From Sleeping to Stockpiling: Energy Conservation via Scheduling in Wireless Networks

David Shuman and Mingyan Liu
University of Michigan

Information Theory and Applications Workshop

February 11, 2009

Thanks also to Aditya Mahajan and Ashutosh Nayyar for many helpful discussions

Introduction

Energy conservation is a key design issue in wireless networks in general, and specifically in wireless sensor networks

- Limiting the Idle Time of a Node's Radio
  - Sleep Scheduling for a Single Sensor Node
  - Clock Calibration for an Ultra-Low Power Sensor
- Reducing the Network's Workload
  - Soil Moisture Smart Sensor Web
- Exploiting the Spatial and Temporal Variation of the Wireless Channel
  - Transmission Scheduling with Strict Underflow Constraints

Introduction

Energy conservation is a key design issue in wireless networks in general, and specifically in wireless sensor networks

- Limiting the Idle Time of a Node’s Radio
  - **Sleep Scheduling for a Single Sensor Node**
  - Clock Calibration for an Ultra-Low Power Sensor
- Reducing the Network’s Workload
  - Soil Moisture Smart Sensor Web
- Exploiting the Spatial and Temporal Variation of the Wireless Channel
  - Transmission Scheduling with Strict Underflow Constraints

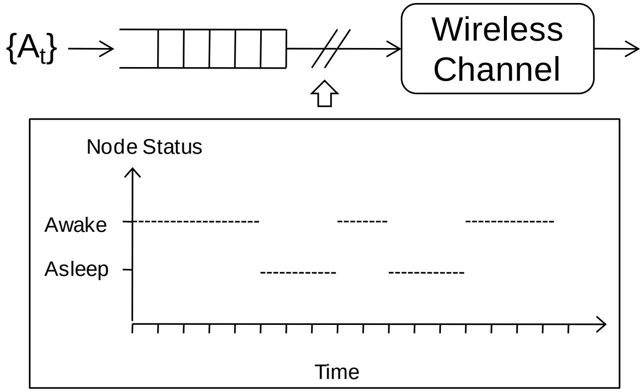
Sleep Scheduling System Model

Single Node

- Consider a single node in a wireless sensor network
- Focus on tradeoff between delay and energy consumption
- Packets (sensed information) arrive at the node, are stored in a buffer, and must be transmitted across a wireless channel

Key Modeling Assumptions

- Bernoulli arrival process with success probability *p*
- Node sleeps for *N* time slots at a time
  - While asleep, the node is unable to transmit packets, but packets continue to arrive at the node
  - In place of additional costs or setup time for switching modes
- Packets arriving in one slot cannot be transmitted until the following slot
- Only one packet transmission per slot



Problem Formulation as a Completely Observed Markov Decision Process

Information State

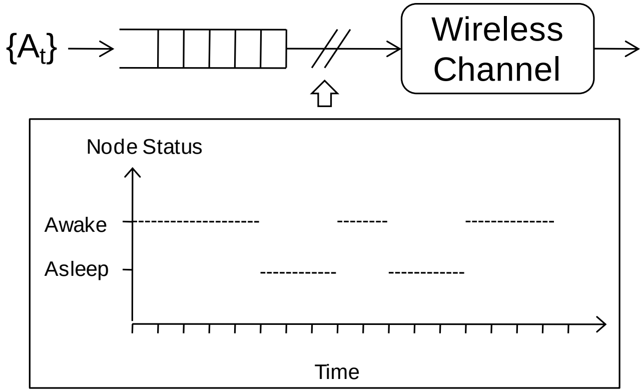
- X_t : current queue length
- S_t : number of slots remaining until node wakes up (0 → node is awake)

Action Space

- Two control actions available when node is awake:
  - $U_t = 1$ ("Stay awake")
  - $U_t = 0$ ("Sleep")

System Dynamics

- $X_{t+1} = X_t + A_t - 1_{\{X_t > 0,\, S_t = 0,\, U_t = 1\}}$
- $S_{t+1} = \begin{cases} S_t - 1, & \text{if } \; S_t > 0 \\ N - 1, & \text{if } \; S_t = 0,\; U_t = 0 \\ 0, & \text{if } \; S_t = 0,\; U_t = 1 \end{cases}$



Problem Formulation as a Completely Observed Markov Decision Process

Two Control Objectives

- Conserve energy through duty-cycling
- Minimize packet queuing delay

Cost Structure

- Constant, positive cost *d* incurred at each time slot the node is awake
- Constant, positive cost *c* incurred by each backlogged packet, at each time slot

Infinite Horizon Average Expected Cost Optimization

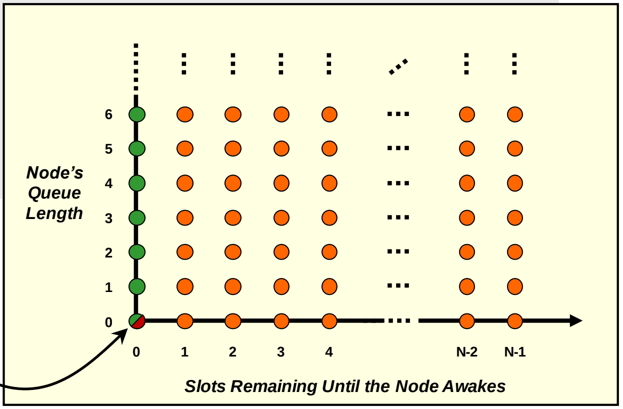
Optimization Criterion

- $$J^{\pi} := \lim_{T\to\infty} \frac{1}{T}\cdot E^{\pi}\left\{\sum_{t=0}^{T-1} d\cdot U_t + \sum_{t=1}^{T} c\cdot X_t \,\middle|\, \mathcal{F}_0\right\}$$

Stationary Optimal Policy

- When the node is awake and the queue is non-empty, the optimal action is to stay awake and transmit a packet
- When the node is awake and the queue is empty, the optimal action is given by the threshold decision rule:

$$\left(\frac{p}{1-p}\right)\cdot\left(\frac{N-1}{2}\right) \underset{\textit{Sleep } (U_t^*=0)}{\overset{\textit{Stay Awake } (U_t^*=1)}{\gtrless}} \frac{d}{c}$$



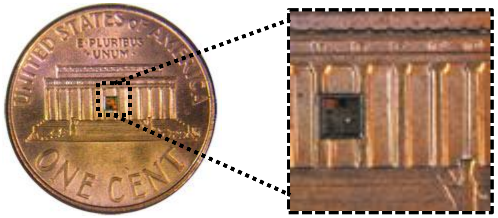
For more on the finite horizon problem:
D. Shuman and M. Liu, Asilomar, 2006

Outline

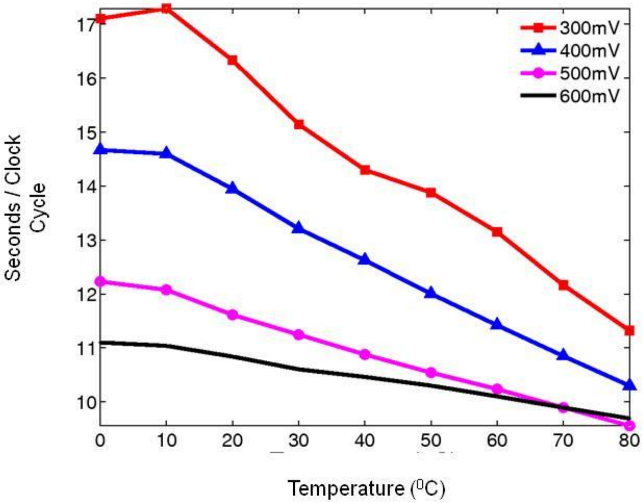
- Limiting the Idle Time of a Node’s Radio
  - Sleep Scheduling for a Single Sensor Node
  - **Clock Calibration for an Ultra-Low Power Sensor**
- Reducing the Network’s Workload
  - Soil Moisture Smart Sensor Web
- Exploiting the Spatial and Temporal Variation of the Wireless Channel
  - Transmission Scheduling with Strict Underflow Constraints

Ultra-Low Power Sensor Platform Built Around the Phoenix Processor

- Ultra-low power microchip developed at the University of Michigan by Professors David Blaauw and Dennis Sylvester and their students
- Originally designed for medical implants
  - e.g., to monitor intraocular pressure in glaucoma patients
- To manage energy consumption, platform operates in three different modes
  - Sleep mode (on the order of 1-10pW)
  - Processor mode (on the order of 1µW)
  - Radio mode (on the order of 1mW)
- Typical operation is to stay in sleep mode for extended periods of time (10-60 minutes), wake up very briefly (less than a second), and go back to sleep
  - Ultra-low power clock’s task is to time the sleep periods
- Speed of the ultra-low power clock is dependent on the ambient temperature
  - Relationship can be measured fairly reliably in off-line lab setting



The processor is one square millimeter, the same size as its thin-film battery



Relationship between ambient temperature, supply voltage, and clock period

Data courtesy of Y. Lin, D. Sylvester, and D. Blaauw, Custom Integrated Circuits Conference, Sept. 2007

Clock Calibration Using Temperature Measurements

Problem

- Temperature variations lead to inaccurate clock
- May lead to wasted energy consumption as a result of two unsynchronized devices trying to communicate

Potential Solution

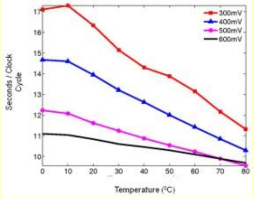
- During the long sleep phases, wake the processor up occasionally to take temperature measurements
- Use these temperature measurements to recalibrate the local clock, and more accurately estimate elapsed real time

Research Question:
How should we dynamically schedule these measurements so as to minimize the clock error?

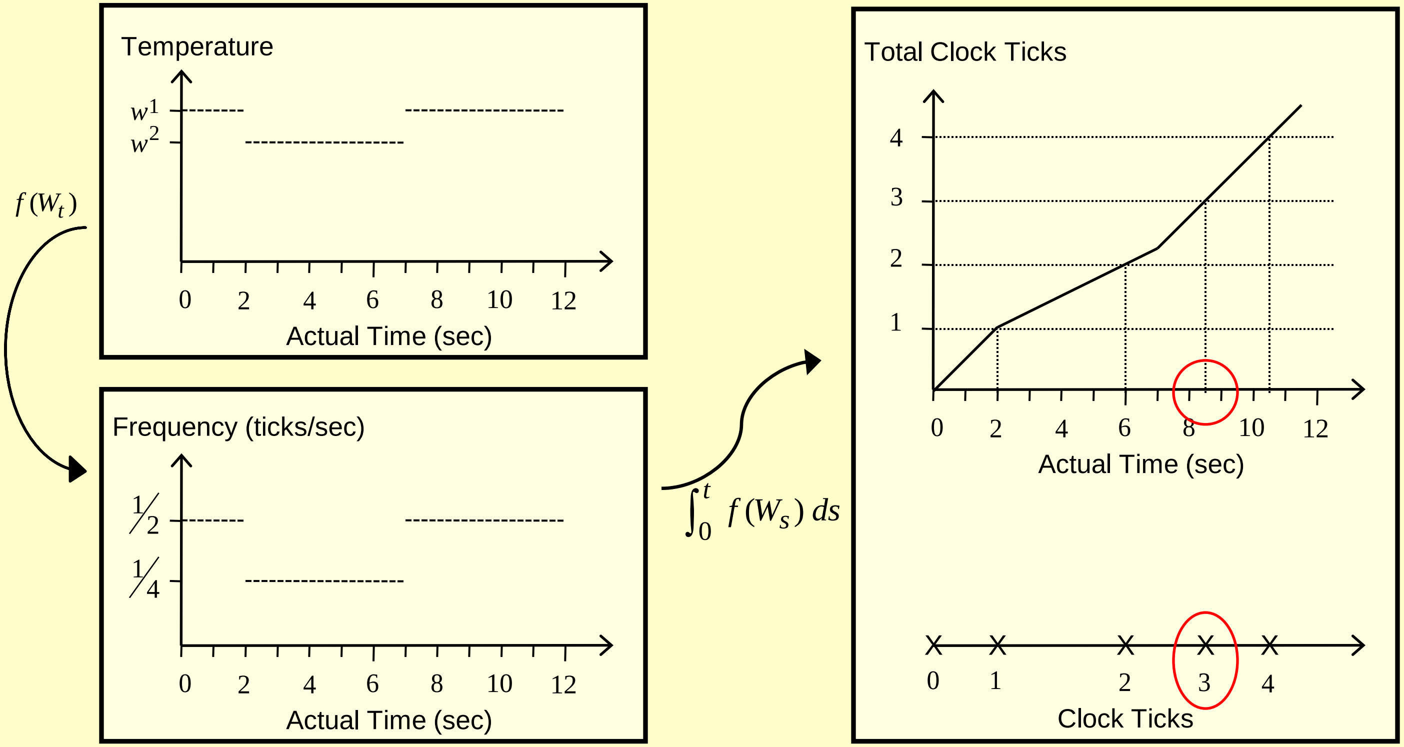
Timing of Clock Ticks

- Unusual feature: time is not a given, but rather the quantity we are trying to estimate
  - Temperature evolution affects speed of the low-power clock
  - Speed of the low-power clock affects timing of decision epochs
  - Not immediately clear in what time scale to define the problem

Toy Example

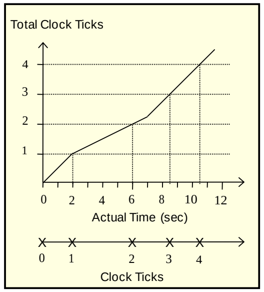


- w^1 → 1 clock tick every 2 seconds
- w^2 → 1 clock tick every 4 seconds



Continuous Time Problem Formulation

- Model as a Partially Observed Semi-Markov Decision Process (POSMDP)
  - – Continuous underlying time scale
  - – Model temperature process, $\{W_t\}_{t\geq 0}$, as a finite state continuous time Markov process
  - – Decision epochs occur at local clock ticks (random inter-decision times)
    - k^{th} decision epoch occurs at actual time t such that $C_t := \int_0^t f(W_s)\,ds = k$



  - – Allow M measurements while trying to time T seconds
  - – Partially-observed state at the k^{th} decision epoch: (X_k, W_k, N_k)
  - – Timing at each decision epoch:
    - (1) Controller observes temperature perfectly if a measurement is scheduled for that decision epoch
    - (2) Controller decides whether to declare that T seconds have elapsed
    - (3) If it does not declare the end, controller decides whether to schedule a measurement for the next epoch (if any measurements remain)

- POSMDP can be transformed to an equivalent finite state, finite action MDP

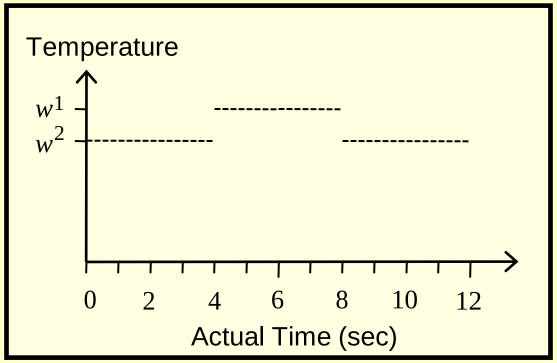
- This approach is conceptually straightforward, but difficult from a computational standpoint

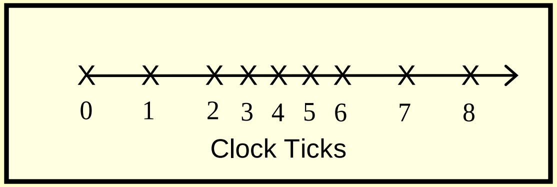
Discrete Time Problem Formulation

- Also possible to model underlying time scale as discrete if the temperature process and possible clock frequencies satisfy some extra conditions

Toy Example 2

- Temperature changes can only happen at 2,4,6,8,… seconds
- Temperature process at these times is a discrete time Markov process
- $w^1 \rightarrow$ 1 clock tick every second
- $w^2 \rightarrow$ 1 clock tick every 2 seconds





Toy Example 2 – Optimal Scheduling

- Initial temperature is w^2
- Transition matrix: $P = \begin{bmatrix} .9 & .1 \\ .3 & .7 \end{bmatrix}$
- Goal is to measure T=12 seconds
- L-1 distortion criteria
- Problem:

$$\min_{\pi} E^{\pi}\left\{ \left| T - \hat{T} \right| \,\middle|\, W_0 = w^2 \right\}$$

Open-loop (no measurements)

- Optimal to declare 12 seconds have elapsed after 9 clock ticks
- Resulting expected distortion is **1.85**

One measurement allowed

- Optimal to take the measurement at the 4th clock tick
- If measurement is w^1, wait 6 more ticks before declaring (at 10th tick)
- If measurement is w^2, wait 2 more ticks before declaring (at 6th tick)
- Resulting expected distortion is **1.00**

Two measurements allowed

- Optimal to take the 1st measurement at the 2nd clock tick
- If 1st measurement is w^1, wait 4 more ticks before 2nd measurement
  - If 2nd measurement is w^1, wait 4 more ticks before declaring (at 10th tick)
  - If 2nd measurement is w^2, wait 2 more ticks before declaring (at 8th tick)
- If 1st measurement is w^2, wait 2 more ticks before 2nd measurement
  - If 2nd measurement is w^1, wait 4 more ticks before declaring (at 8th tick)
  - If 2nd measurement is w^2, wait 2 more ticks before declaring (at 6th tick)
- Resulting expected distortion is **0.57**

Summary and Future Work

- Sleep scheduling for a single sensor node
  - Formulated model to examine tradeoff between packet delay and energy consumption
  - Completely characterized optimal policy in infinite horizon average expected cost problem
  - Proved some structural results for finite horizon expected cost problem
  - Future work includes completing the characterization of the optimal policy for the finite horizon problem

- Clock calibration for an ultra-low power sensor
  - Presented a potential solution to make the ultra-low power timer more accurate, namely taking temperature measurements
  - Formulated measurement scheduling problems in both continuous and discrete time
  - Developed numerical solution for small dimensional discrete time problems
  - Future work includes:
    - Improving numerical solution techniques, and possibly considering approximations
    - Developing more accurate application-specific temperature models
    - Implementing measurement policies in hardware to test actual energy savings