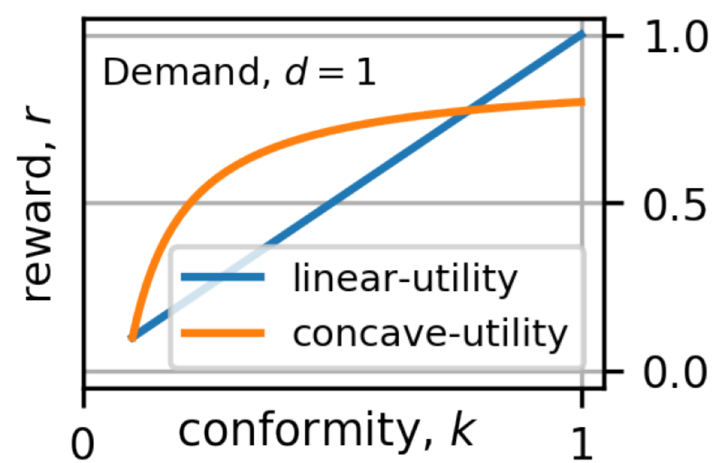
Energy Neutral Performance (ENP)





Sensing and Transmission





Energy Scheduling

- At each timestep, the node allocates energy budget for
  - Sensing (w.r.t. user requirements)
  - Transmission (w.r.t. channel conditions)

- By taking the user-defined relative priority (ω) between
  - Sensing (ω_{sense})
  - Transmission (ω_{tx})
  - Energy-neutral performance (ω_{ENP})

- While ensuring long-term energy neutrality
  - Lower Downtimes = Better Energy Neutrality

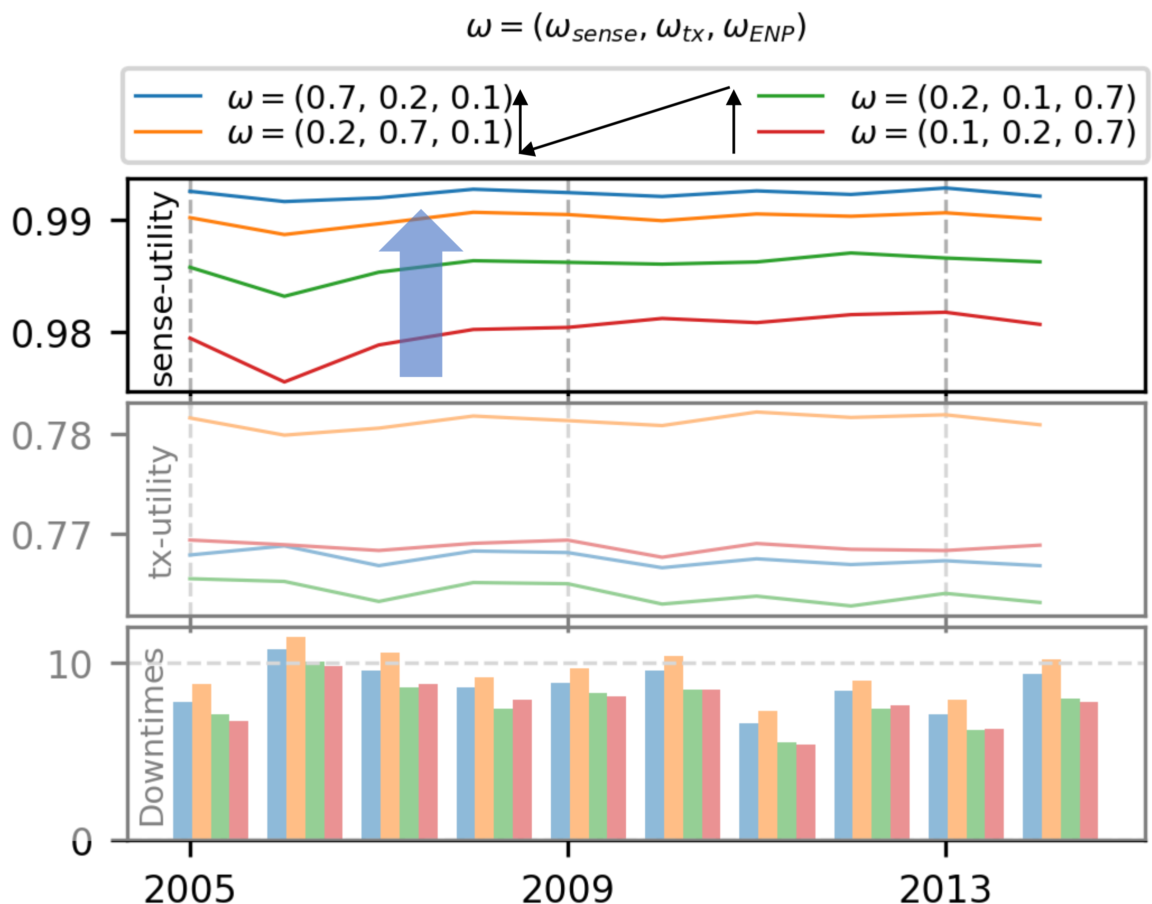
Runtime Tradeoffs (2-tasks)

Off-policy MORL Algorithm

- Tradeoff between sensing, transmission and energy-neutrality

Increasing ω_{sense} increases *sense-utility* correspondingly:

red → green → orange → blue



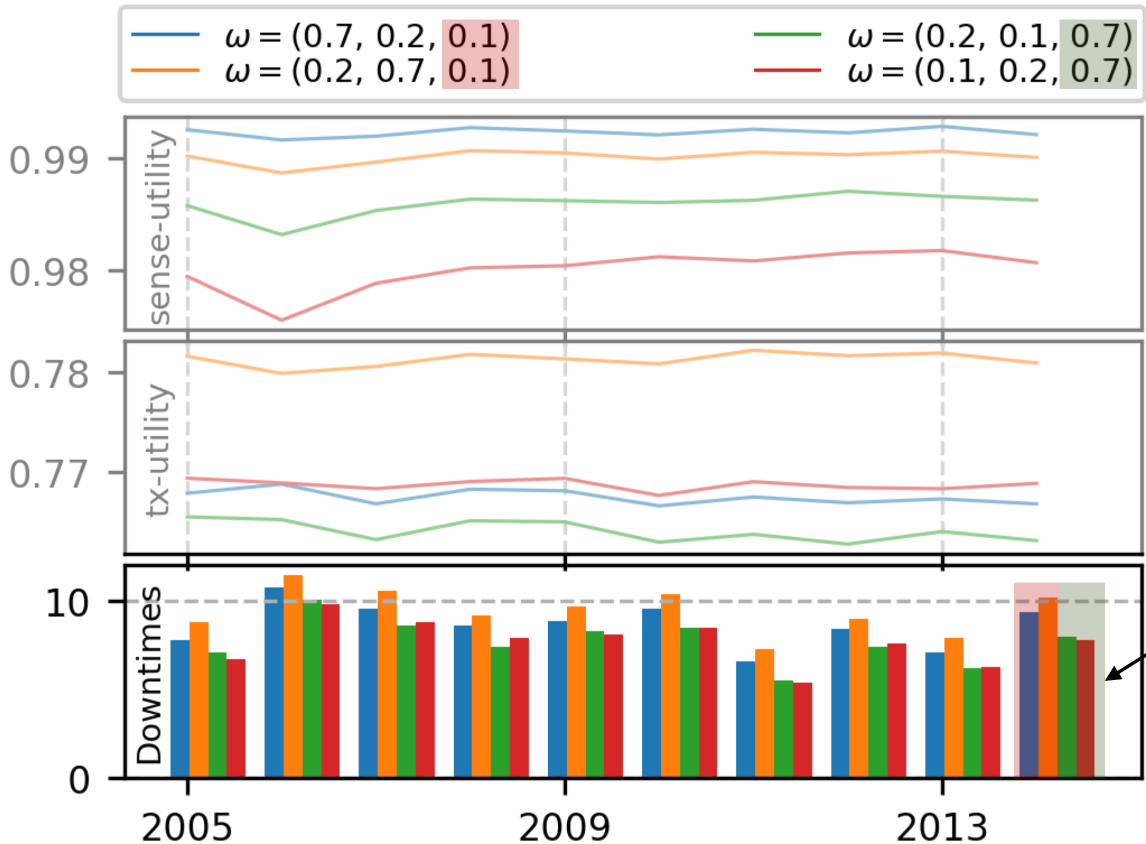
Runtime Tradeoffs (2-tasks)

Off-policy MORL Algorithm

- Tradeoff between sensing, transmission and energy-neutrality
- $\omega = (\omega_{sense}, \omega_{tx}, \omega_{ENP})$

Less priority for ENO

More priority for ENO



Higher ω_{ENP} (green/red) ensure lower downtimes than lower ω_{ENP} (blue/orange)

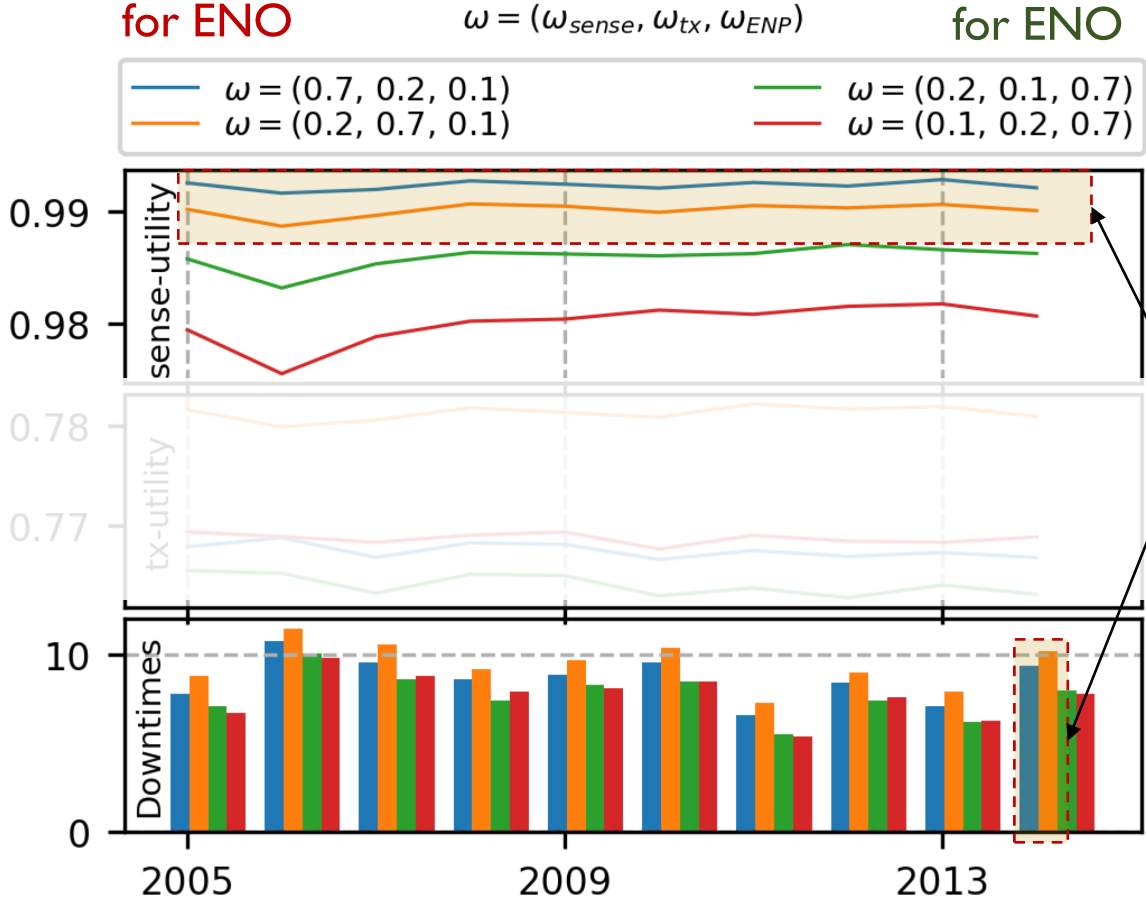
Runtime Tradeoffs (2-tasks)

Off-policy MORL Algorithm

- Tradeoff between sensing, transmission and energy-neutrality
- $\omega = (\omega_{sense}, \omega_{tx}, \omega_{ENP})$

Less priority for ENO

More priority for ENO

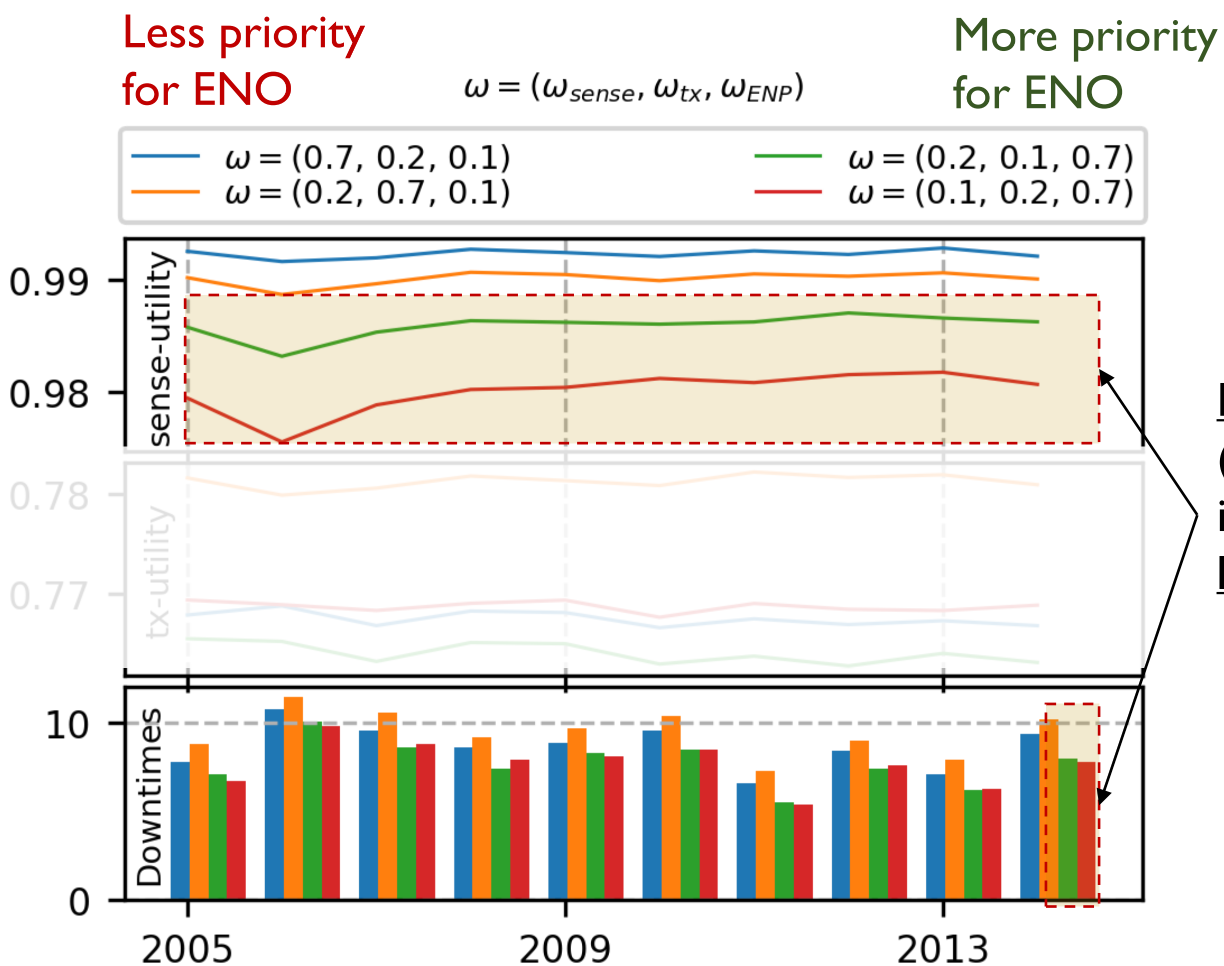


<u>Higher sensing utility</u> is achieved by trading off with <u>lower energy neutrality</u> (i.e., higher downtimes)

Runtime Tradeoffs (2-tasks)

Off-policy MORL Algorithm

- Tradeoff between sensing, transmission and energy-neutrality
- $\omega = (\omega_{sense}, \omega_{tx}, \omega_{ENP})$



Better energy neutrality
(i.e., higher downtimes)
is achieved by trading off with
lower sense-utility

Conclusion

- EHWSNs require RL based methods for
  - Adaptive, scalable policies

- RL is difficult for EHWSNs due to
  - Difficult problem formulation
  - High learning costs
  - High computation costs
  - **Multiple objective optimization problem**

- Traditional MDPs and scalarization methods
  - Are sub-optimal
  - Have high-learning costs
  - Cannot tradeoff
  - Require complicated reward functions

- Proposed MORL framework
  - Can learn near-optimal policies
  - With low learning costs
  - Can tradeoff at runtime
  - Simple/precise and diverse rewards can be used

Shaswot SHRESTHAMALI
https://www.shaswot.com

Masaaki KONDO
https://www.acsl.ics.keio.ac.jp/

Hiroshi NAKAMURA
http://www.hal.ipc.i.u-tokyo.ac.jp

慶應義塾
Keio University

東京大学
THE UNIVERSITY OF TOKYO

独立行政法人
科学技術振興機構
Japan Science and Technology Agency
CREST

日本学術振興会
Japan Society for the Promotion of Science

Thank You

Your questions/comments and feedback are most welcome