Network Architectures
Sensor Networks
Lecture 1:
Sensor Networks and MAC Layer

Dr. Cecilia Mascolo

UNIVERSITY OF CAMBRIDGE

About Me

- Reader in Mobile Systems
  - NetOS Research Group
- Lecturer on Social and Technological Networks Analysis

- Research on Mobile, Social and Sensor Systems
- More specifically,
  - Human Mobility and Social Network modelling
  - Opportunistic Mobile Networks
  - Mobile Sensor Systems and Networks

twitter

foursquare

UNIVERSITY OF CAMBRIDGE

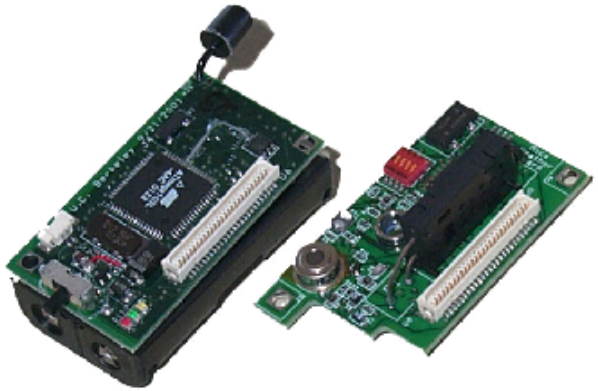
In This Lecture

- Discussion on sensor network systems through an example
  - Issues related to energy trade offs, reprogramming and routing
- Principles of sensor operating systems
- Discussion of possible MAC layer solutions

Sensors and Sensor Networks

A sensor is a device which allows “sensing” of the environment. It is usually small and resource constrained.
A **sensor network** is composed of a large number of sensor nodes, which are deployed either inside the phenomenon or very close to it.

- Sometimes Random deployment
- Cooperative capabilities



UNIVERSITY OF CAMBRIDGE

Sensor Systems vs Standard or Mobile Systems

- Sensor nodes are limited in power, computational capacities and memory.
- Sensor nodes are prone to failures (especially because they are often deployed in challenging conditions)
- The topology of a sensor network might not change frequently:
  - Many deployments involve sensors with fixed location
  - Some deployments may have mobile sensors

Example applications

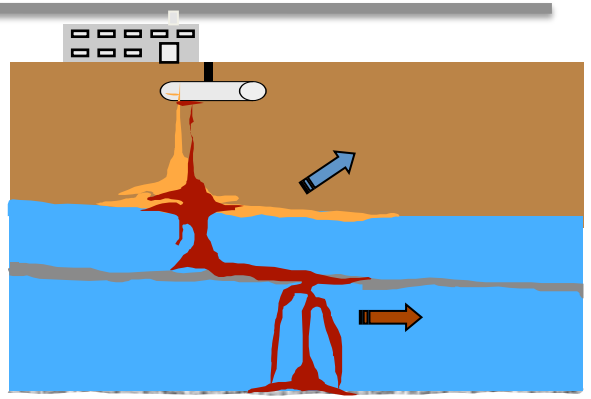
Seismic Structure response

Marine Microorganisms

UNIVERSITY OF CAMBRIDGE

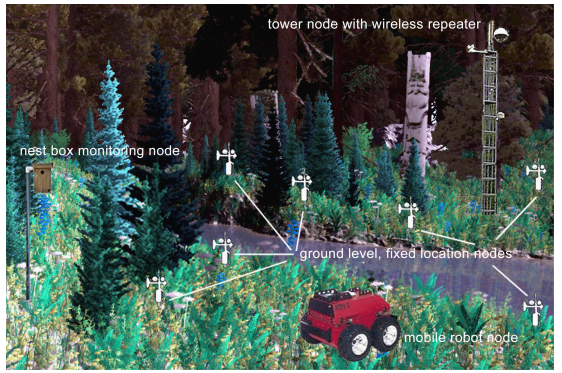
- Micro-sensors, on-board processing, wireless interfaces feasible at very small scale--can monitor phenomena “up close”
- Enables spatially and temporally dense environmental monitoring

Embedded Networked Sensing will reveal previously unobservable phenomena

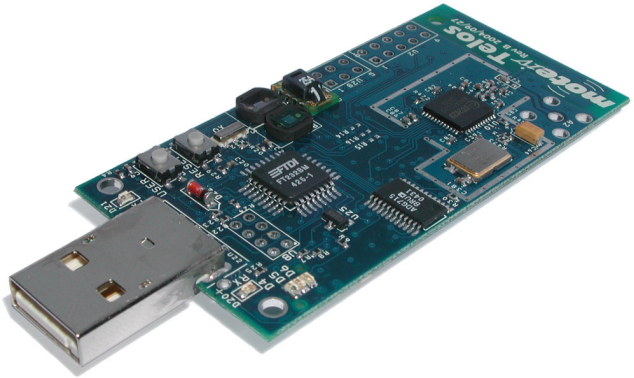


Contaminant Transport

Ecosystems, Biocomplexity



Experimental Platform



- Standards Based
  - USB
  - Radio:
    - IEEE 802.15.4 (CC2420 radio)
    - Zigbee: Ultralow power
- 8-bit microprocessor, 4MHz CPU
  - ATMEGA 128, ATMEL 8535, or Motorola HCS08
- ~4Kb RAM
  - holds run-time state (values of the variables) of the program
- ~128Kb programmable Flash memory
  - holds the application program
  - Downloaded via a programmer-board or wirelessly
- Additional Flash memory storage space

UNIVERSITY OF CAMBRIDGE

Wireless Sensor Networks for Habitat Monitoring

A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson

First ACM International Workshop on Wireless Sensor Networks and Applications, WSNA 2002, Atlanta, Georgia, USA, September 28, 2002

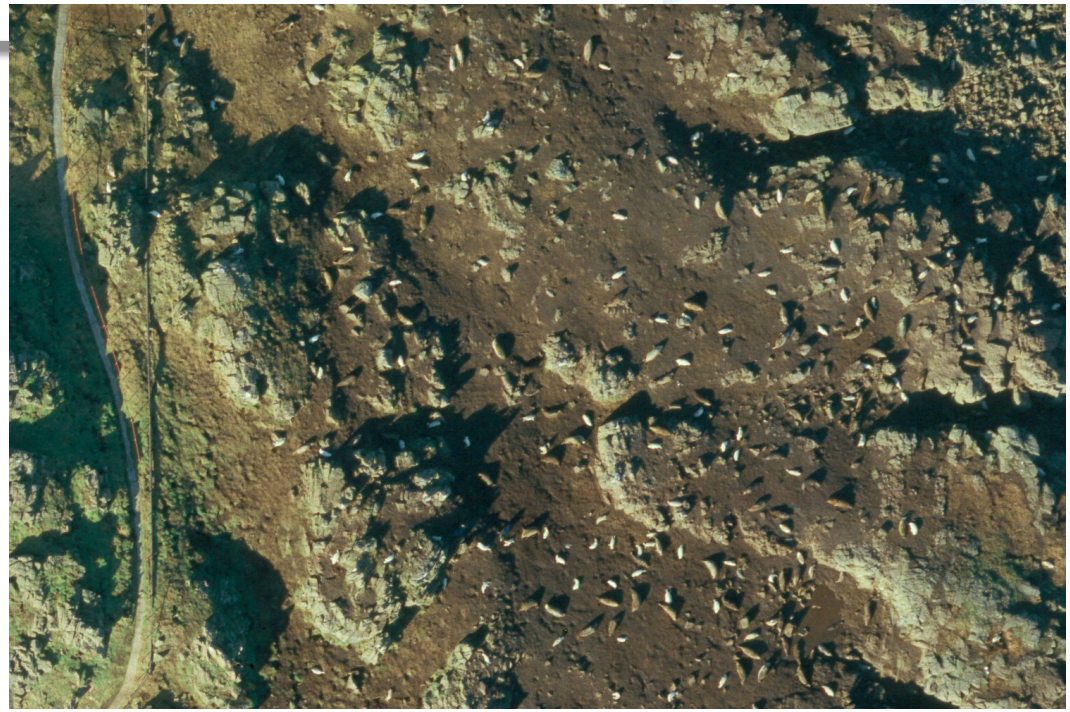
UNIVERSITY OF CAMBRIDGE

Motivation

- What environmental factors make for a good nest?
- How much can they vary?
- What are the occupancy patterns during incubation?
- What environmental changes occur in the burrows and their surroundings during the breeding season?

UNIVERSITY OF CAMBRIDGE

Current Observation Techniques



- Prolonged Observation
- Adding colours to food
- Counting on aerial pictures

- More advanced:
- Cameras [usually watching what happens on the video for hours]
- VHF tracking

UNIVERSITY OF CAMBRIDGE

VHF tracking

UNIVERSITY OF CAMBRIDGE

VHF = Very High Frequency radio
Data = bearing

Motivation

Problems

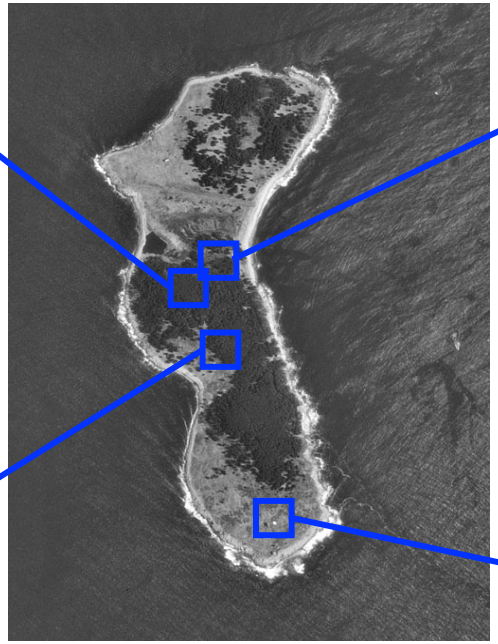
- Seabird colonies are very sensitive to disturbances
  - The impact of human presence can distort results by changing behavioral patterns and destroy sensitive populations
  - Repeated disturbance will lead to abandonment of the colony

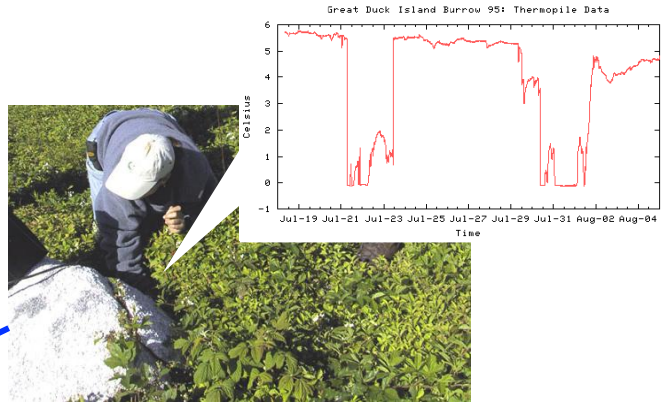
Solution

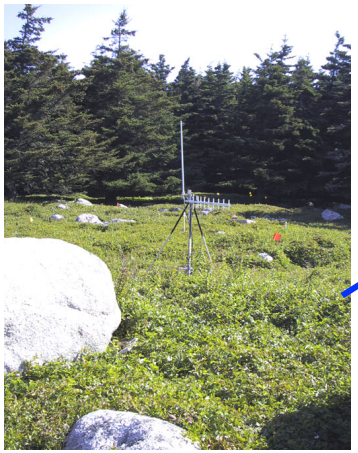
- Deployment of a sensor network

UNIVERSITY OF CAMBRIDGE

Great Duck Island Project

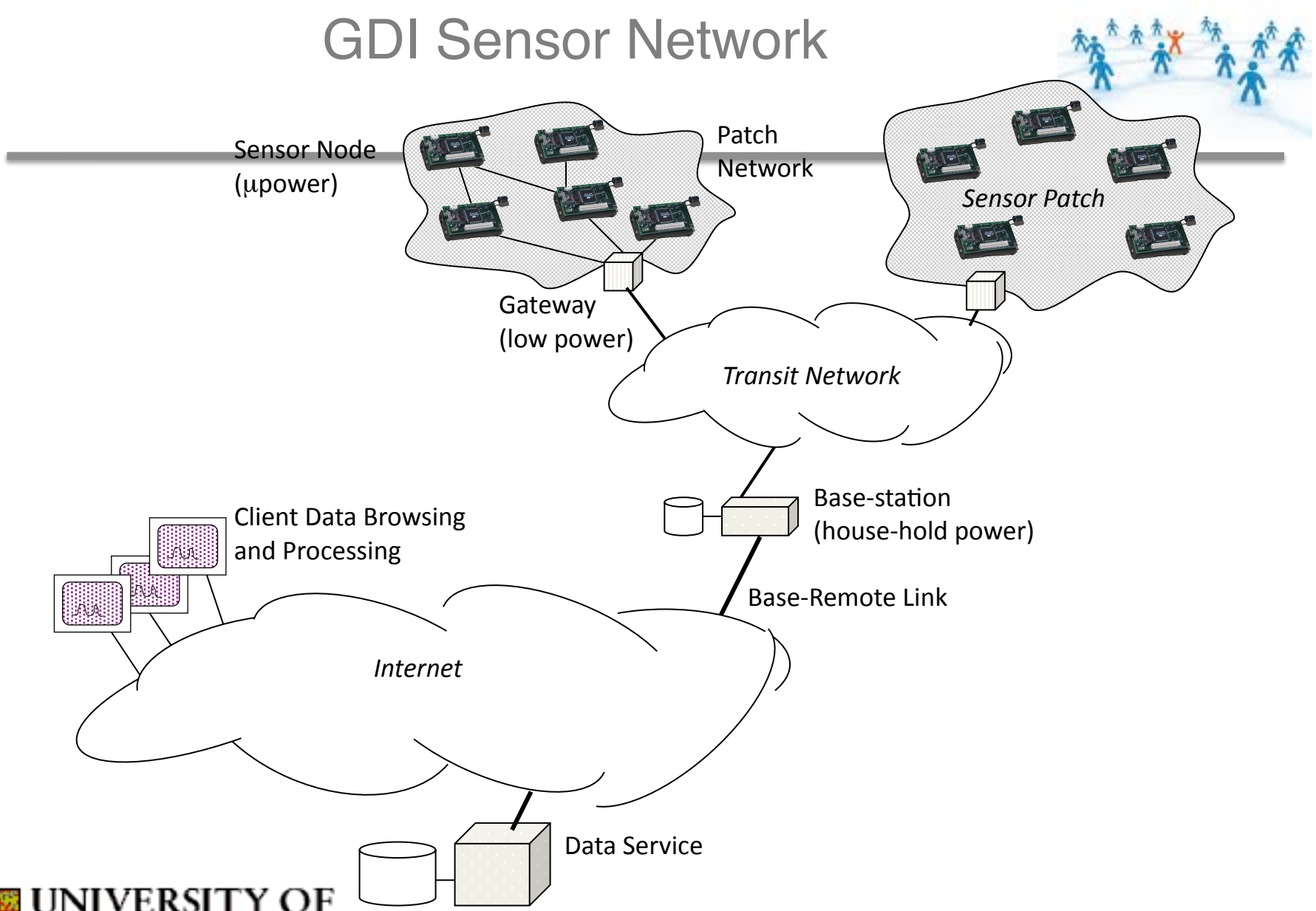






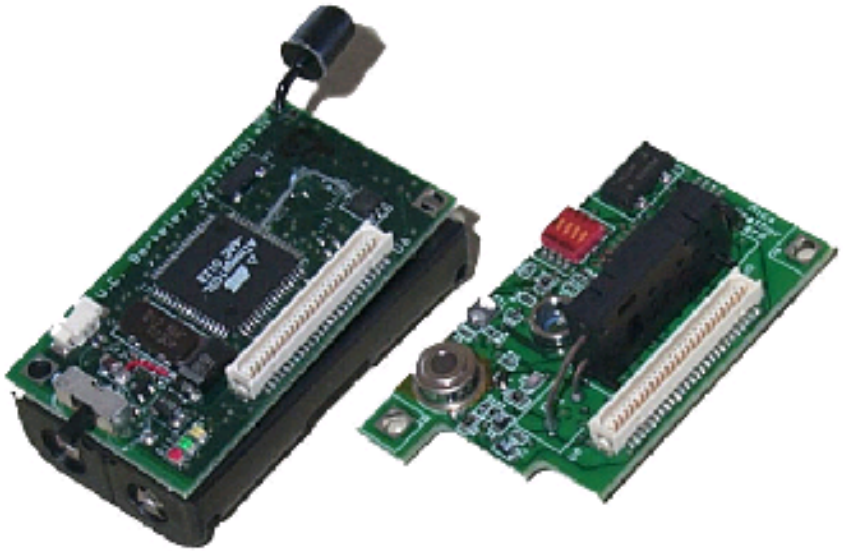
UNIVERSITY OF CAMBRIDGE

GDI Sensor Network



UNIVERSITY OF CAMBRIDGE

Mica Sensor Node



Left: Mica II sensor node
2.0x1.5x0.5 cu. In.

Right: weather board with temperature, thermopile (passive IR), humidity, light, accelerometer sensors, connected to Mica II node

- Single channel, 916 Mhz radio for bi-directional radio @40kbps
- 4MHz micro-controller
- 512KB flash +4KB RAM
- 2 AA batteries (~2.5Ah)
- Sensors are pre-calibrated (±1-3%) and interchangeable

UNIVERSITY OF CAMBRIDGE

Power Management

Sensor Node Power

- Limited Resource (2 AA batteries)
- Estimated supply of 2200 mAh at 3 volts
- Each node has 8.128 mAh per day (9 months)
- Processor draws apx 5 mA => can run at most 1.4 hours/day
- Nodes near the gateway will do more forwarding

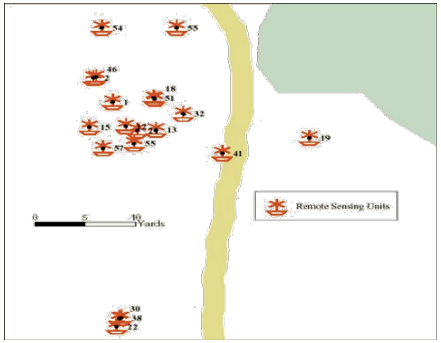
Operation	**nAh**
Transmitting a packet	20.000
Receiving a packet	8.000
Radio listening for 1 millisecond	1.250
Operating sensor for 1 sample (analog)	1.080
Operating sensor for 1 sample (digital)	0.347
Reading a sample from the ADC	0.011
Flash Read Data	1.111
Flash Write/Erase Data	83.333

Data Sampling

- Compression on the data is used
  - Distributed aggregation can be used to only send a portion of the data (min/max temp value)
  - Coding techniques (Huffman coding)
- Energy needed for sampling is allocated and the rest of the energy is considered for communication and network management

Routing

- Sensors transmit multihop to gateway which is an always on node
- Determine initial routing tree and assign a “level” to each sensor
- Nodes at same level communicate at same time and “wake up time” moves as a wave through the tree.
- [alternatively wake up sub-trees instead of levels]
- Schedule is fixed and not reconfigurable. Power is wasted when certain paths are not very productive.
- For more reconfigurable/efficient schedule use low power MAC protocol [see later slides]



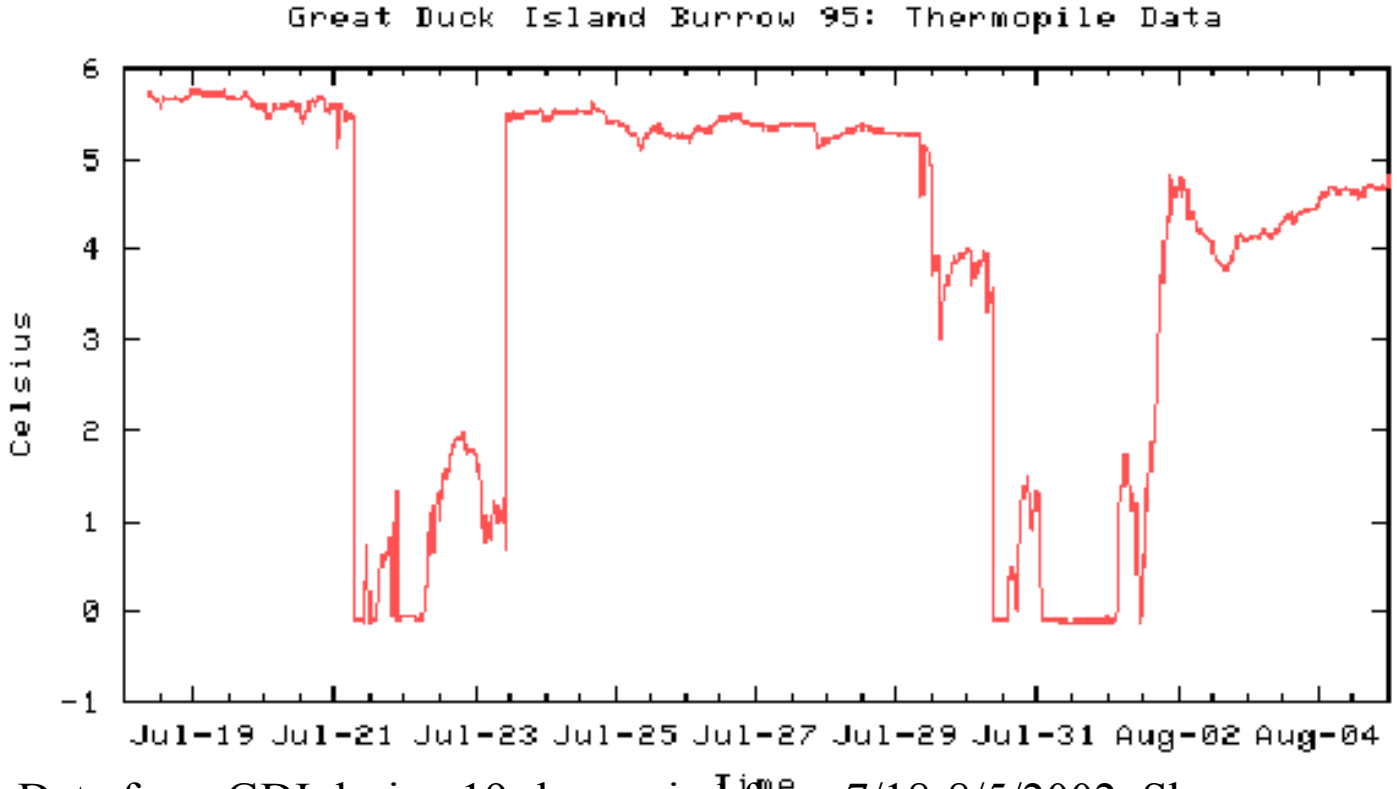
UNIVERSITY OF CAMBRIDGE

Network Re-tasking

- Initially collect absolute temperature readings

- After initial interpretation, could be realized that information of interest is contained in significant temperature changes
- 3 ways of retasking
  - Parameter changes through network management/monitoring
  - Virtual machine retasking [only small functions]
  - Full retasking of native code [more expensive but more efficient than VM code]

Sensed Data: Occupancy of Burrow



Data from GDI during 19-day period from 7/18-8/5/2002. Show difference between ambient temperature and the object in the thermopile's field of view. It indicates that the petrel left on 7/21, return on 7/23, and between 7/30 and 8/1

UNIVERSITY OF CAMBRIDGE

Health and Status Monitoring

- Monitor the mote's health and the health of neighbouring motes
- Sensor duty cycle can be dynamically adjusted to alter lifetime
- Periodically include battery voltage level with sensor readings (0~3.3volts)
- Can be used to infer the validity of the mote's sensor readings

Conclusions

Seminal paper (2002)

Applied wireless sensor networks to real-world habitat monitoring

- Two small scale sensor networks deployed at Great Duck Island and James Reserve (one patch each)
- Not extensive validation and evaluation of results

- Very high level paper
- Gives an idea of problems faced by these deployments
- Lots of discussion of what could be applied: not clear what has been…

What Operating System runs on a sensor?

- Operating system useful to simplify programming tasks and to allow more control over operations of the system
- But what can we do with such a constrained device?
- Given the kind of applications needed it is important to support concurrency…[frequent and parallel collection from different sensors]
- If you want to read about a more recent deployment paper:

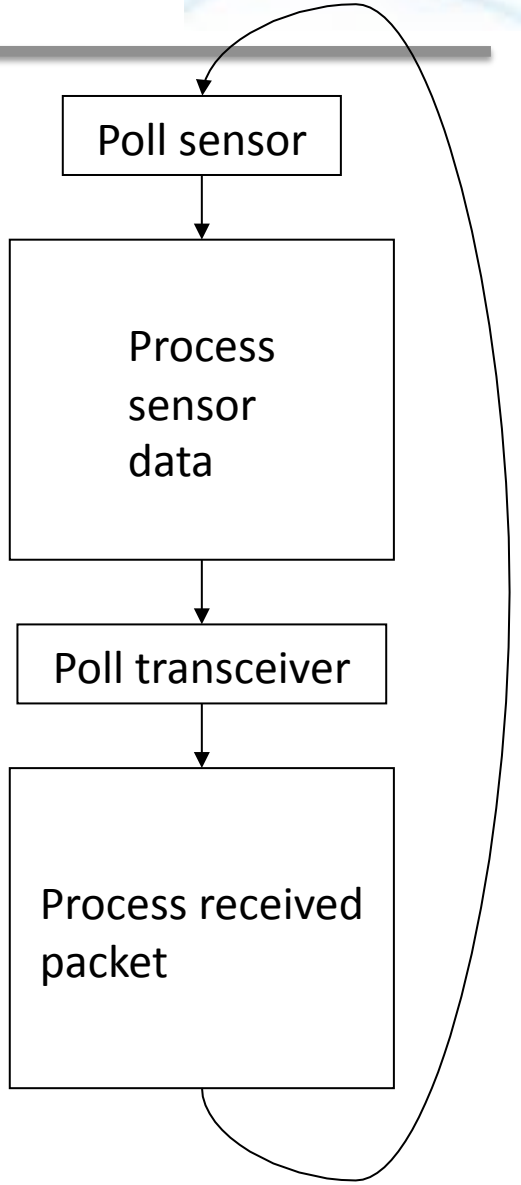
  Evolution and Sustainability of a Wildlife Monitoring Sensor Network. Vladimir Dyo, Stephen A. Ellwood, David W. Macdonald, Andrew Markham, Cecilia Mascolo, Bence Pasztor, Salvatore Scellato, Niki Trigoni, Ricklef Wohlers, Kharsim Yousef. In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (Sensys 2010). Zurich, Switzerland. November 2010.

UNIVERSITY OF CAMBRIDGE

Main issue: How to support concurrency

- Simplest option: No concurrency, sequential processing of tasks
  - Not satisfactory: Risk of missing data (e.g., from transceiver) when processing data, etc.

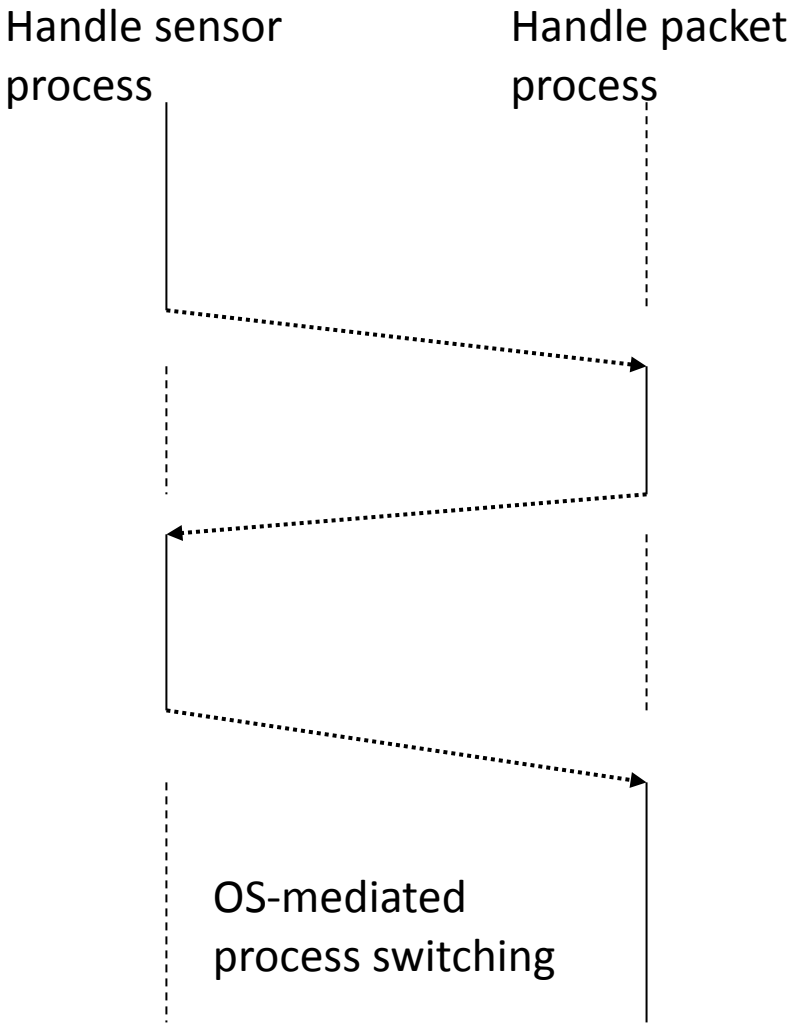
  ! Interrupts/asynchronous operation has to be supported
- Why concurrency is needed
  - Sensor node's CPU has to service the radio modem, the actual sensors, perform computation for application, execute communication protocol software, etc.



UNIVERSITY OF CAMBRIDGE

Traditional concurrency: Processes

- Traditional OS: processes/threads
  - Based on interrupts, context switching
  - But: memory overhead, execution overhead
- concurrency mismatch
  - One process per protocol entails too many context switches
  - Many tasks in WSN small with respect to context switching overhead

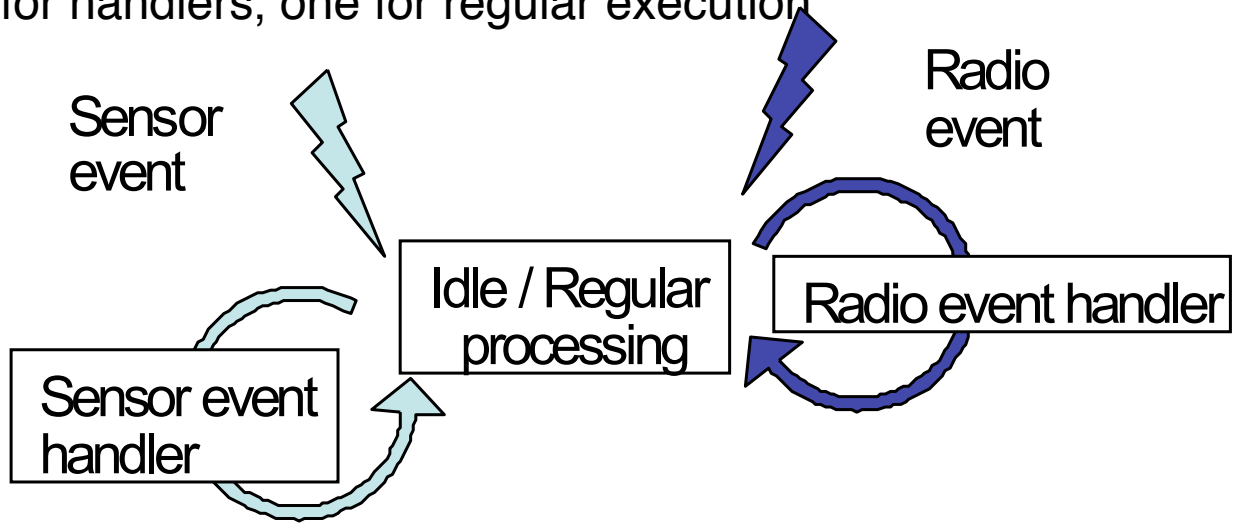


UNIVERSITY OF CAMBRIDGE

Event-based concurrency

- Alternative: Switch to ***event-based programming model***
  - Perform regular processing or be idle
  - React to events when they happen immediately
  - Basically: interrupt handler
- Problem: must not remain in interrupt handler too long
  - Danger of losing events
  - Only save data, post information that event has happened, then return

  → ***Run-to-completion*** principle
  - Two contexts: one for handlers, one for regular execution



UNIVERSITY OF CAMBRIDGE

TinyOS: Tasks and Command/ Event Handlers

- TinyOS: an OS for sensor networks
- *Event handlers* must run to completion
  - Must not wait an indeterminate amount of time
  - Only a ***request*** to perform some action
- *Tasks*, on the other hand, can perform arbitrary, long computation
  - Also have to be run to completion
  - But can be interrupted by handlers

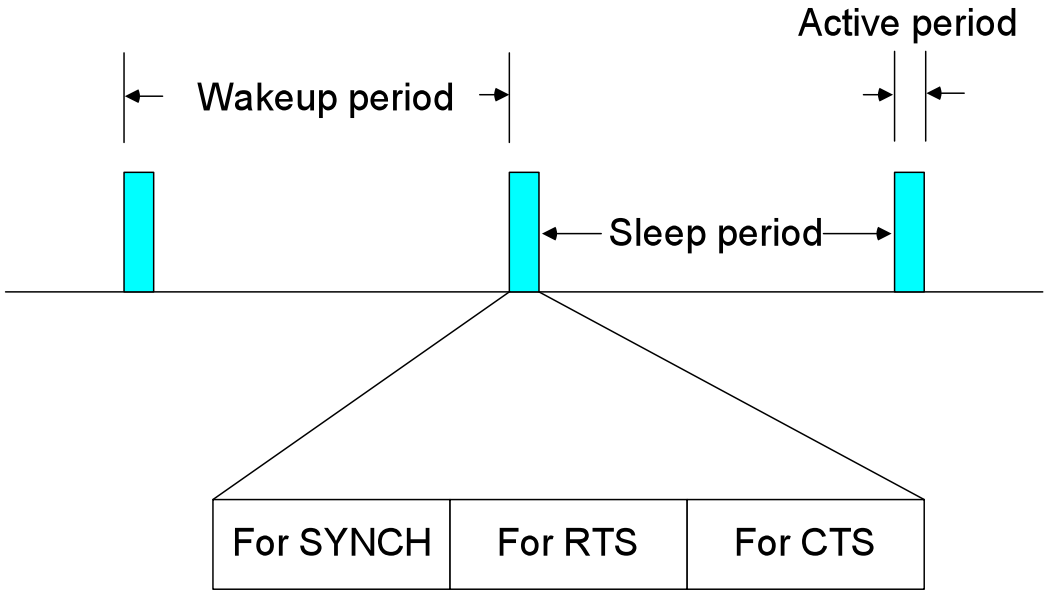
  ! No need for stack management, tasks are atomic with respect to each other

Energy Management

- We have seen that processing is not the greatest source of energy consumption
- Communication is the main source of energy consumption
- The paper used a static assignment of sleep and awake times
- More advanced protocols have been proposed which allow dynamic assignment but also allow to adapt to current traffic load

Sensor-MAC (S-MAC)

- idle listening is particularly unsuitable if average data rate is low
  - Most of the time, nothing happens
- Idea: Switch nodes off, ensure that neighboring nodes turn on simultaneously to allow packet exchange (rendez-vous)
  - Only in these ***active periods***, packet exchanges happen
  - Need to also exchange wakeup schedule between neighbors
  - When awake, essentially perform RTS/CTS
- Use SYNCH, RTS, CTS phases



S-MAC

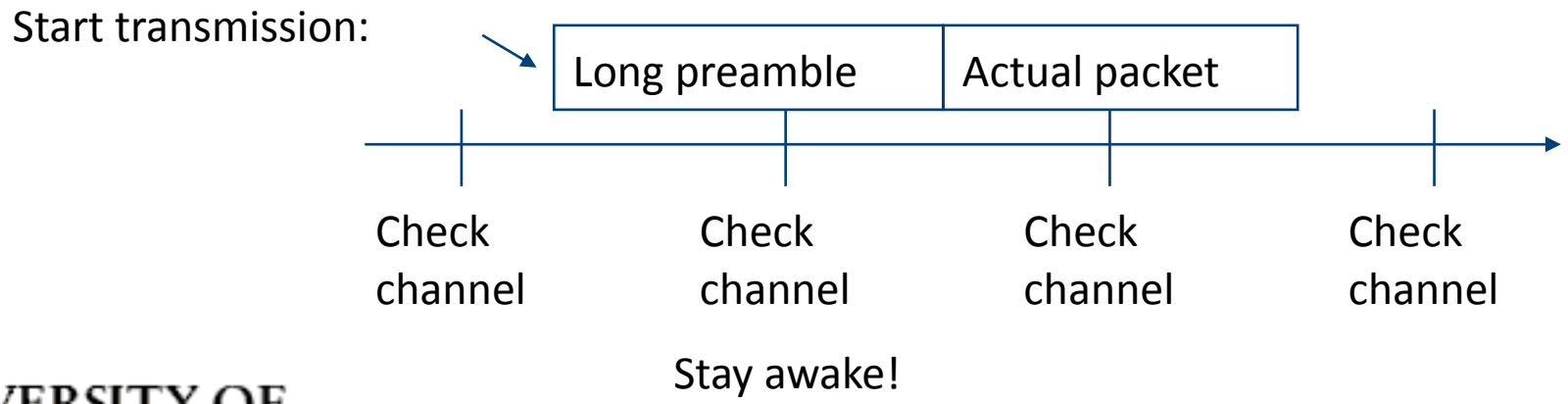
- SYNC phase divided into time slots using CSMA and backoff to send schedule to neighbours
- Y chooses a slot and if no signal was received before it will start to transmit its schedule to X otherwise wait for next wake up of X
- RTS phase: X listens for RTS packets (CSMA contention)
- CTS phase: X sends one and extends its wake up time

S-MAC synchronized islands

- Nodes try to pick up schedule synchronization from neighboring nodes
- If no neighbor found, nodes pick some schedule to start with
- If additional nodes join, some node might learn about two different schedules from different nodes
  - "Synchronized islands"
- To bridge this gap, it has to follow both schemes and use more energy

Preamble Sampling

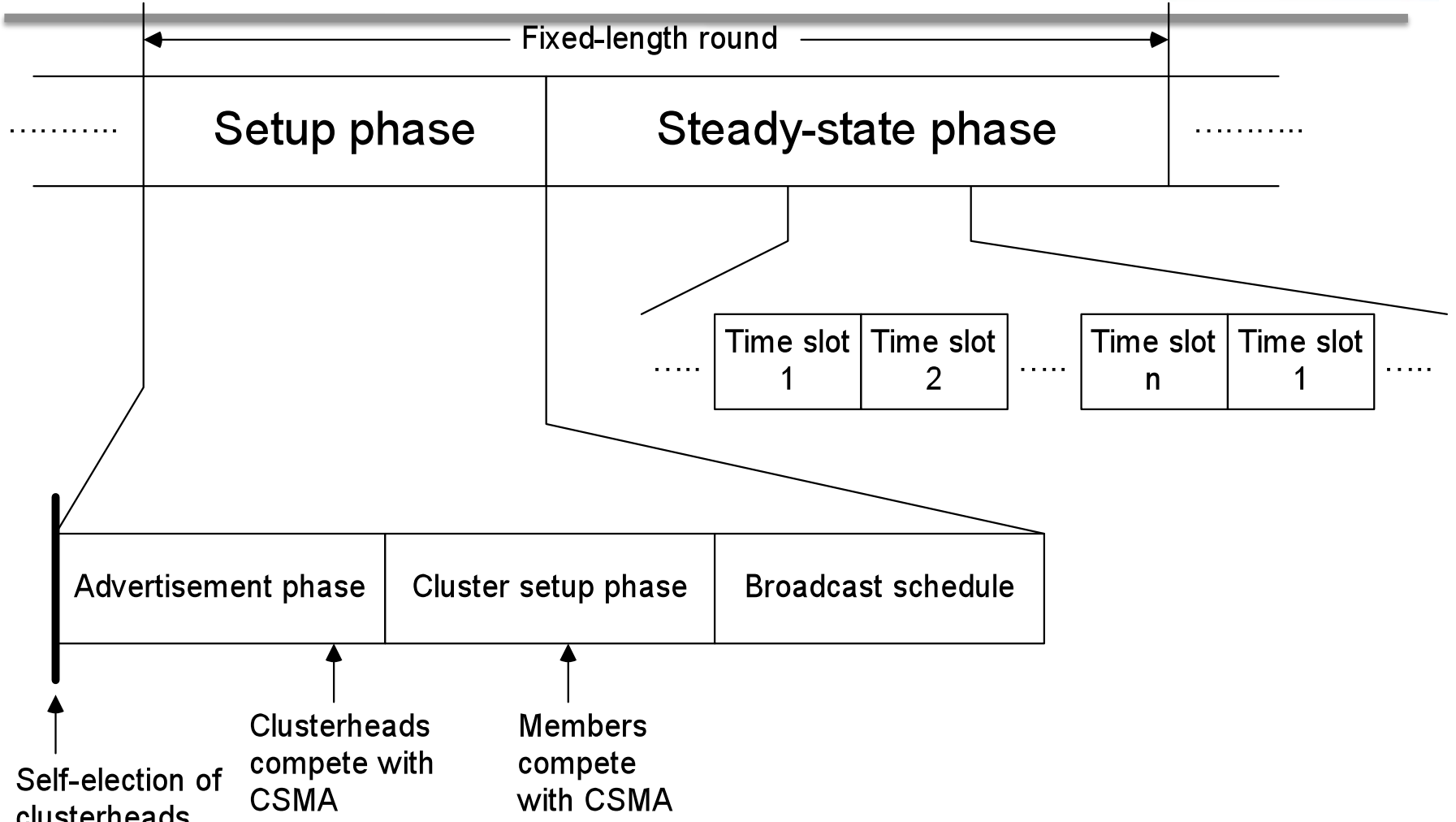
- So far: Periodic sleeping supported by some means to synchronize wake up of nodes to ensure rendez-vous between sender and receiver
- Alternative option: Don't try to explicitly synchronize nodes
  - Have receiver sleep and only periodically sample the channel
- Use ***long preambles*** to ensure that receiver stays awake to catch actual packet. Example: WiseMAC:



Low-Energy Adaptive Clustering Hierarchy (LEACH)

- Given: dense network of nodes, reporting to a central sink, each node can reach sink directly
- Idea: Group nodes into "***clusters***", controlled by ***clusterhead***
  - Setup phase; details: later
  - About 5% of nodes become clusterhead (depends on scenario)
  - Role of clusterhead is rotated to share the burden
  - Clusterheads advertise themselves, ordinary nodes join CH with strongest signal
  - Clusterheads organize
    - CDMA code for all member transmissions
    - TDMA schedule to be used within a cluster
- In steady state operation
  - CHs collect & aggregate data from all cluster members
  - Report aggregated data to sink using CSMA

LEACH rounds



UNIVERSITY OF CAMBRIDGE

References

- Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J. 2002. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international Workshop on Wireless Sensor Networks and Applications* (Atlanta, Georgia, USA, September 28 - 28, 2002). WSNA '02. ACM, New York, NY, 88-97.
- TinyOS tutorial: http://www.tinyos.net/tinyos-1.x/doc/tutorial/
- SMAC: Ye, W., Heidemann, J., and Estrin, D. 2004. Medium access control with coordinated adaptive sleeping for wireless sensor networks. IEEE/ACM Trans. Netw. 12, 3 (Jun. 2004), 493-506.
- WISEMAC: El-Hoiydi, A. and Decotignie, J. 2004. WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. In Proceedings of the Ninth international Symposium on Computers and Communications 2004 Volume 2 (Iscc"04) - Volume 02 (June 28 - July 01, 2004). ISCC. IEEE Computer Society, Washington, DC, 244-251.
- LEACH: Wendi Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, Energy-Efficient Communication Protocols for Wireless Microsensor Networks, Proc. Hawaaian Int'l Conf. on Systems Science, January 2000.