DEWDROP: AN ENERGY-AWARE RUNTIME FOR COMPUTATIONAL RFID

Michael Buettner (UW), Benjamin Greenstein (Intel Labs, Seattle), David Wetherall (UW)

UNIVERSITY OF WASHINGTON

intel

Key Question

How can we run programs on embedded computers using only scavenged RF energy?

Battery free, "invisible" sensing and computation is key to truly ubiquitous computing applications

Scenario: Activity Recognition for Elder Care

Elders can stay at home longer if caregivers know they are safe

If we know what (and how) objects are used we can determine activities

- Taking medicine, making a meal

What we want: A non-intrusive way to gather data on object use

Existing Solutions

Cameras: Remote monitoring

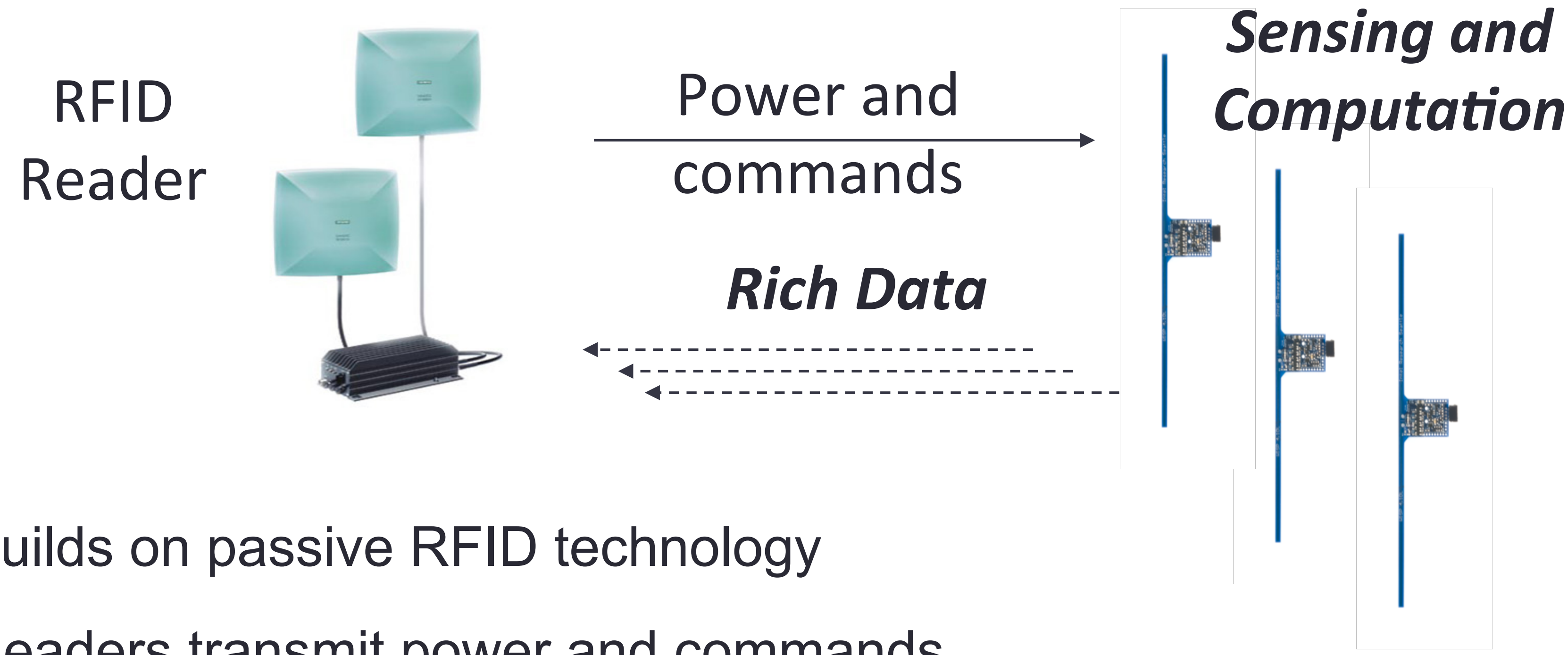
- **Cons:** Obvious privacy concerns

“Mote” based sensor networks: Detect object use from accelerometer data

- **Cons:** Batteries limit deployment
  - **Size**
  - **Lifetime**
  - **Cost**

Infeasible to deploy motes on 10s of everyday objects

Proposed Solution: Computational RFID



- Builds on passive RFID technology
- Readers transmit power and commands
- Battery-free tags harvest RF to compute, sense, communicate
- Prototype hardware now becoming available
  - Goal: RFID tag “sticker” form factor costing $1

Dewdrop: A Runtime for CRFIDs

Enables CRFID tags to use the scarce available energy to run programs:

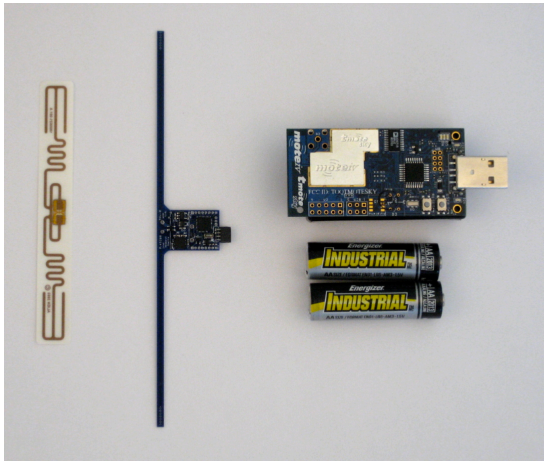
- With varied and non-deterministic energy needs
- When input power varies by two orders of magnitude

Dewdrop runs programs at close to their maximum rate, and where they could not otherwise run

Outline

- Intel WISP – A CRFID Tag
- Challenges to Running Programs Efficiently
- Dewdrop Design
- System Evaluation

Intel **W**ireless **I**dentification and **S**ensing **P**latform

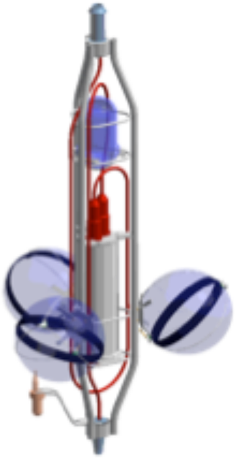


- Features
  - 16-bit TI MSP430, 8K flash
  - 3D accelerometer, light, temp
  - 10 uF capacitor for energy storage
  - 4 m range with standard readers
- Community
  - In use at 30+ universities, ~50 publications

WISP Applications

- Exercise, sleep monitoring
  - [Borrielo 2008, Stankovic 2010]
- Neural monitoring, medical implantables
  - [Yeager 2010, Halperin 2008]

- Cold-chain, undersea neutrino detector
  - [Yeager 2007, Trasatti 2011]
- RFID security
  - [Fu 2009, Kohno 2008]



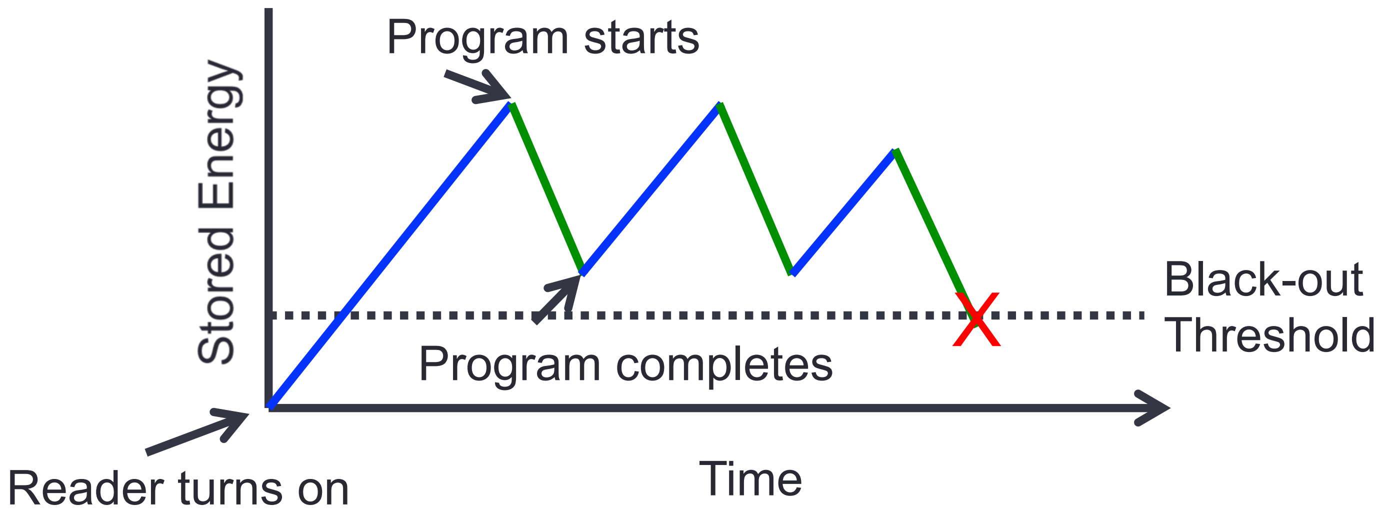
- CRFID programmability
  - [Ransford 2011, Gummeson 2010]

- ***Most use WISPs < 1 m from reader where energy is plentiful***

Challenges to Running Programs Efficiently

1. CRFIDs have miniscule energy stores
2. Programs have different energy needs
3. Platform inefficiencies
4. Energy is harvested even while executing

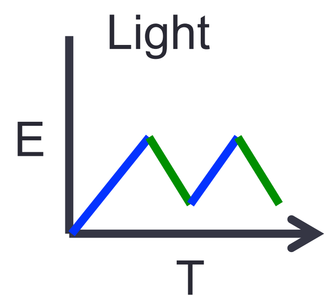
CRFIDs have miniscule energy stores

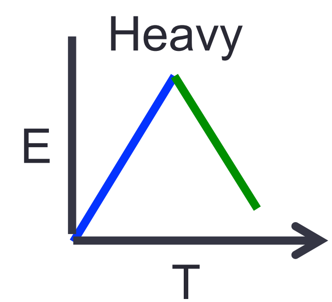


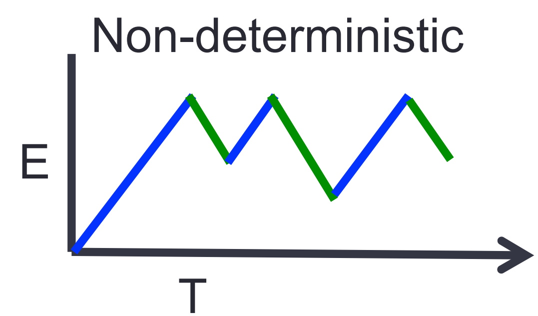
- Low power mode (~1uA) to store energy, maintain state
  Active mode (~100s of uA) to compute and sense
- 100s of ms to charge, 10s of ms to discharge
- **Tags must store enough energy to complete program before beginning execution**

Programs have different energy needs

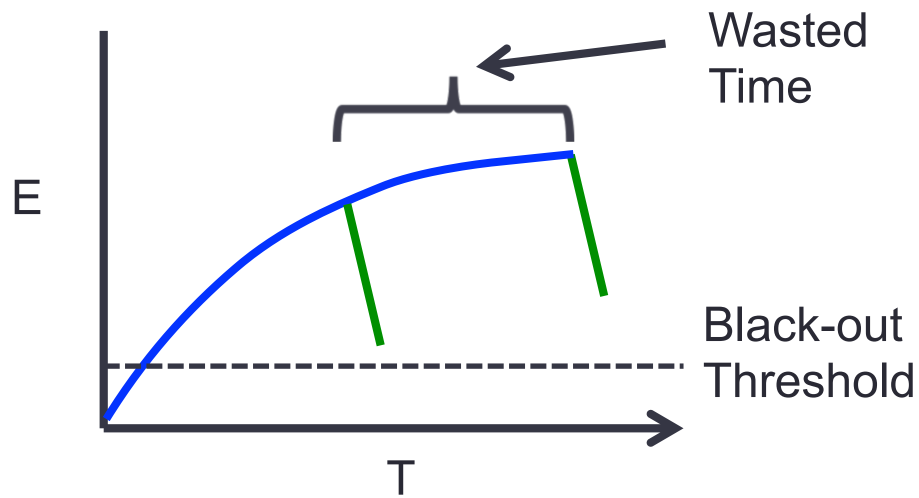
- Wide range of energy needs
  - E.g., Sense, sense and communicate
- May be non-deterministic
  - E.g., RFID MAC protocol
- Run-to-completion
  - E.g., communication, sampling sensors
- Tags run only one program at a time
- ***Tags must store different amounts of energy when running different programs***





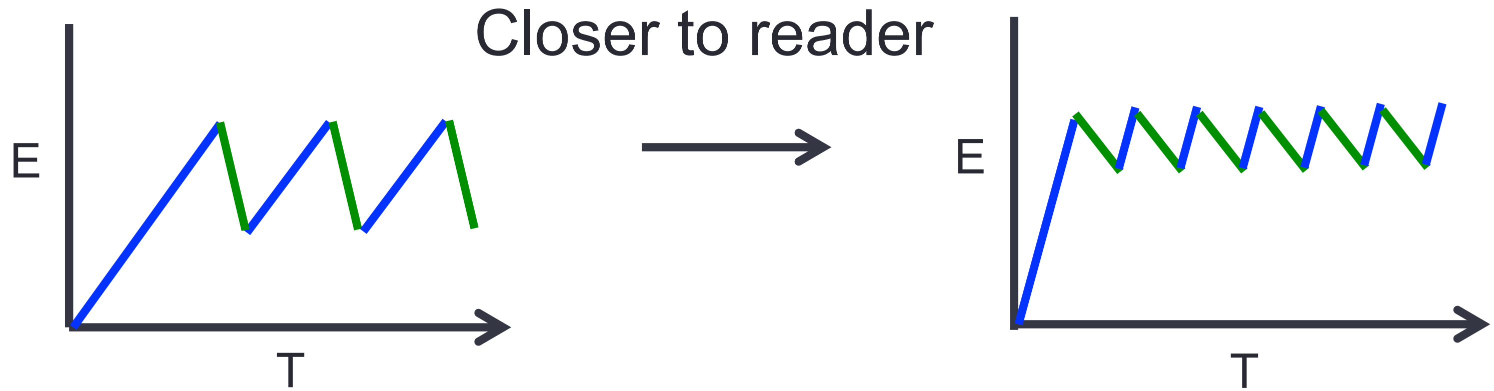


CRFIDs have inefficiencies



- The more stored energy, the longer it takes to store additional energy
  - CRFIDs use capacitors as they are small and can recharge indefinitely
- Voltage regulation ➔ inefficient to operate with more stored energy
- ***Storing excess energy is inefficient***

Energy is harvested even while executing



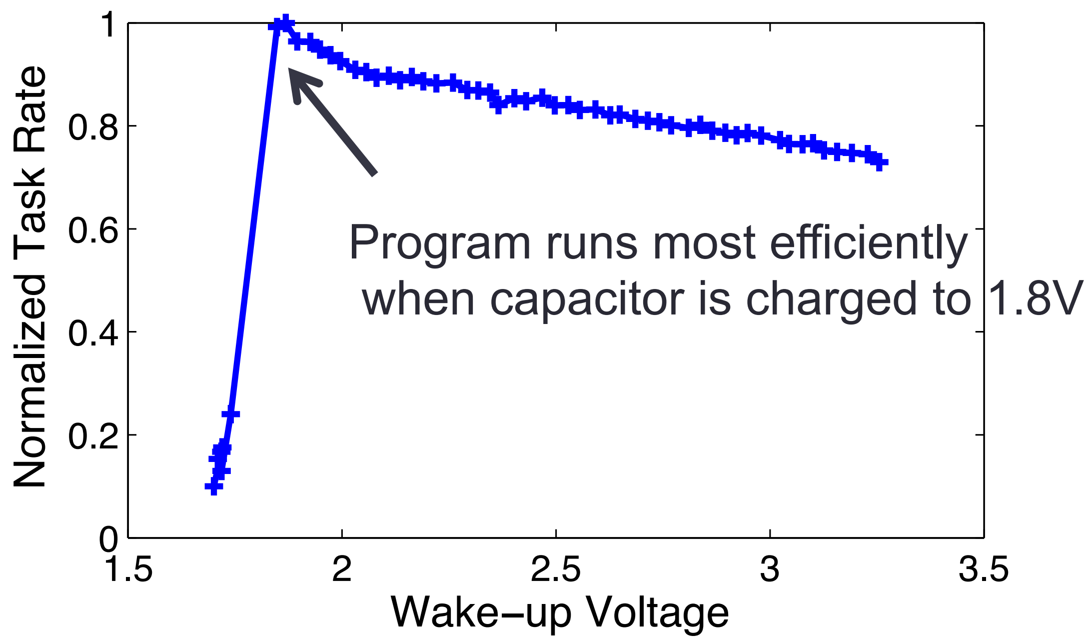
- Received power supplements stored energy
  - Reader frequency hopping ➔ power changes every 400 ms

- ***The amount of stored energy required depends on the distance from the reader and RF environment***

Challenges to Running Programs Efficiently:

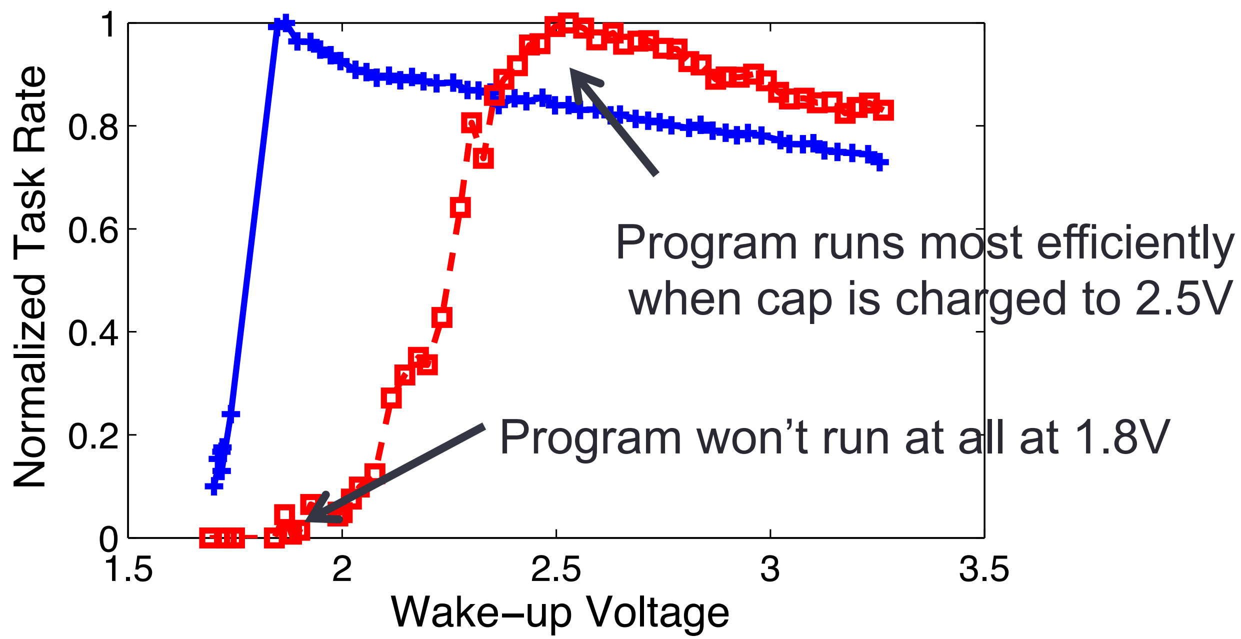
Implications

Storing the right amount of energy increases performance



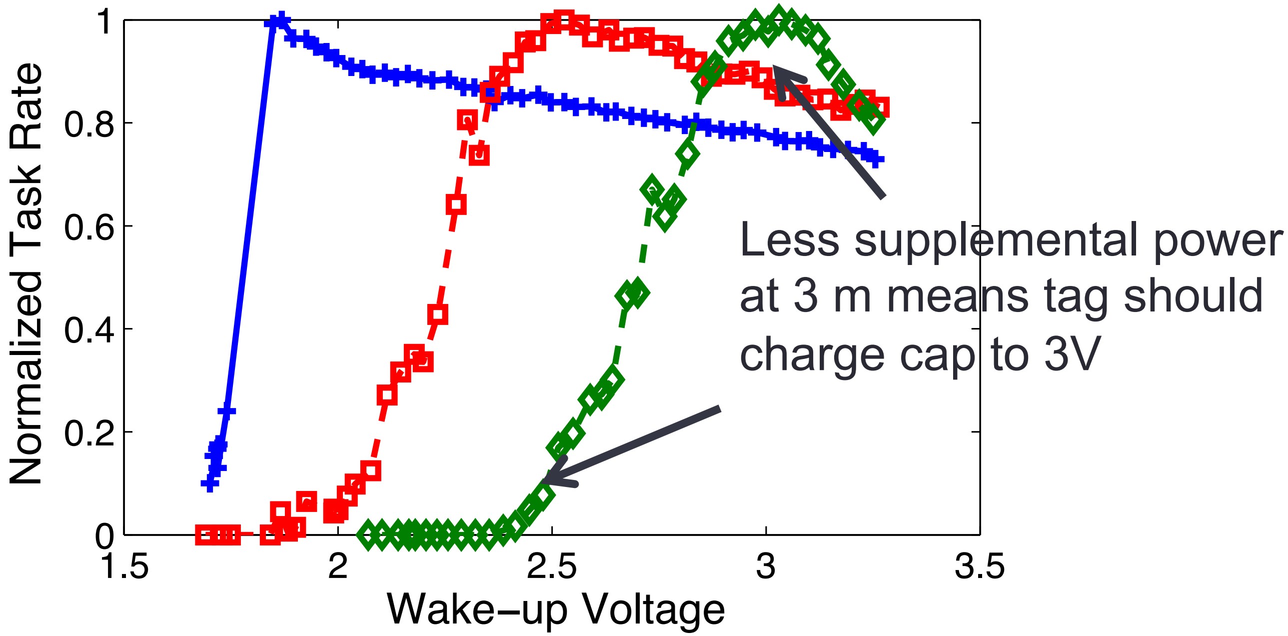
- Wake-up voltage: Determines the amount of energy stored before starting program
- Light WISP program: sample accelerometer, 1.5 m from reader

No fixed threshold works for all programs



- Heavy, non-deterministic program: sample accelerometer and transmit value to reader, 1.5 m

No fixed threshold works for all distances



- Heavy, non-deterministic program, **3 m from reader**
- **CRFID must adapt to program needs and environment**

Outline

- Intel WISP – A CRFID Tag
- Challenges to Running Programs Efficiently
- Dewdrop Design
- System Evaluation

Dewdrop: An Energy-aware Runtime

Adaptively find the **wake-up voltage** that **maximizes execution rate** for the **program** and **RF environment**

Two factors that reduce execution rate:

- Not storing enough energy: Program fails and it takes time to recharge and execute again
- Storing too much energy: Overcharging wastes time

Constraint: Runtime operation must be simple

- Every active cycle costs energy
- No floating point, no hardware multiply/divide

Adapt to the Program and Environment

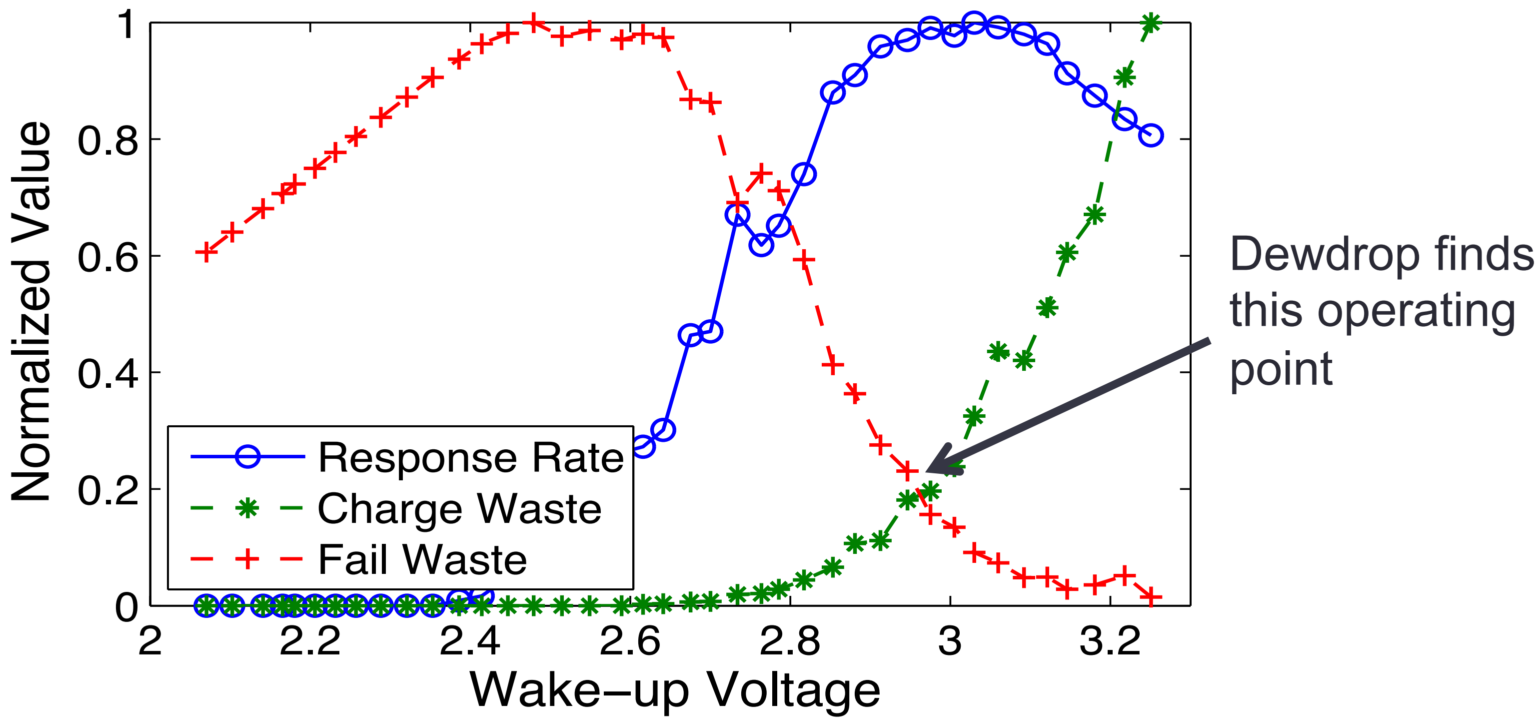
- **Goal:** Maximize execution rate →
  Minimize time wasted from program failure and overcharging

- **Heuristic:** Total waste is minimized when the wasted time from failures and overcharging is equal

- **On program complete:**
  Update running average of time wasted overcharging
- **On program failure:**
  Update running average of time wasted failing

- **If $Avg_{overcharge}$ > Avg_{fail}:** decrease wake-up threshold by β
- **Else:** increase wake-up threshold by β

Heuristic results in a good operating point



- **Equalizing the sources of wasted time results in efficient program execution**

Dewdrop Implementation

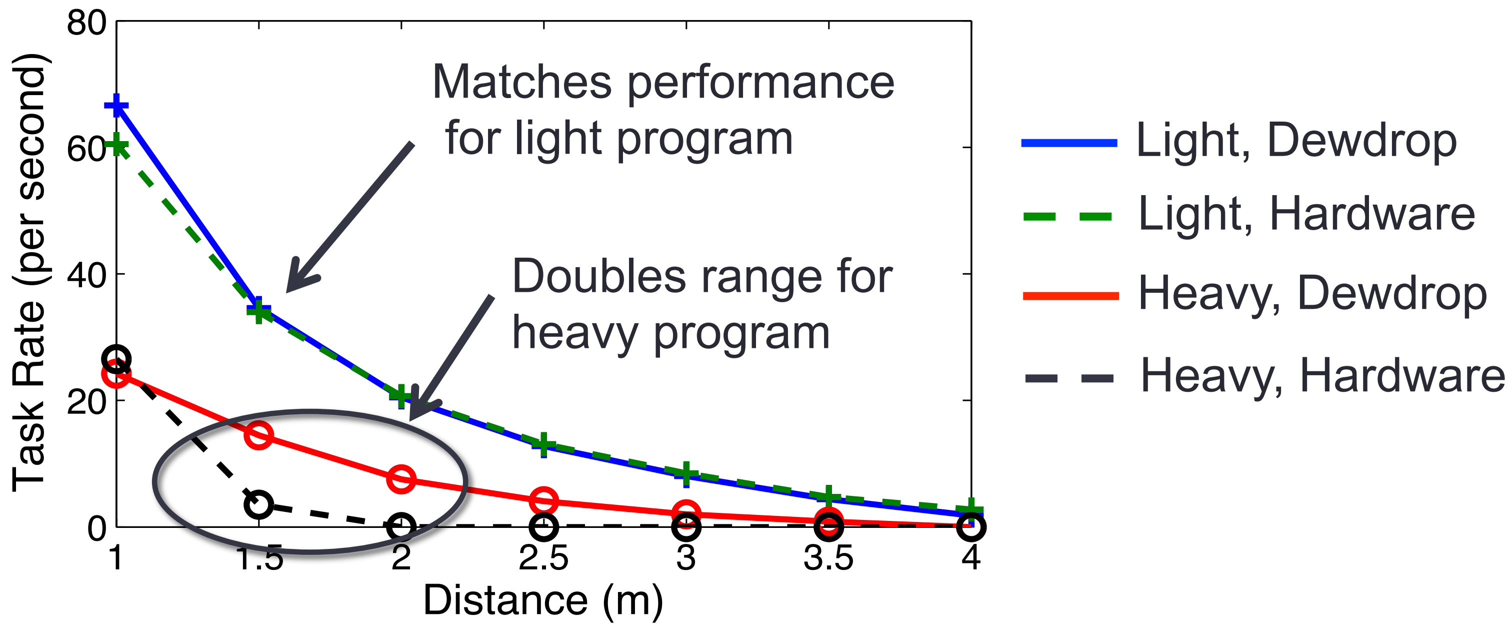
1. Low power wake-up
   - No hardware mechanism to wake up at specified voltage
   - Dewdrop polls capacitor voltage periodically until target is reached
   - Exponentially adapted polling interval is lightweight and accurate

2. Low power voltage sampling
   - Waking up to sample voltage consumes precious energy
   - We reduced the energy cost of voltage sampling by a factor of 4

More details in the paper

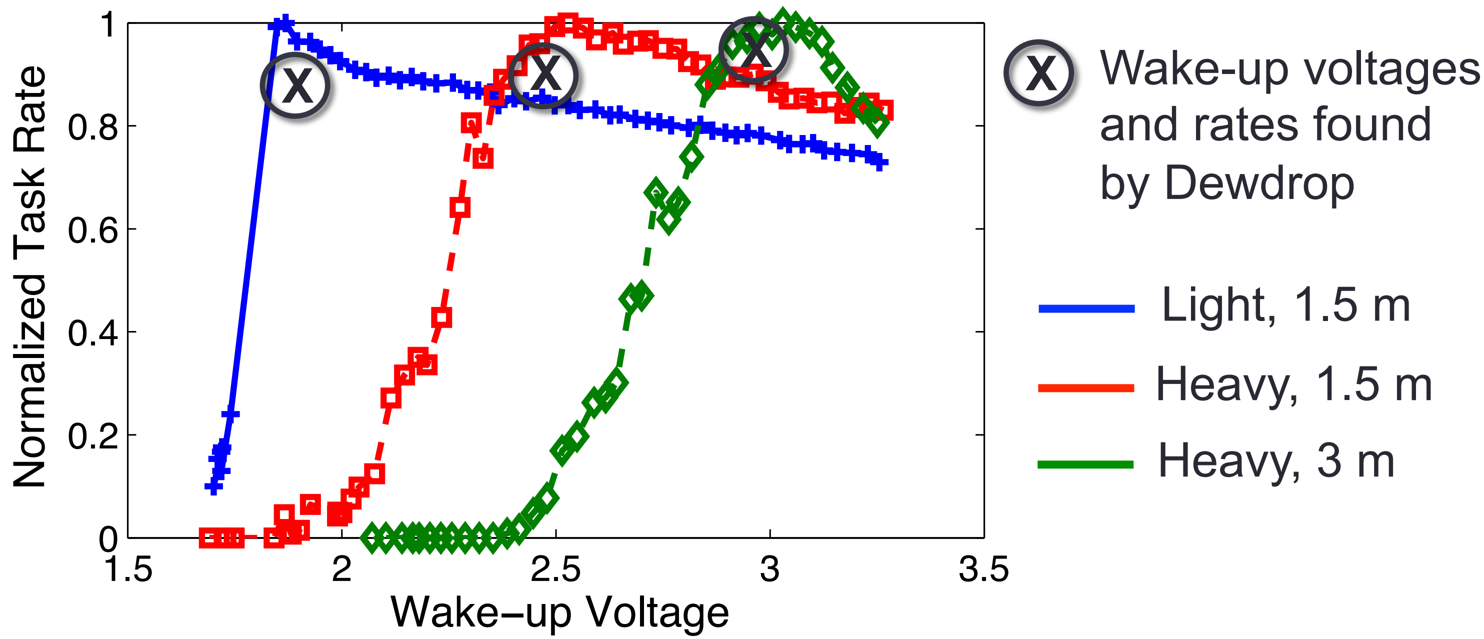
System Evaluation

Dewdrop makes good use of scarce energy



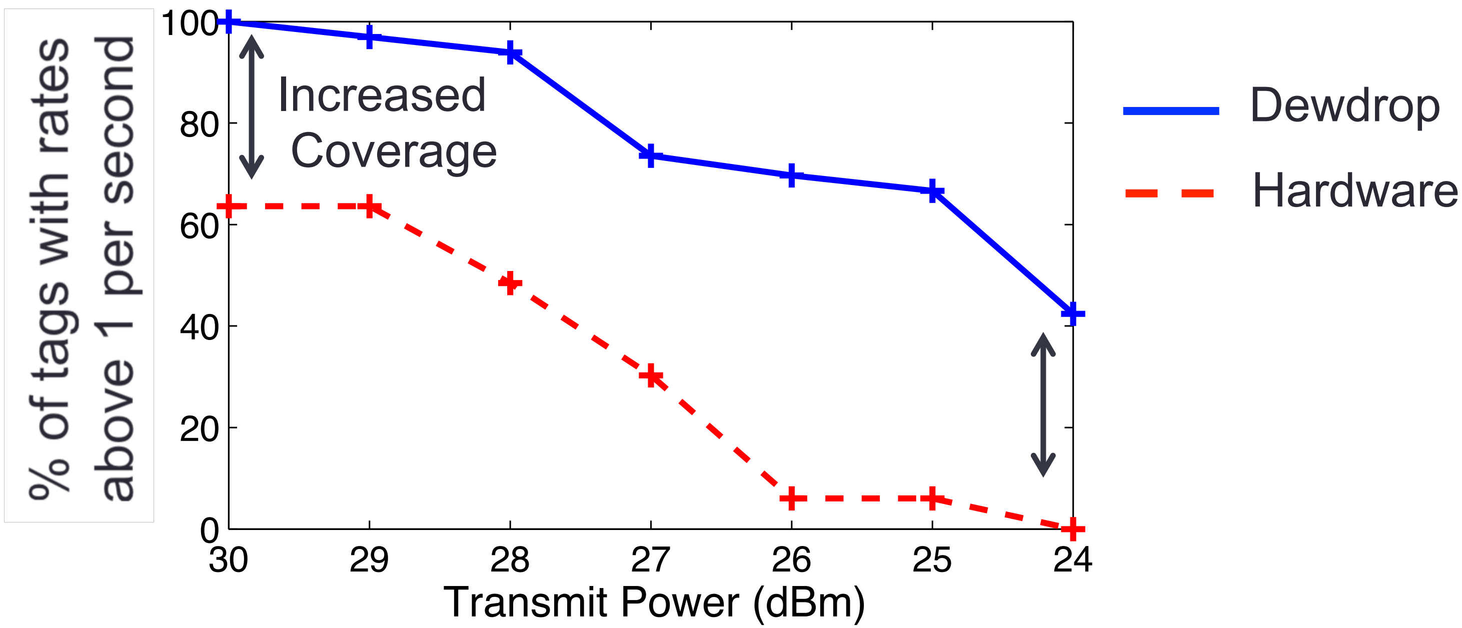
- Compare to efficient, but inflexible, hardware mechanism
  - State-of-the-art before Dewdrop
- Execution rate should scale with received power: $1/d^2$

Dewdrop finds an efficient operating point



- Dewdrop finds wake-up voltage within 0.1V of best
- Generally achieves > 90% of max rate for all distances

Dewdrop increases application coverage



- Elder care scenario: 1 reader, tagged objects in apartment
- 11 WISPs streaming accelerometer data (3 trials)
- **Dewdrop can run the program with much less power**

Conclusion

- Running programs using harvested RF energy is feasible
  - Batteryfree ➔ small, perpetual, embeddable
- Dewdrop makes CRFIDs more usable and useful
- Technology trends will increase range and performance
  - Passive device range expected to continue doubling every 4 years
  - WISP 5.0 in development
- WISPs and tools are available to the community
  - WISP hw/sw open source, USRP-based RFID reader

Questions

- WISP Wiki: wisp.wikispaces.com
- UW Sensor Systems Group: sensor.cs.washington.edu

- www.cs.washington.edu/homes/buettner
- buettner@cs.washington.edu