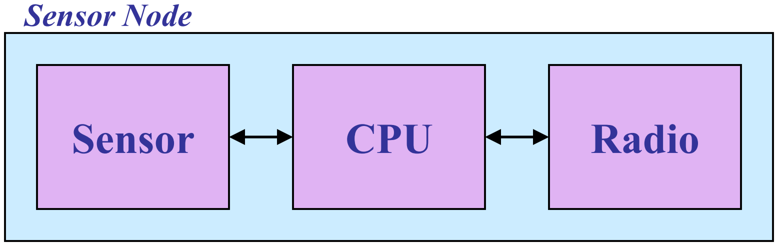
Wireless Sensor Networks

Carleton
UNIVERSITY
Canada's Capital University

WSN Definition

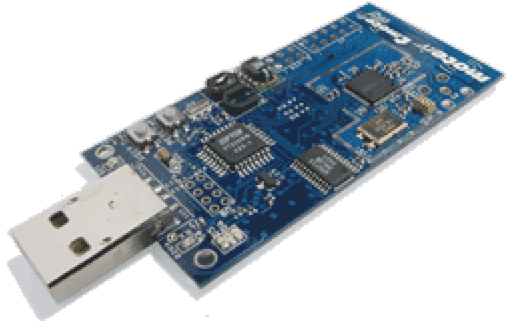
- Wikipedia entry
  - A **wireless sensor network** (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations.
- Electrical Engineering Glossary Definition for Wireless Sensor Network:
  - Wireless Sensor Network, or WSN, is a network of RF transceivers, sensors, machine controllers, microcontrollers, and user interface devices with at least two nodes communicating by means of wireless transmissions.

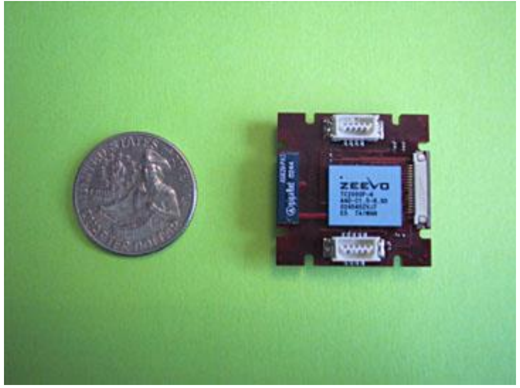


- Technology that enables WSN is relatively new (wireless data networks, microcontrollers/computers)

Wireless Sensor Nodes

- Micro-sensors
  - Sensor module (e.g., acoustic, seismic, image)
  - A digital processor for signal processing and network protocol functions
  - Radio for communication (Zigbee/IEEE 802.15.4)
  - Battery-operated, but increasingly focus on energy-scavenging
- Sensors monitor environment
  - Cameras, microphones, physiological, magnetic, pressure, biological sensors, etc.
  - Gather data for some purpose

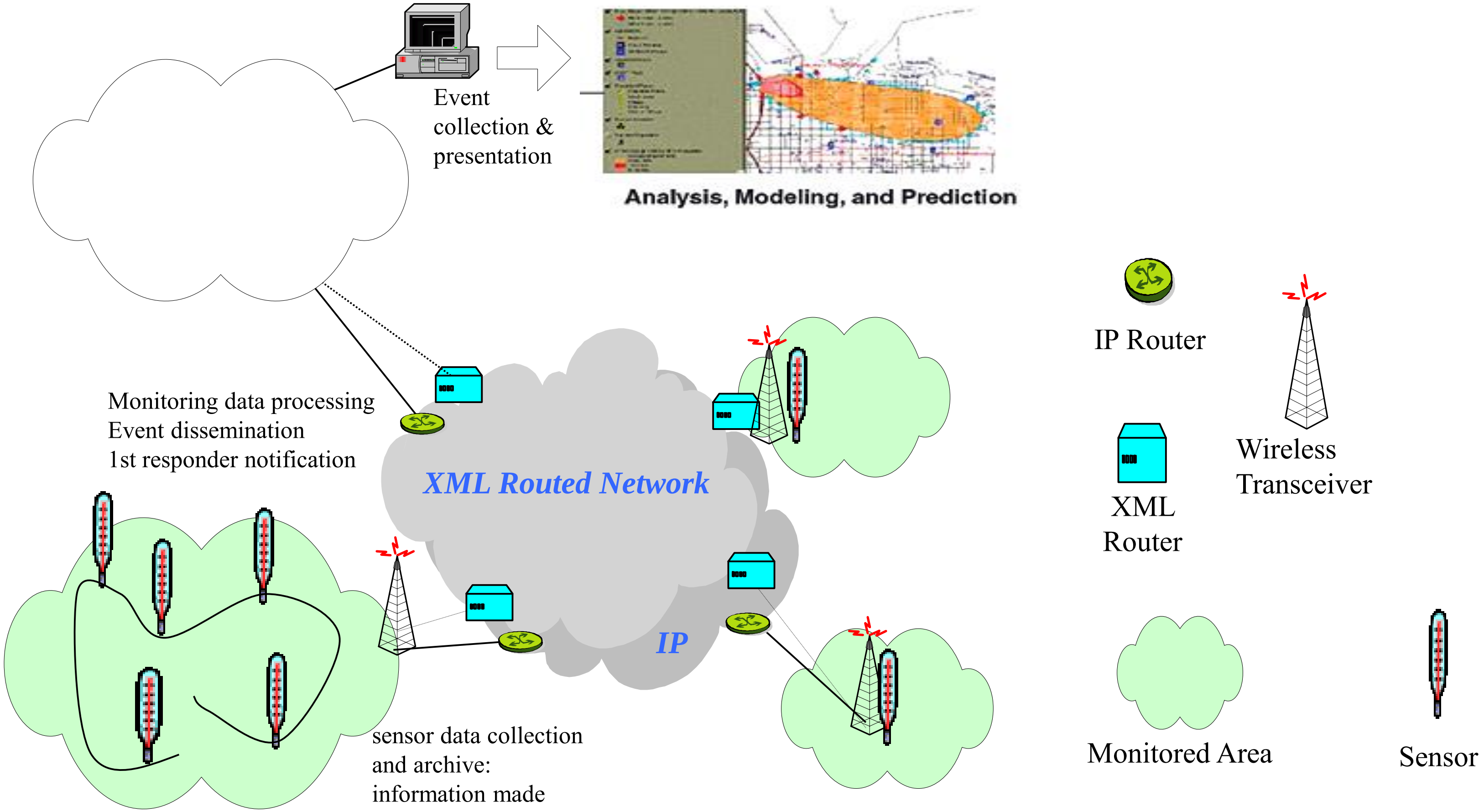




Wireless Sensor Networks (WSNs)

- Sensor data limited in range and accuracy
  - Each node can only gather data from a limited physical area of the environment
  - Data may be noisy
- Tens, hundreds, thousands of nodes scattered throughout an environment → Sensor nodes have to be cheap
- Each sensor can collect data
- Data routed via other sensors to
  - One or more *sink* or *base station* nodes
  - Other sensors
- Networking sensors enables
  - Extended range of sensing → improved quality
  - Fault tolerance due to redundancy in data from different sensors
  - Distributed processing of large amounts of data
  - Duty-cycling individual nodes
  - Scalability: quality can be traded for system lifetime
  - “Team-work”: nodes can help each perform a larger sensing task

Complete Architecture: multiple WSN, fixed Core (Example: surveying multiple airports, border crossings, etc.)



Event collection & presentation

Analysis, Modeling, and Prediction

Monitoring data processing
Event dissemination
1st responder notification

XML Routed Network

IP

sensor data collection and archive: information made available via web services

IP Router

Wireless Transceiver

XML Router

Monitored Area

Sensor

Wireless Sensor Networks: Examples

Carleton
UNIVERSITY
Canada's Capital University

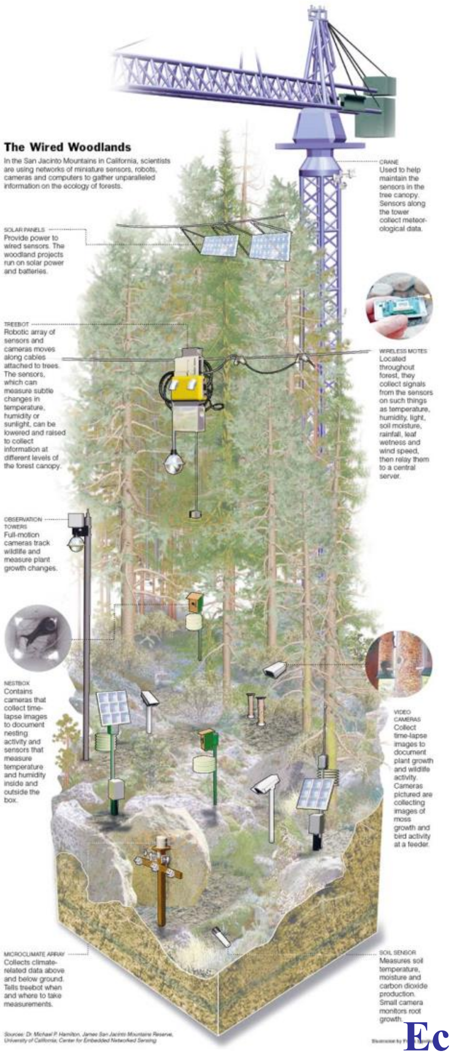
WSN Applications

- Many ways to categorize them
  - Monitoring Space
    - Habitat monitoring, precision agriculture, surveillance, treaty verification, indoor climate control, ….
  - Monitoring Things
    - Structural monitoring, ecophysiology, medical diagnostics, urbain terrain mapping, ….
  - Monitoring interaction of Things with each other and surrounding Space
    - Wildlife habitat, disaster management, pervasive computing, asset tracking, manufacturing process flow, ….
  - Wireless Sensor Networks: A Survey: by application area
    - Military Applications
    - Environmental Applications
    - Health Applications
    - Home Applications
    - Other Commercial Applications
- Key point: lots of applications

Example Application: Environmental Monitoring

- Example projects
  - ZebraNet
  - Ecology of rare plants in Hawaii
  - California Redwood Forest (first real example)
  - Great Duck Island: bird monitoring
- Collecting detailed data about some phenomenon can help advance respective science/knowledge

A New Instrument for the Sciences



The Wired Woodlands

In the San Jacinto Mountains in California, scientists are using networks of miniature sensors, robots, cameras and computers to gather unparalleled information on the ecology of forests.

SOLAR PANELS
Provide power to wired sensors. The woodland projects run on solar power and batteries.

CRANE
Used to help maintain the sensors in the tree canopy. Sensors along the tower collect meteorological data.

TREEBOT
Robotic array of sensors and cameras moves along cables attached to trees. The sensors, which can measure subtle changes in temperature, humidity or sunlight, can be lowered and raised to collect information at different levels of the forest canopy.

WIRELESS MOTES
Located throughout forest, they collect signals from the sensors on such things as temperature, humidity, light, soil moisture, rainfall, leaf wetness and wind speed, then relay them to a central server.

OBSERVATION TOWERS
Full-motion cameras track wildlife and measure plant growth changes.

NESTBOX
Contains cameras that collect time-lapse images to document nesting activity and sensors that measure temperature and humidity inside and outside the box.

VIDEO CAMERAS
Collect time-lapse images to document plant growth and wildlife activity. Cameras pictured are collecting images of moss growth and bird activity at a feeder.

MICROCLIMATE ARRAY
Collects climate-related data above and below ground. Tells treebot when and where to take measurements.

SOIL SENSOR
Measures soil temperature, moisture and carbon dioxide production. Small camera monitors root growth.

Sources: Dr. Michael P. Hamilton, James San Jacinto Mountains Reserve, University of California; Center for Embedded Networked Sensing

> "Nothing tends so much to the advancement of knowledge as the application of a **new instrument**. The native intellectual powers of men in different times are not so much the causes of the different success of their labours as the peculiar nature of the means and artificial resources in their possession."

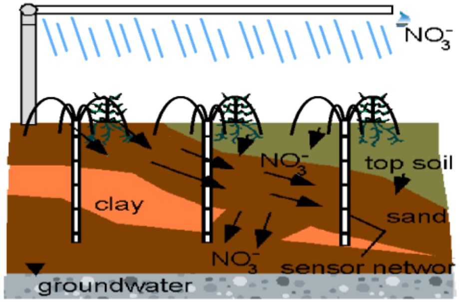
-Sir Humphry Davy

exponent of the scientific method;
discoverer of sodium and potassium.

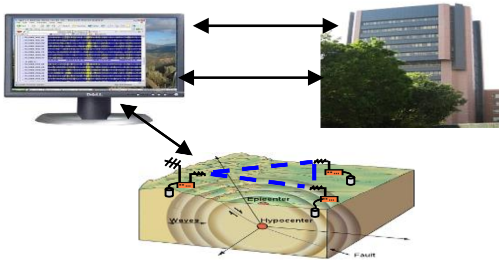
Embedded Networked Sensing helps reveal previously unobservable phenomena

NY Times, May 10, 2005

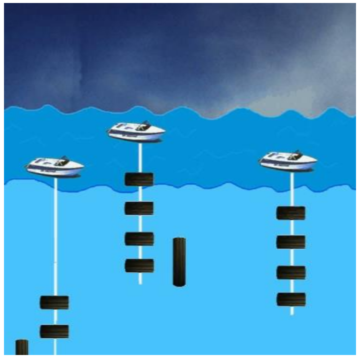
Ecosystems, Biocomplexity



Contaminant Transport



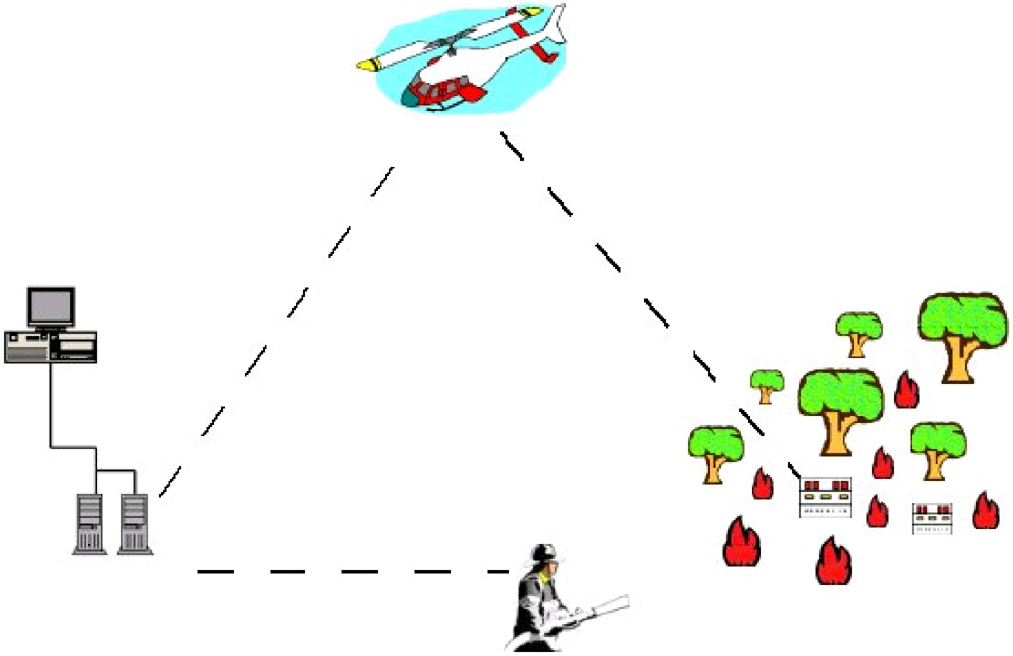
Seismic Structure Response



Marine Microorganisms

Example Application: Firebug

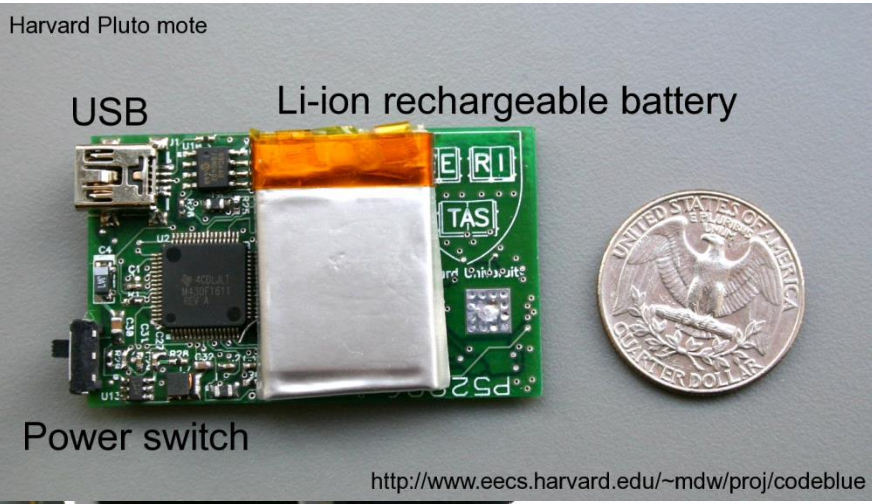
The FireBug system is composed of a network of GPS-enabled, wireless thermal sensors, a control layer for processing sensor data, and a command center for interactively communicating with the sensor network. (http://firebug.sourceforge.net)

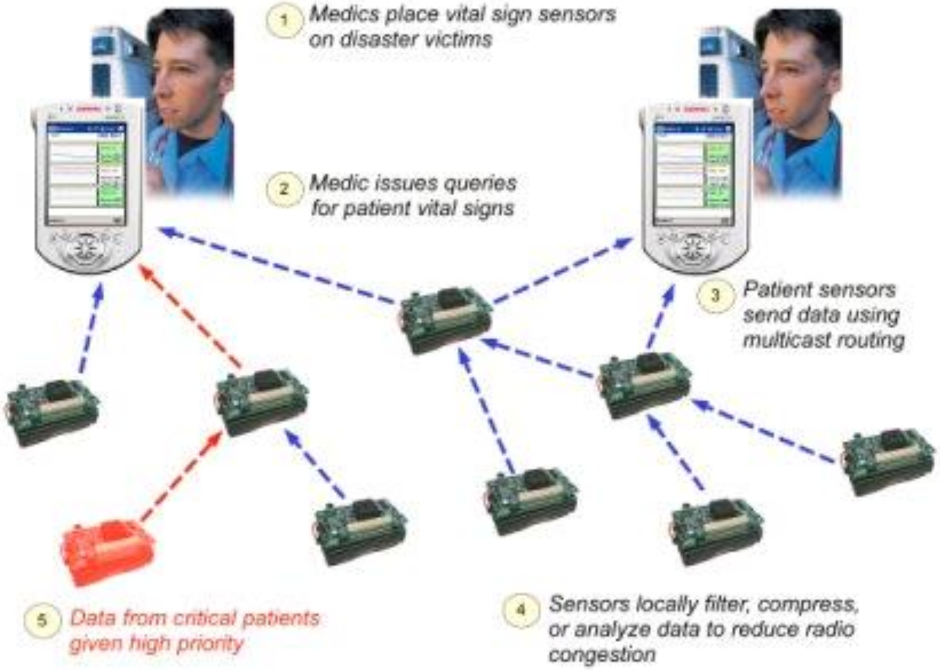


Challenges:

1) How to detect fire (i.e., processing the sensor readings)
2) Harden the sensors (collect more info longer)
3) Ensure sensor nodes work

Example Application: CodeBlue

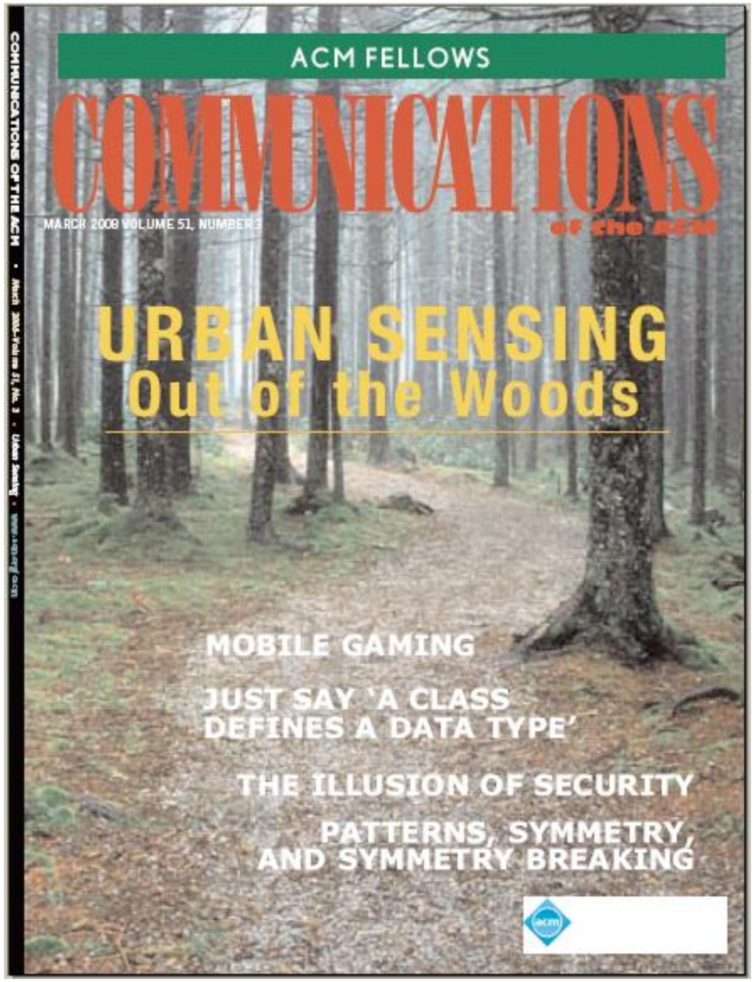




Collect heart rate (HR), oxygen saturation (SpO2), and EKG data,
relay it over a short-range (100m) wireless network to any number of receiving devices
Display data in real time. The sensor devices process the vital sign data, for example, raise an alert condition when vital signs fall outside of normal parameters.
Any adverse change in patient status signaled to a nearby medical expert.

Challenges: wearable sensors, reliability, QoS/prioritization, privacy, ….

Communications of the ACM, March 2008



Cover story: collect information from users in urban environments
-Rich sensing infrastructure (CCTV, RFID, …)
-Cellphones: location, sound, sight (camera)
-additional sensors in cellphones

New data collection paradigm:
-Previous: fully centralized sensing
-Now: participatory sensing, distr. sensing

Raises questions about who owns and has access to what data, build data repositories/commons

Web 2.0 and OSS model:
Participatory collaborative efforts between citizens and scientists, artists, business, …

May or may not be a good thing

Runes (EU project on Tunnel Safety)

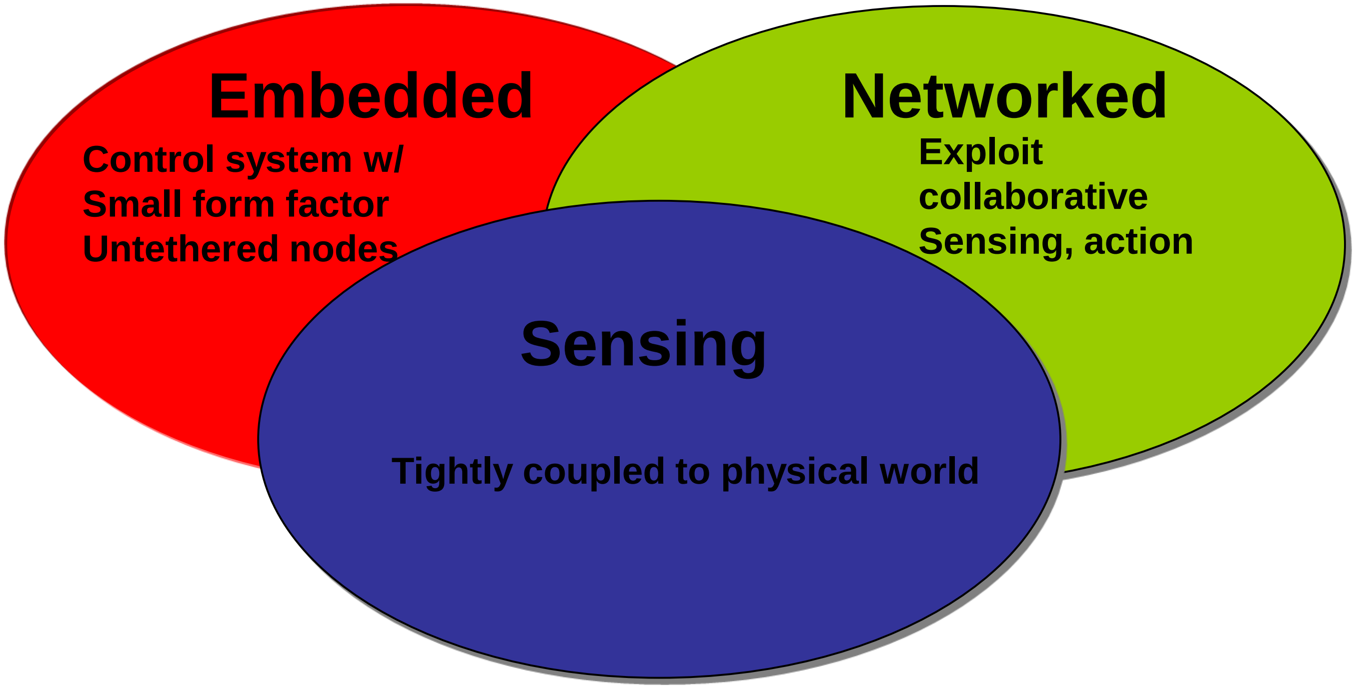
WSN Applications: Summary

- Original applications very much driven by this idea of a "new instrument" to advance science
  - See also NEON (http://www.neoninc.org/), the National Ecological Observatory Network
- Later, more applications about impacting everyday life in new and useful ways (includes humans "in the loop" very much)
  - FireBug: monitor for fire fronts, help firefighters to combat fire, warn them about changes, etc.
  - CodeBlue: monitor patients, alert medical personal
  - Smart Homes: allow elderly/sick to stay at home, in their usual environment
  - RUNES: provide improved tunnel safety
- All applications above are still "single-purpose WSN": designed for one purpose, run by a single organization, very centralized approach
- What happens if single WSN supports many different applications for different users (resource scheduling and contention), application requires data from multiple sensors/domains/networks, …

Enabling Technologies

Embed numerous distributed devices to monitor and interact with physical world

Network devices to coordinate and perform higher-level tasks



Embedded

Control system w/
Small form factor
Untethered nodes

Networked

Exploit collaborative
Sensing, action

Sensing

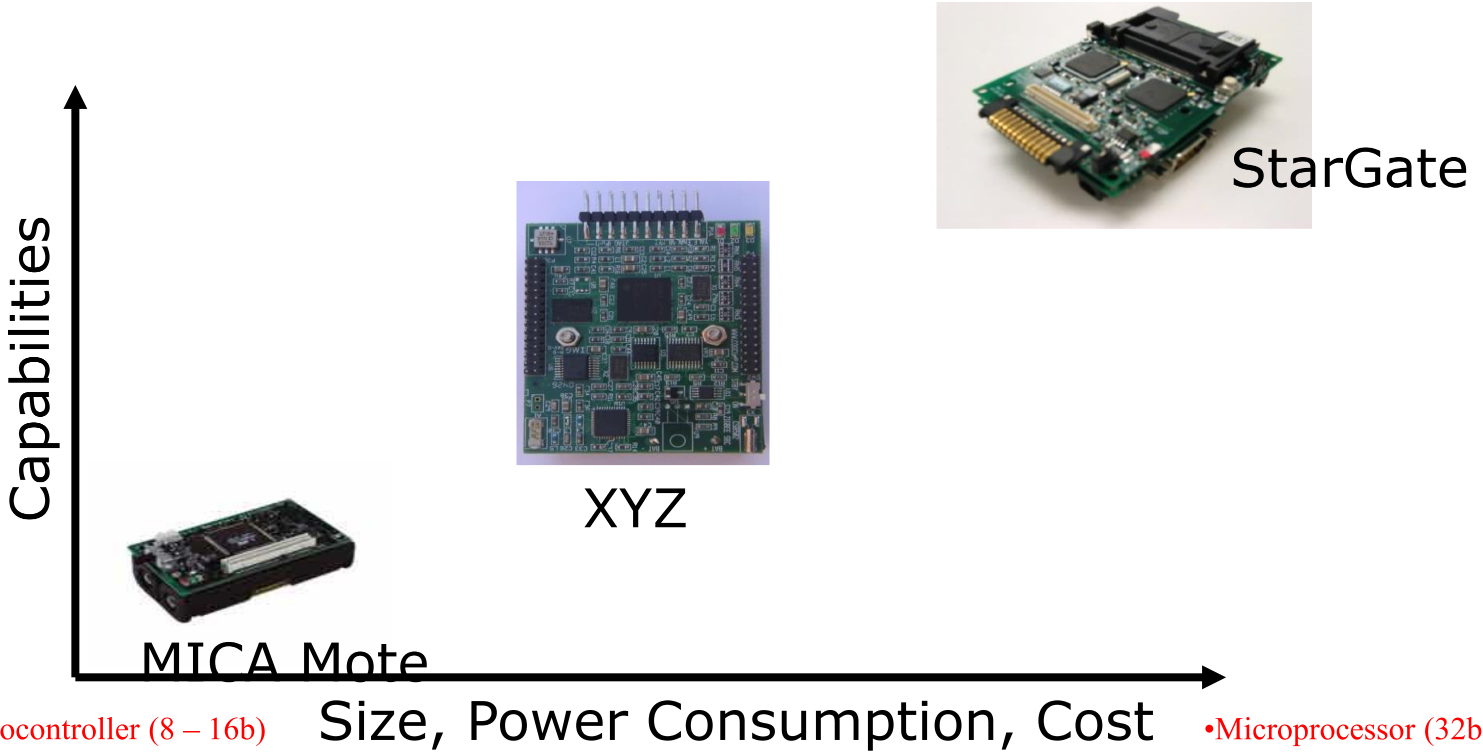
Tightly coupled to physical world

Exploit spatially and temporally dense, in situ, sensing and actuation

Sensors

- Vast majority of work uses Sensors, few examples include actuators
- Used to be taken for granted, but
  - Sensor capabilities determine core WSN functionality (hence new sensor developed all the time)
  - Sensors may consume a significant amount of energy (cameras, for example)
  - Provide very different types and amount of data, may be "programmable"/controllable
  - Biomedical sensors have unique constraints
  - Calibrating sensors is not trivial even for "trivial" sensors
- Example: Open GeoSpatial Consortium (http://www.opengeospatial.org/)
  - SensorML: version 1.0 approved July 2007
    - The primary focus of SensorML is to define processes and processing components associated with the measurement and post-measurement transformation of observations.
  - TransducerML: version 1.0 approved July 2007
    - TML defines:
      - a set of models describing the response characteristics of a transducer
      - an efficient method for transporting sensor data and preparing it for fusion through spatial and temporal associations

Trade-offs among Nodes



- Microcontroller (8 – 16b)
- Narrow Band radio
- Low bit rate, low performance sensors
- *Long* relative lifetime at continuous operation

- Microprocessor (32b)
- Broad Band radio
- High performance sensors
- *Short* relative lifetime at continuous operation

Various Energy Costs

- Energy/bit >> Energy/op even for short ranges!

Mote-class Node	Transmit	**720 nJ/bit**	Processor	**4 nJ/op**
	Receive	**110 nJ/bit**	**~ 200 ops/bit**	
Microserver-Class Node	Transmit	**6600 nJ/bit**	Processor	**1.6 nJ/op**
	Receive	**3300 nJ/bit**	**~ 6000 ops/bit**	

- Even larger in actuality, due to protocol overheads
  - E.g. Effective energy/bit on Mica2 ≈ 0.01-0.1 mJ

- Energy/bit sent ~ Energy/bit stored
  - 802.15.4 radio (250 kbps): 0.2 µJ/bit (1 byte @ 1.5 µJ, 32 µs)
  - Atlmet flash: 0.4 µJ/bit (1 byte @ 3 µJ, 78 µs)
  - However, for high density Flash storage costs go down

Comparison of Energy Sources

	Power (Energy) Density	Source of Estimates
Batteries (Zinc-Air)	1050 -1560 mWh/cm^3 (1.4 V)	Published data from manufacturers
Batteries(Lithium ion)	300 mWh/cm^3 (3 - 4 V)	Published data from manufacturers
Solar (Outdoors)	15 mW/cm^2 - direct sun 0.15mW/cm^2 - cloudy day.	Published data and testing.
Solar (Indoor)	.006 mW/cm^2 - my desk 0.57 mW/cm^2 - 12 in. under a 60W bulb	Testing
Vibrations	0.001 - 0.1 mW/cm^3	Simulations and Testing
Acoustic Noise	3E-6 mW/cm^2 at 75 Db sound level 9.6E-4 mW/cm^2 at 100 Db sound level	Direct Calculations from Acoustic Theory
Passive Human Powered	1.8 mW (Shoe inserts >> 1 cm^2)	Published Study.
Thermal Conversion	0.0018 mW - 10 deg. C gradient	Published Study.
Nuclear Reaction	80 mW/cm^3 1E6 mWh/cm^3	Published Data.
Fuel Cells	300 - 500 mW/cm^3 ~4000 mWh/cm^3	Published Data.

With aggressive energy management, sensor nodes might live off the environment.

New Design Themes

- Long-lived systems that can be untethered and unattended
  - Low-duty cycle operation with bounded latency
  - Exploit redundancy and heterogeneous tiered systems
- Leverage data processing inside the network
  - Thousands or millions of operations per second can be done using energy of sending a bit over 10 or 100 meters
  - Exploit computation near data to reduce communication
- Self configuring systems that can be deployed ad hoc
  - Un-modeled physical world dynamics makes systems appear ad hoc
  - Measure and adapt to unpredictable environment
  - Exploit spatial diversity and density of sensor/actuator nodes
- Achieve desired global behavior with adaptive localized algorithms
  - Can’t afford to extract dynamic state information needed for centralized control

From Embedded Sensing to Embedded Control

- Embedded in unattended “control systems”
  - Different from traditional Internet, PDA, Mobility applications
  - More than control of the sensor network itself

- Critical applications extend beyond sensing to control and actuation
  - Transportation, Precision Agriculture, Medical monitoring and drug delivery, Battlefied applications
  - Concerns extend beyond traditional networked systems
    - Usability, Reliability, Safety

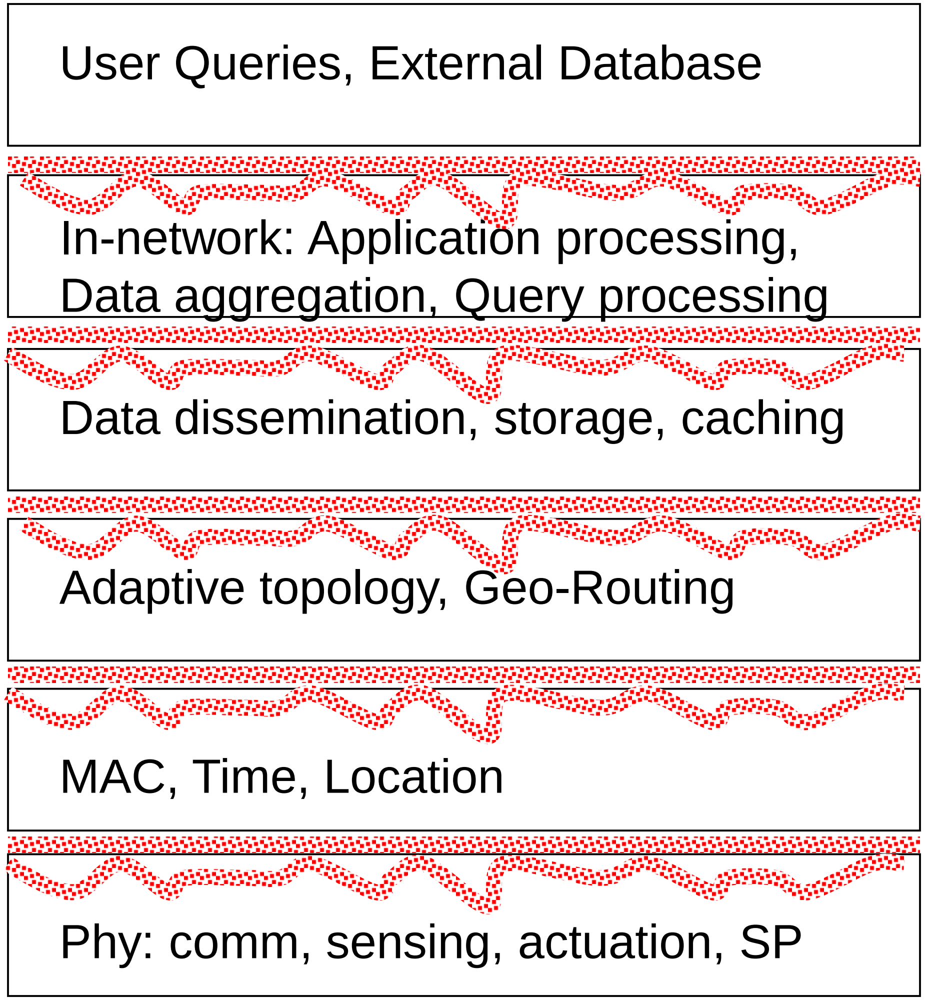
- Need systems architecture to manage interactions
  - Current system development: one-off, incrementally tuned, stove-piped
  - Serious repercussions for piecemeal uncoordinated design: insufficient longevity, interoperability, safety, robustness, scalability...

Sample Layered Architecture

Resource constraints call for more tightly integrated layers

Open Question:

Can we define an Internet-like architecture for such application-specific systems??



User Queries, External Database

In-network: Application processing, Data aggregation, Query processing

Data dissemination, storage, caching

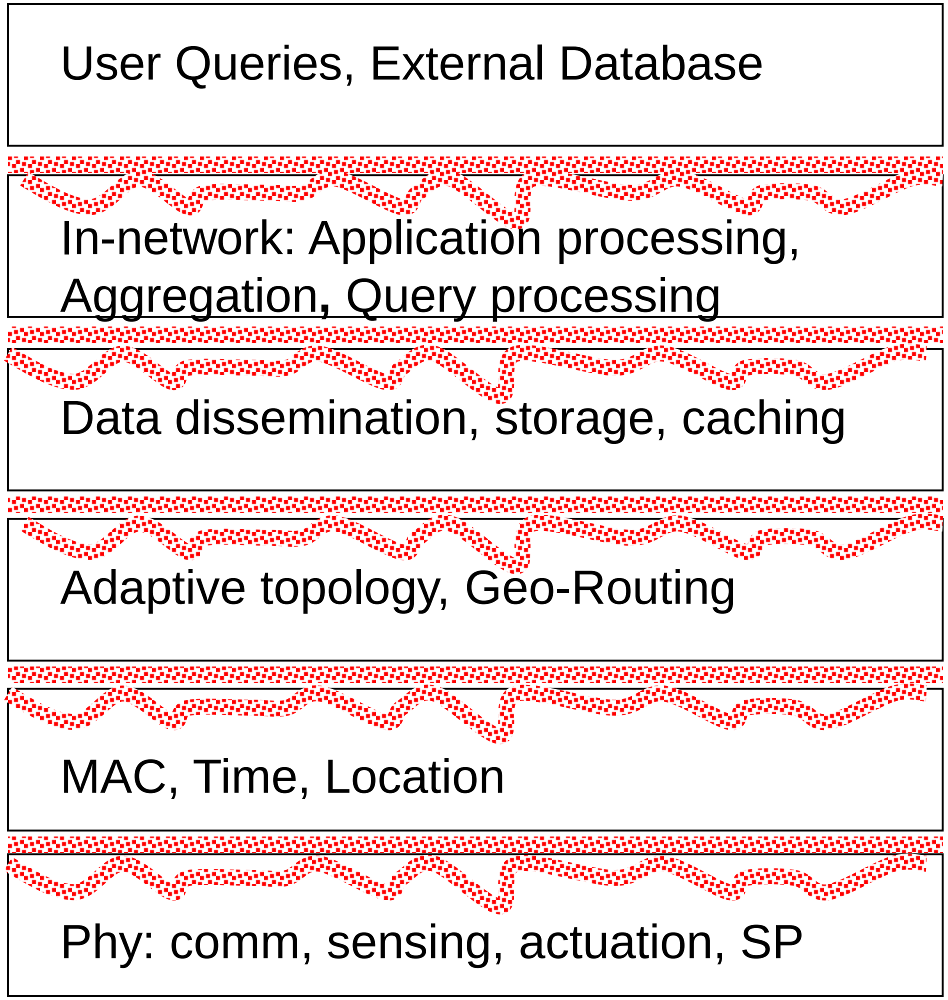
Adaptive topology, Geo-Routing

MAC, Time, Location

Phy: comm, sensing, actuation, SP

Wireless Sensor Network Protocols

- Primary theme: building long-lived, massively-distributed, physically-coupled systems:
  - Coordinating to minimize duty cycle and communication
    - Adaptive MAC
    - Adaptive Topology
    - Routing
  - In-network processing
    - Data centric routing
    - Programming models



User Queries, External Database

In-network: Application processing, Aggregation, Query processing

Data dissemination, storage, caching

Adaptive topology, Geo-Routing

MAC, Time, Location

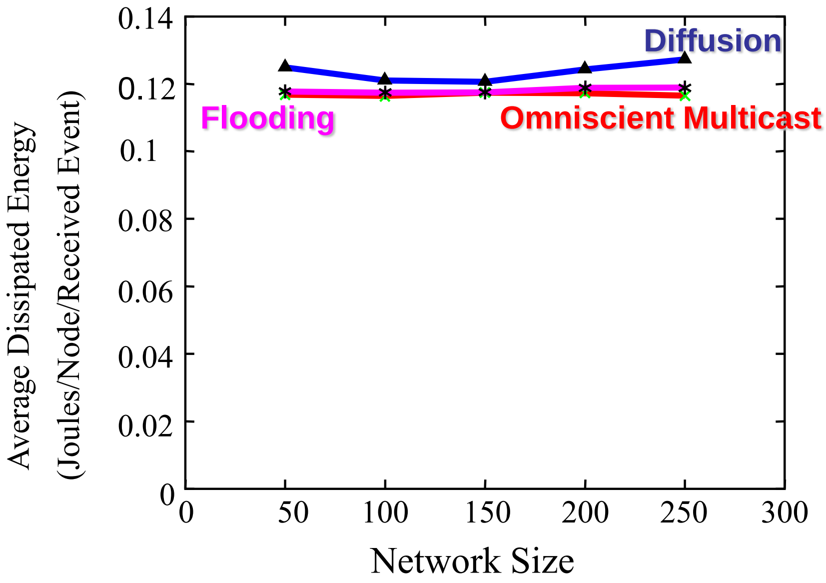
Phy: comm, sensing, actuation, SP

Medium Access Control in Sensor Nets

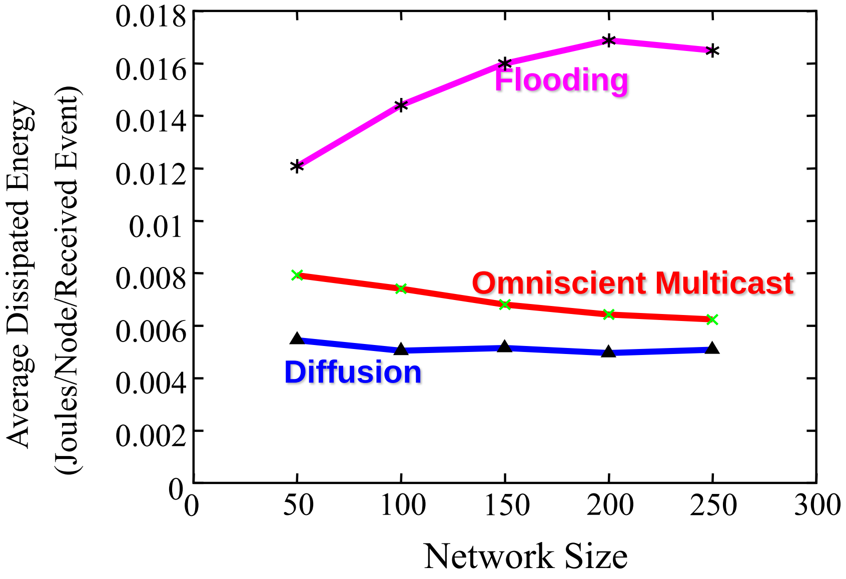
- Important attributes of MAC protocols
  - Collision avoidance
  - Energy efficiency
  - Scalability in node density
  - Latency
  - Fairness
  - Throughput
  - Bandwidth utilization

MAC Impact on Sensor Networks

- Major sources of energy waste
  - Idle listening when no sensing events, Collisions, Control overhead, Overhearing

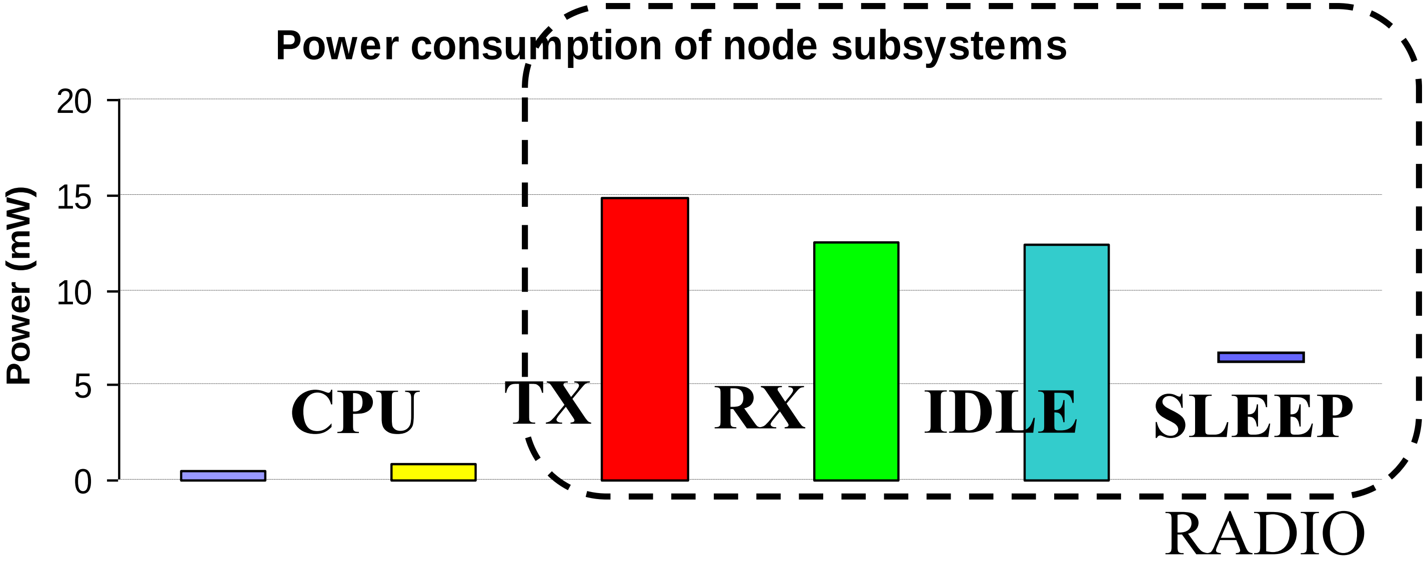


Over 802.11-like MAC



Over energy-aware MAC

Identifying the Energy Consumers



$$E_{TX} \approx E_{RX} \approx E_{IDLE} >> E_{SLEEP}$$

- Need to shutdown the radio

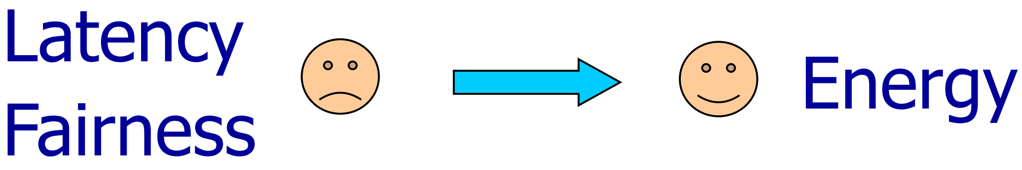
Energy Efficiency in MAC

- Major sources of energy waste
  - Idle listening
    - Long idle time when no sensing event happens
    - Collisions
    - Control overhead
    - Overhearing
- Try to reduce energy consumption from all above sources
- TDMA requires slot allocation and time synchronization
- Combine benefits of TDMA + contention protocols

Common to all wireless networks

Sensor-MAC (S-MAC) Design

- Tradeoffs



Latency
Fairness

Energy

- Major components of S-MAC
  - Periodic listen and sleep
  - Collision avoidance
  - Overhearing avoidance
  - Message passing

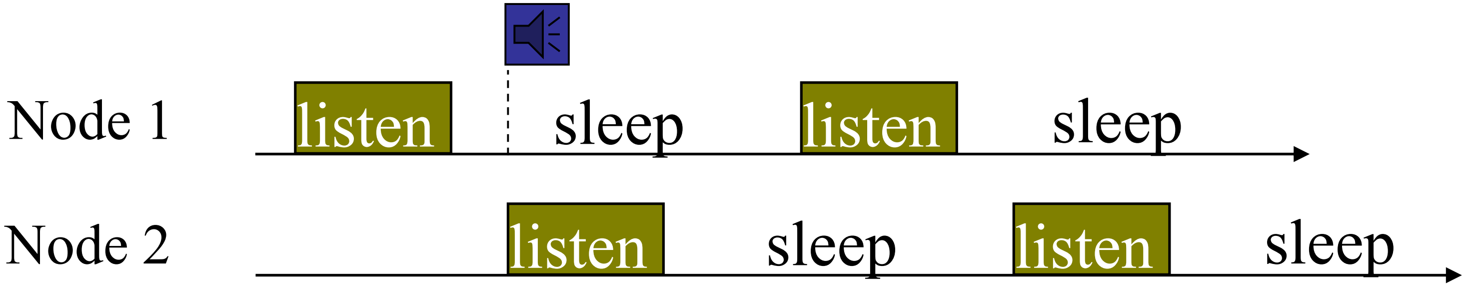
Periodic Listen and Sleep

- Problem: Idle listening consumes significant energy
  - Nodes do not sleep in IEEE 802.11 ad hoc mode

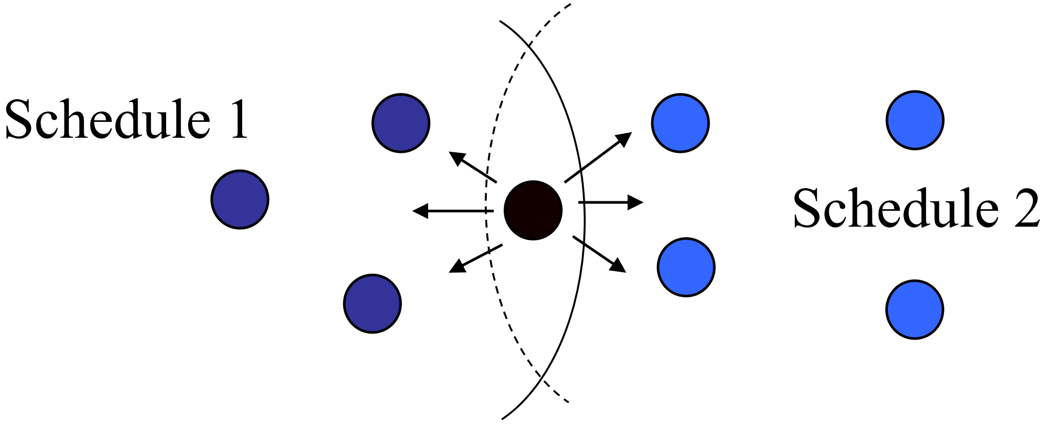
- Solution: Periodic listen and sleep
  - Turn off radio when sleeping
  - Reduce duty cycle to ~10% (200 ms on/2s off)
  - Increased latency for reduced energy

Periodic Listen and Sleep

- Schedules can differ



- *Preferable if* neighboring nodes have same schedule
  — easy broadcast & low control overhead



Border nodes:
two schedules,
broadcast twice

Periodic Listen and Sleep

- Schedule maintenance
  - Remember neighbors' schedules
    - to know when to send to them
  - Each node broadcasts its schedule every few periods
  - Refresh on neighbor's schedule when receiving an update
  - Schedule packets also serve as beacons for new nodes to join a neighborhood

Collision Avoidance

- Problem: Multiple senders want to talk
- Options: Contention vs. TDMA
- Solution: Similar to IEEE 802.11 ad hoc mode (DCF)
  - Physical and virtual carrier sense
  - Randomized backoff time
  - RTS/CTS for hidden terminal problem
  - RTS/CTS/DATA/ACK sequence

Overhearing Avoidance

- Problem: Receive packets destined to others
- Solution: Sleep when neighbors talk
  - Basic idea from PAMAS
  - But we only use in-channel signaling
- Who should sleep?
  - All immediate neighbors of sender and receiver
- How long to sleep?
  - The duration field in each packet informs other nodes the sleep interval

Message Passing

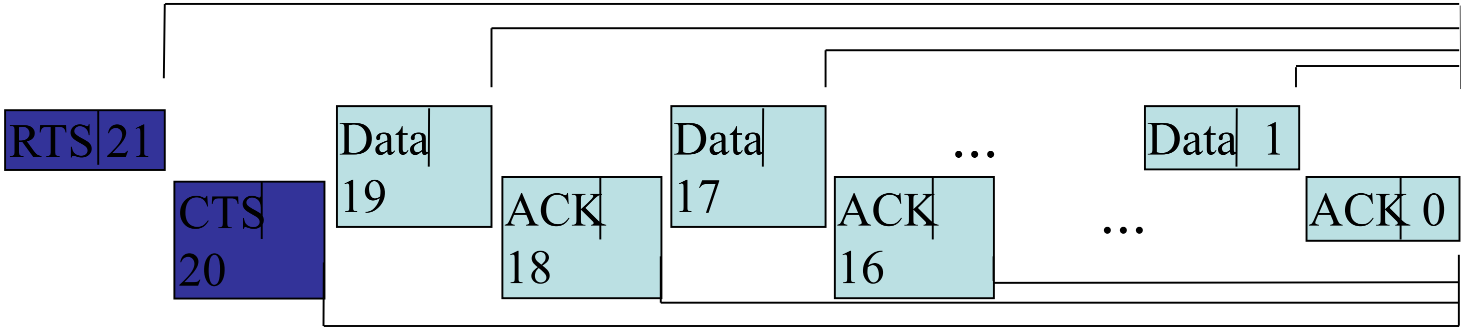
- Problem: In-network processing requires entire message
- Solution: Don’t interleave different messages
  - Long message is fragmented & sent in burst
  - RTS/CTS reserve medium for entire message
  - Fragment-level error recovery
    - extend Tx time and re-transmit immediately
- Other nodes sleep for whole message time

Fairness ☹ → ☺ Energy
Message-level latency

Message Passing vs. 802.11 Fragmentation

Time reservation by *duration* field

- MP



RTS 21

CTS 20

Data 19

ACK 18

Data 17

ACK 16

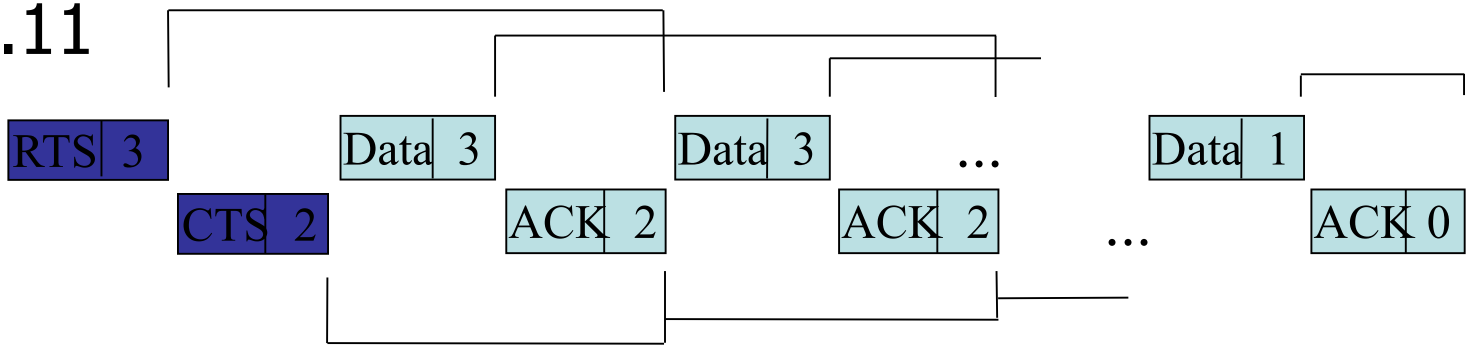
…

…

Data 1

ACK 0

- 802.11



RTS 3

CTS 2

Data 3

ACK 2

Data 3

ACK 2

…

…

Data 1

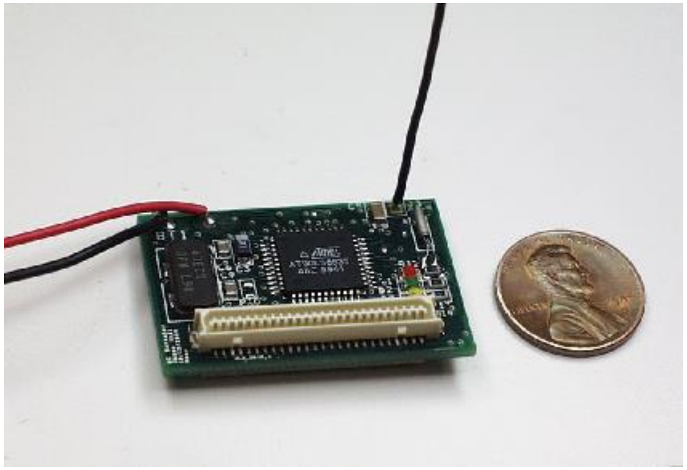
ACK 0

✓ If ACK is not received, give up Tx — fairness

✓ No indication of entire time — other nodes keep listening

Implementation on Testbed Nodes

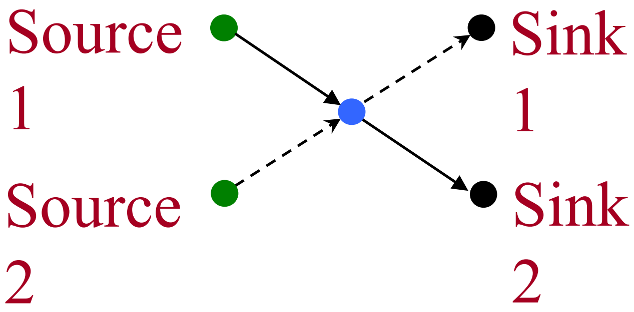
- Platform
  - Motes (UC Berkeley) :
    - 8-bit CPU at 4MHz,
    - 8KB flash, 512B RAM
  - TinyOS: event-driven
  - Also used as NIC for 32-bit embedded PCs



- Compared MAC modules
  - IEEE 802.11-like protocol
  - Message passing with overhearing avoidance
  - S-MAC (2 + periodic listen/sleep)
  - URL: http://www.isi.edu/scadds/smac/

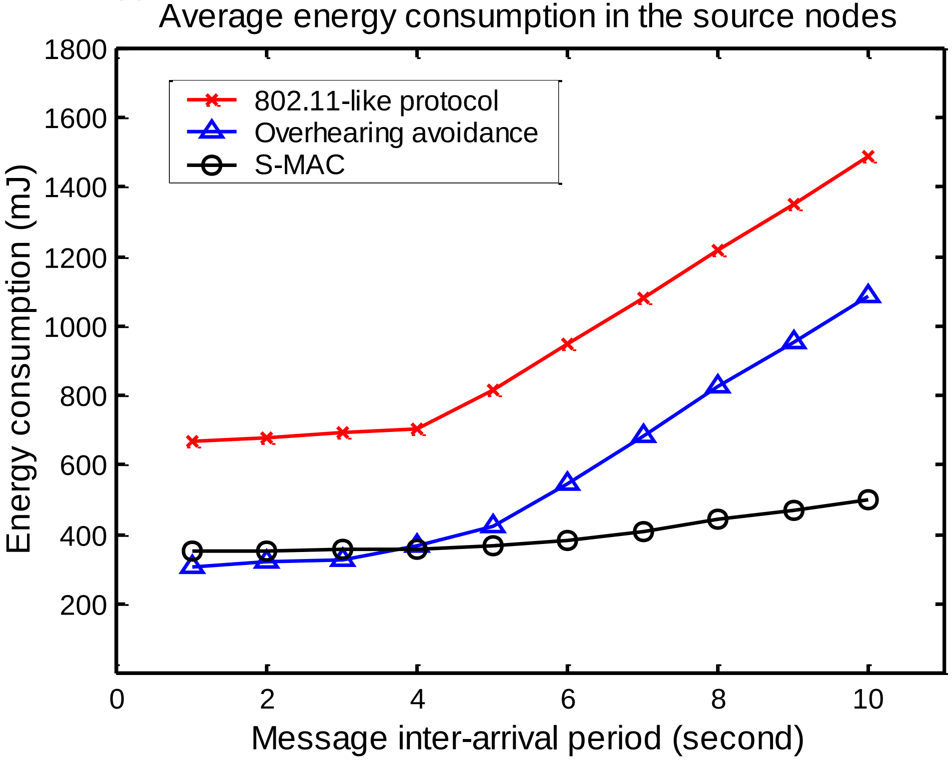
S-MAC Experimental Results

- Topology and measured energy consumption on source nodes



- Each source node sends 10 messages
  - — Each message has 10 fragments x 40B
- Measure total energy
  - — Data + control + idle

Energy consumed



Message Inter-arrival period

PW-MAC

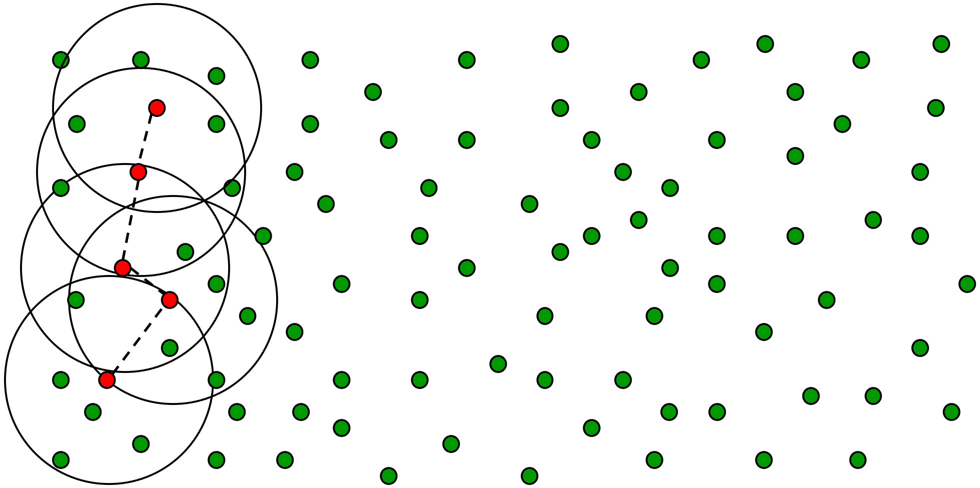
- Refinement of that idea: PW-MAC: An Energy-Efficient Predictive-Wakeup MAC Protocol for Wireless Sensor Networks
- Key idea: each node picks a random sleep cycle, protocol is receiver-initiated
  - Sender-initiated: sender sends (long) preamble to wake up receiver
  - Receiver-initiated: receiver send (short) beacon when waking up
- PW-MAC: nodes exchange seeds of a random number generator, so a sender can predict when an intended receiver will wake up next, will wake up "just before" to be ready to hear the beacon
  - Need to have a mechanism to exchange this state info, see paper
  - Need to deal with clock drifts and other sources of noise
- Protocol implemented and evaluated on MICAz motes, shows consistently good performance

Adaptive Topology

- Can we do more than shut down radio in between transmissions/receptions?
- Can we put nodes to sleep for longer periods of time?
- Goal:
  - Exploit high density (over) deployment to extend system lifetime
  - Provide topology that adapts to the application needs
  - Self-configuring system that adapts to environment without manual configuration

Adaptive Topology: Problem Description

- Simple Formulation (Geometric Disk Covering)
  - Given a distribution of N nodes in a plane.
  - Place a minimum number of disks of radius r (centered on the nodes) to cover them.
  - Disk represents the radio connectivity (simple circle model).
- The problem is NP-hard.

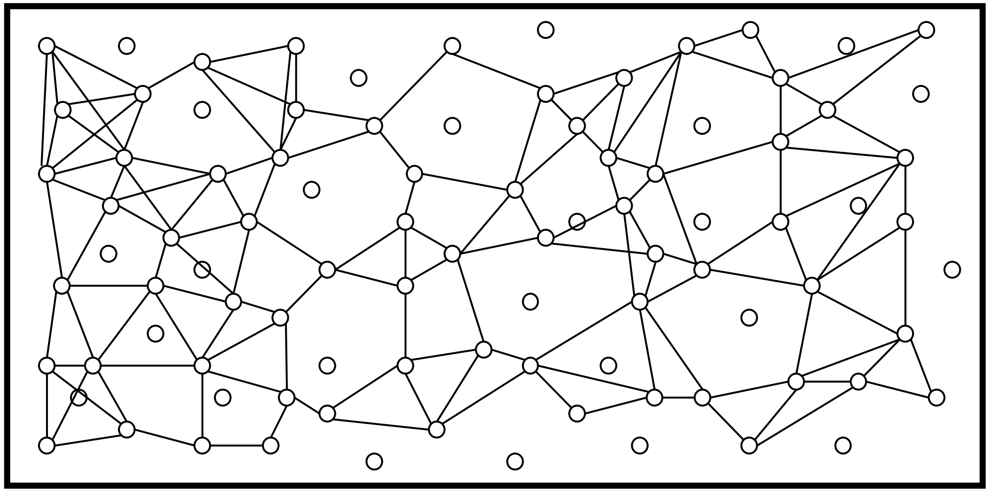


Adaptive Topology: Consider Sensing Range

- Each sensor can detect events in its local environment
- Typically modelled as sensor having a SENSING RANGE
  - Determined by the type of sensor
  - Independent of the sensor node's transmission range
- Typically, we are interested in ensuring that the monitored area is COVERED, not the nodes placed in the area CONNECTED
- If TRANSMISSION RANGE >= 2 * SENSING RANGE, COVERAGE implies CONNECTIVITY
- More general (nodes are cheap and prone to failure):
  - How to ensure k-coverage and or k-connectivity

Tradeoff

- How many nodes to activate?
  - few active nodes:
    - distance between neighboring nodes high -> increase packet loss and higher transmit power and reduced spatial reuse;
    - need to maintain sensing coverage (whole research area on coverage/exposure)
  - too many active nodes:
    - at best, expending unnecessary energy;
    - at worst nodes may interfere with one another by congesting the channel.

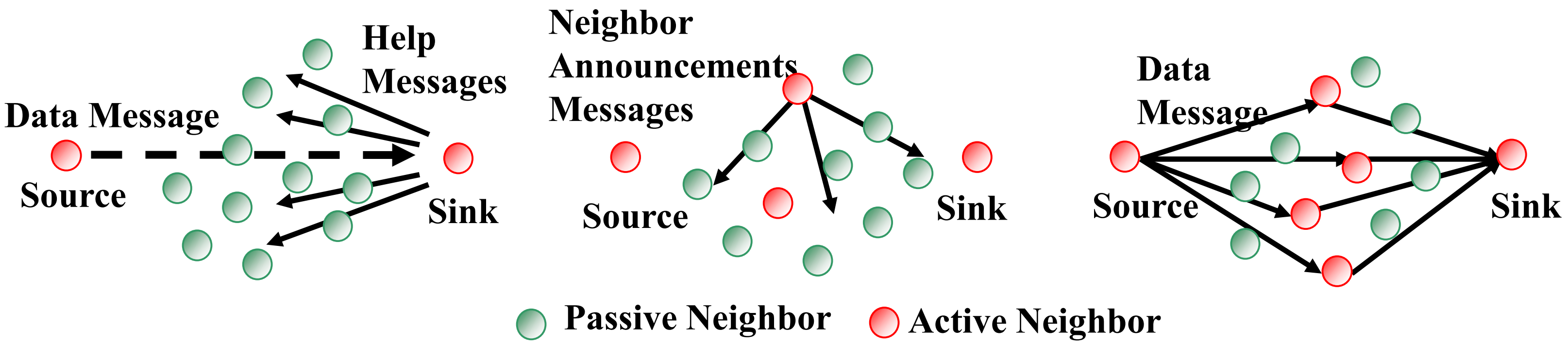


Adaptive Topology Schemes

- Mechanisms being explored:
  - Empirical adaptation: Each node assesses its connectivity and adapts participation in multi-hop topology based on the measured operating region, ASCENT
  - Cluster-based, load sharing within clusters, CEC
  - Routing/Geographic topology based, eliminate redundant links, SPAN, GAF
  - Data/traffic driven: Trigger nodes on demand using paging channel, STEM

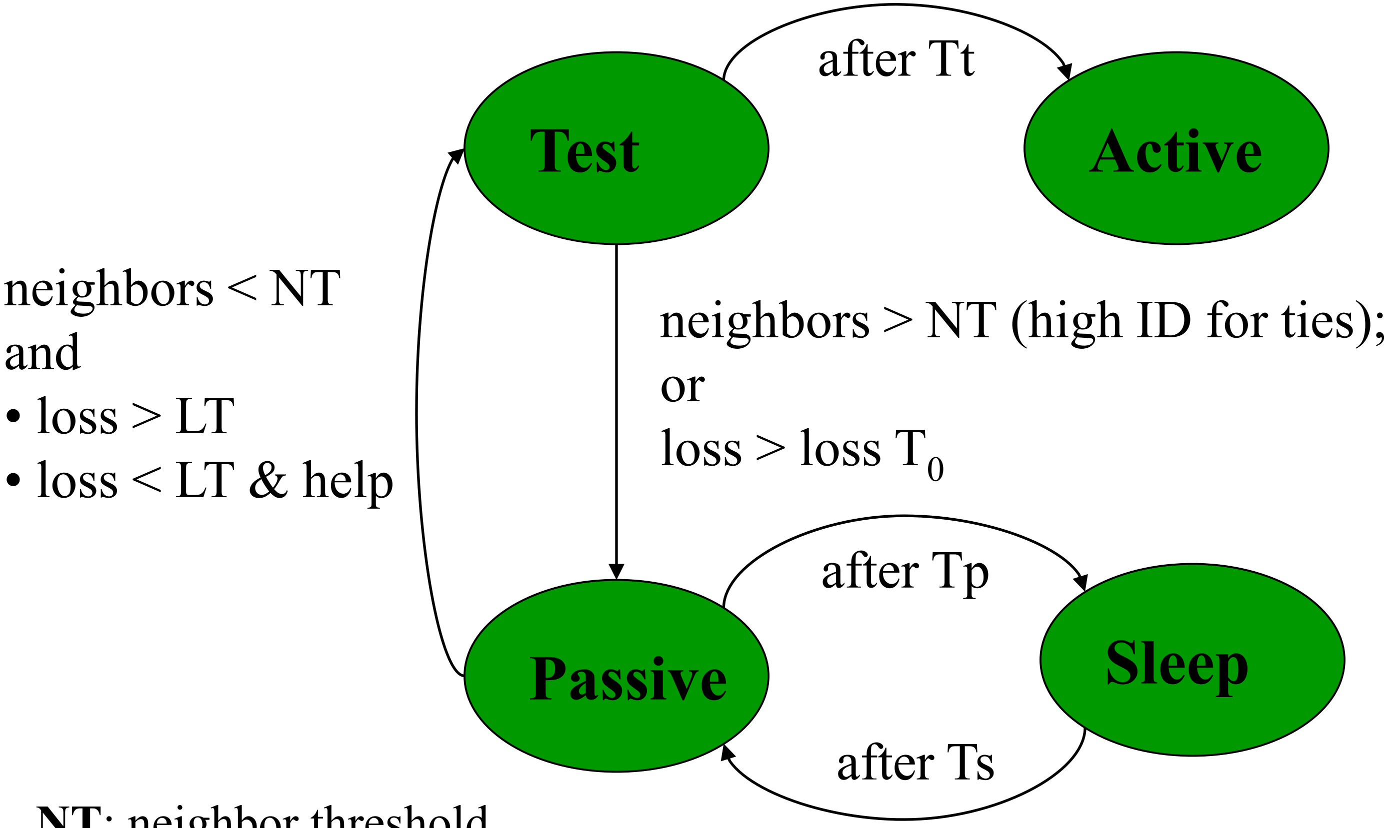
One example algorithm: ASCENT

- The nodes can be in ***active*** or ***passive*** state.
  - Active nodes forward data packets (using routing mechanism that runs over topology).
  - Passive nodes ***do not*** forward any packets but may sleep or collect network measurements.
- Each node ***joins*** network topology or ***sleeps*** according to measured number of neighbors and packet loss, as ***measured locally***.



(a) Communication Hole (b) Self-configuration transition (c) Final State

State Transitions



neighbors < NT
and
- loss > LT
- loss < LT & help

neighbors > NT (high ID for ties);
or
loss > loss T_0

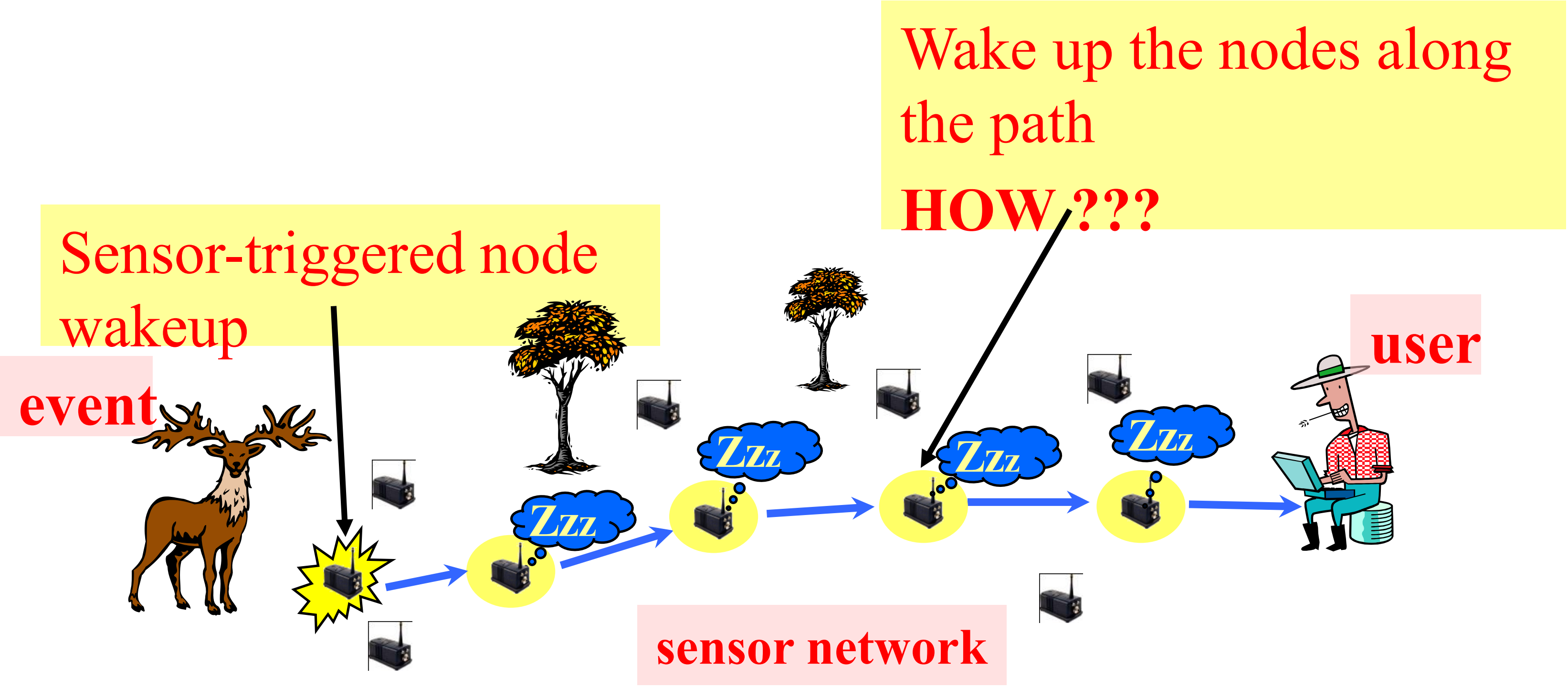
NT: neighbor threshold

LT: loss threshold

T?: state timer values (p: passive, s: sleep, t: test)

	Goal (general: energy savings)	Routing dependency	Assumptions
GAF	preserve routing fidelity	none	geographic information for grid placement radio connectivity directly correlated with geography
SPAN	preserve capacity of the raw topology	gets connectivity matrix and neighbors from routing requires modifications in the routing lookup process	802.11 MAC with Power Savings mode
STEM	tradeoff latency for energy savings	needs routing info to direct the wake-up wave	2 radios/wake-up channel connectivity conditions remain constant in sleeping periods
ASCENT	adapt topology based on application needs	none	radio supports promiscuous mode

STEM: Data drive wakeup


Wake up the nodes along the path

HOW ???

Sensor-triggered node wakeup

event

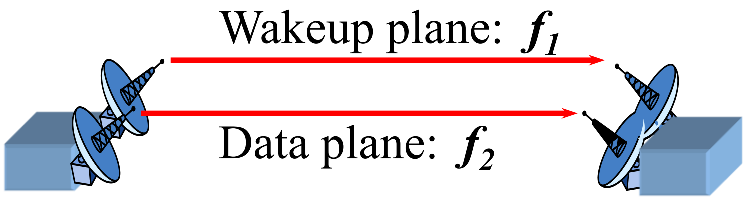
user

sensor network

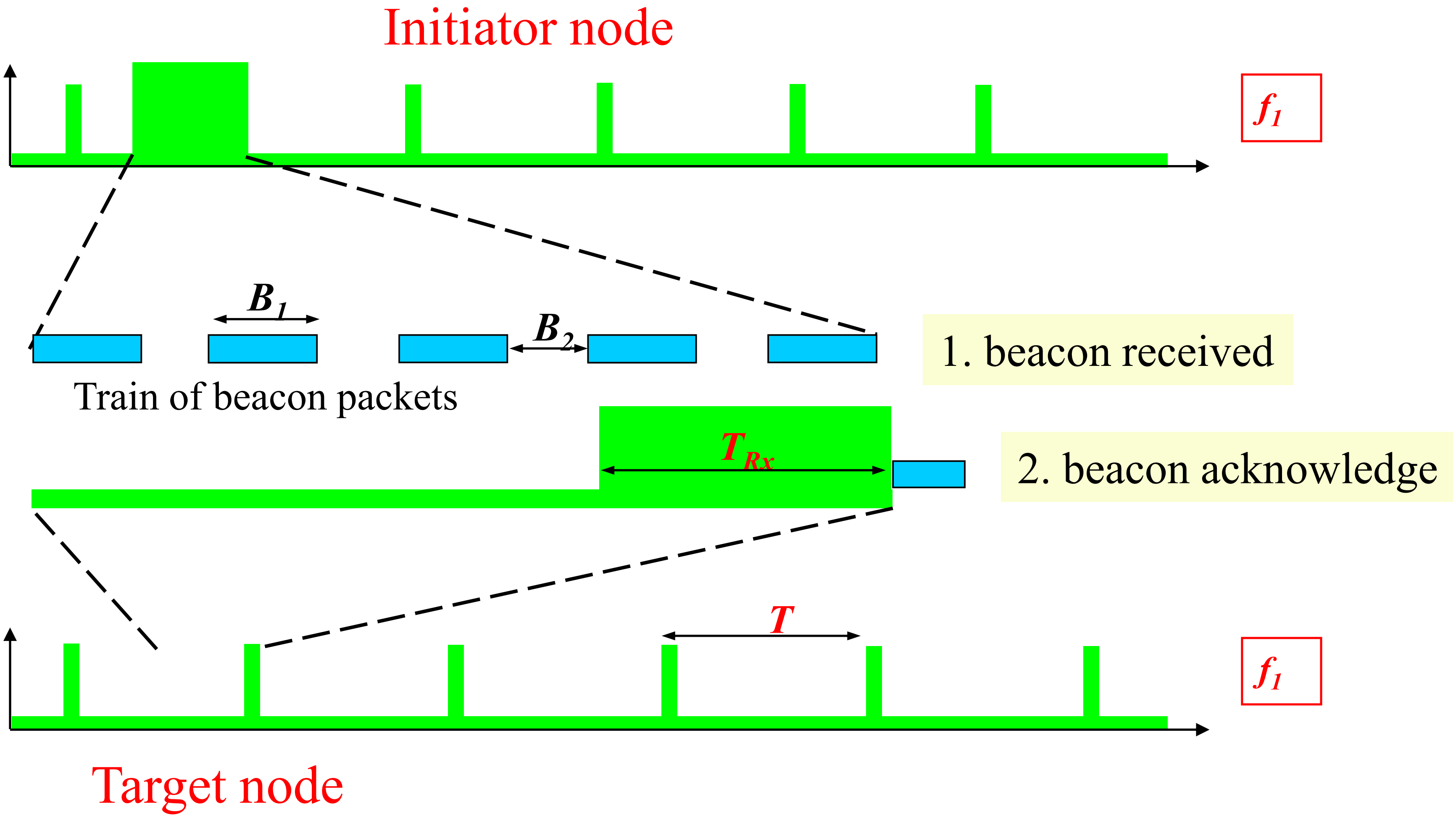
Path nodes need to be woken up

STEM: Sparse Topology and Energy Management

- Need to separate Wakeup and Data Forwarding Planes
- Chosen two separate radios for the two planes
- Use separate radio for the paging channel to avoid interference with regular data forwarding
- Trades off energy savings for path setup latency



Duty Cycled Wakeup Radio



Initiator node

Train of beacon packets

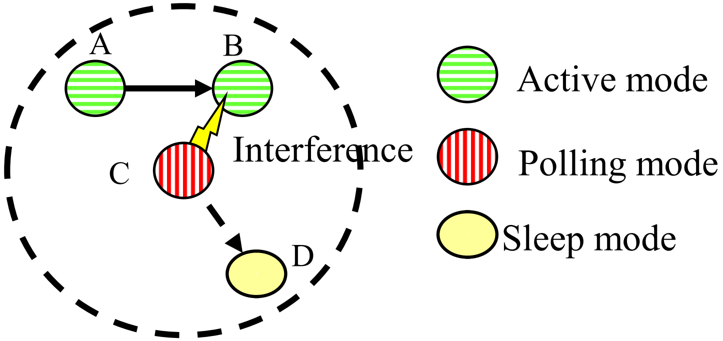
1. beacon received

2. beacon acknowledge

Target node

STEM Design Decisions

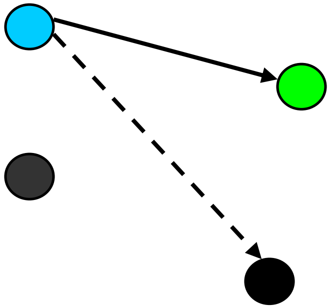
- One radio with one frequency band would cause interference between the wakeup and data planes



- One radio with two frequencies would have to change frequency to listen for setup requests
- One radio with time multiplexed wake-up and data planes would require time synchronization between the two nodes – time sync in ad-hoc networks is an open problem by itself
- The additional radio cost is about 15% of the total cost of the node

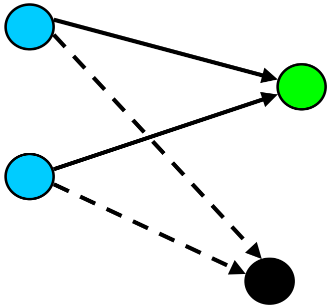
Design Issue: Collision Resolution

1 initiator node



- beacon received correctly
- only intended receiver turns on the data radio and sends a beacon acknowledge in the wakeup plane

more initiator nodes



- upon detection of collision, a node turns on its data radio
- after *T*, the initiator node assumes the target node is up and contacts it on the data plane
- when an expected target node doesn't receive data, it times out and goes back to sleep

Routing

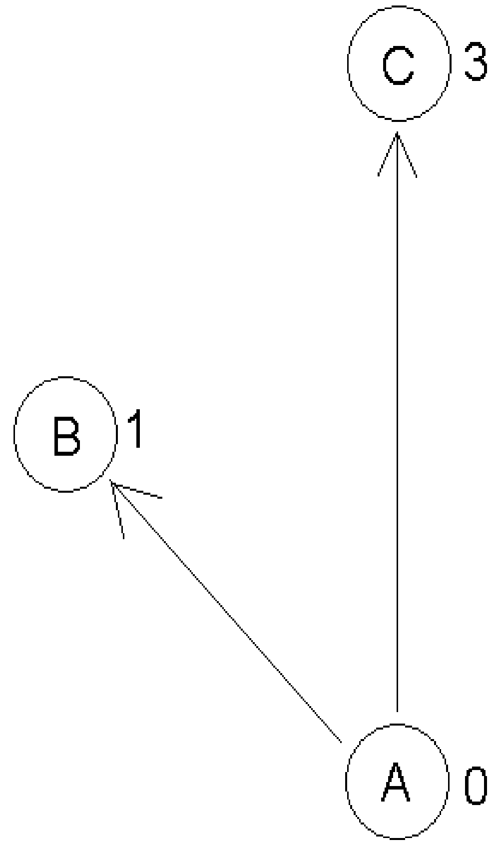
- Given a topology, how to route data?
  - MANET: Reactive [DSR, AODV], proactive [OLSR], TORA, GPSR
  - Location-aided routing: Geocast, Cartesian-LAR

  - Building on Geo Routing
    - GRAB
    - Routing on curve

GRAB: Field Based Minimum Cost Forwarding

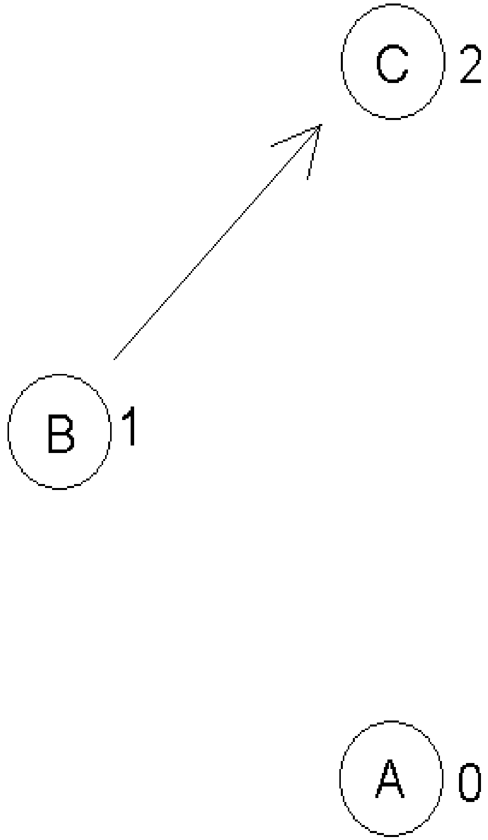
- Each node broadcasts only once
- Cost Function
  - A measure of how expensive it is to get a message back to the sink.
  - Could be based on:
    - Energy needed in radio communication.
    - Hop count.
    - …
- Node Cost
  - Each node keeps a best estimate on its minimum cost.
  - Estimate updated upon receipt of every ADV message.
  - ADV message forwarding deferred for time proportional to nodes cost estimate.

ADV Dissemination Example



- Signal strength is used to measure cost.
- B sees strong signal and judges cost to be 1.
- C sees weak signal and judges cost to be 3.

ADV Dissemination Example contd.



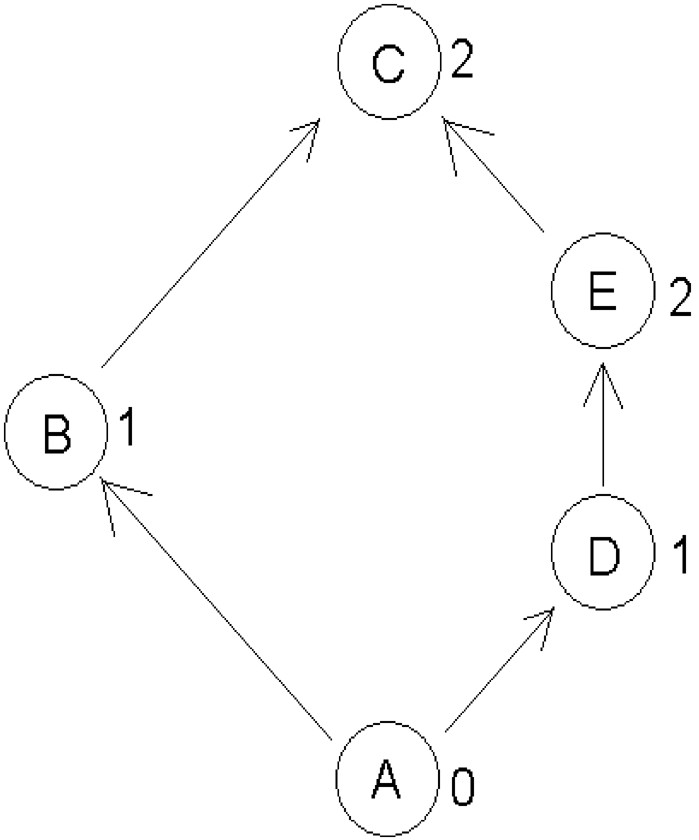
- Because B has a smaller cost, it defers for a shorter time then C.
- C updates itscost to 2 and restarts its deferral timer.
- Each node has optimal cost with minimum broadcast.

Data Dissemination

- A node that decides it has interesting data broadcasts two things (besides data)
    - Total budget to get back to sink.
    - Amount of budget used in initial broadcast.

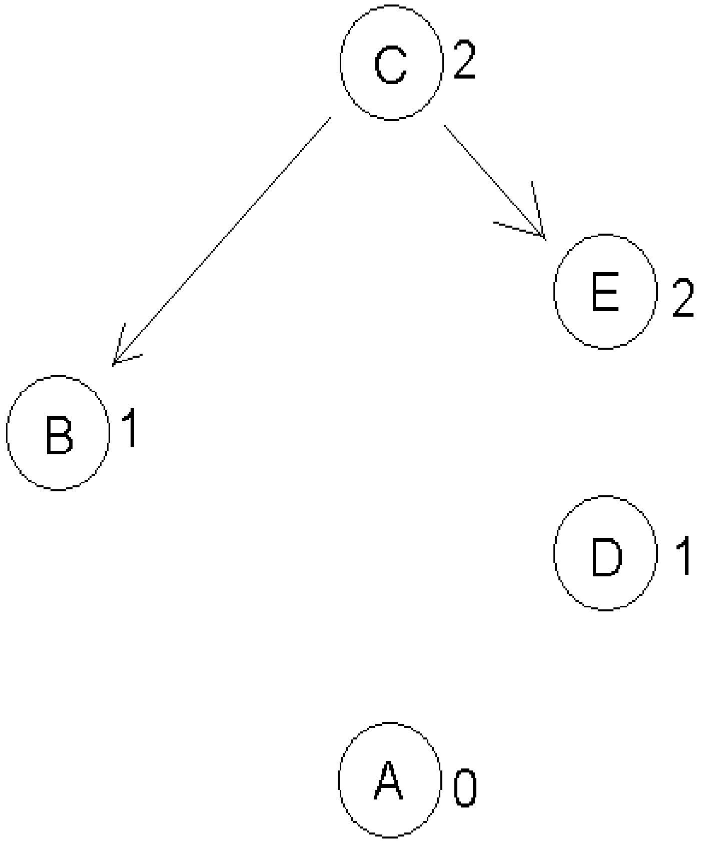
- A node receiving a data message will only forward a data message if Total Budget ≥ Budget Spent So Far + My Cost
    - If the inequality holds then Budget Spent So Far is updated.
    - Otherwise the message is dropped.

Data Dissemination Example



- Assume hop count was used as a cost metric.
- Node A is the sink.
- Node C is the source.

Data Dissemination Example contd.



- Node C sends a data message which specifies
  - Total Budget = 2
  - Budget Spent = 1
- Node E drops message
  - TB < BS + E's Cost
- Node B forwards message.

Routing on a Curve

- Trajectories are a natural name space for embedded networks
- By definition, network structure mimics physical structure that is instrumented
  - Stress along a column
  - Flooding along a river
  - Pollution along a road
- Trajectories come from the application domain

In-Network Processing: Key to Sensor Network Scalability and Realization

- Gupta and Kumar pointed out fundamental limits of large scale wireless networks (per node throughput O(1/sqrtN)
- However, S. Servetto shows that result holds only for independent nodes (Mobicom 2002)
  - Densely deployed sensor network data will be correlated and can be aggregated
- Scalability and lifetime will depend on techniques for in-network processing of data
  - Naming Data: Directed Diffusion
  - Data base perspectives:TAG, Sylph
  - Programming mechanisms: Sensorware, Mate

Directed Diffusion: Data Centric Routing

- Basic idea
  - name data (not nodes) with externally relevant attributes
    - Data type, time, location of node, SNR, etc
  - diffuse requests and responses across network using application driven routing (e.g., geo sensitive or not)
  - optimize path with gradient-based feedback
  - support in-network aggregation and processing
- Data sources publish data, Data clients subscribe to data
  - However, all nodes may play both roles
    - A node that aggregates/combines/processes incoming sensor node data becomes a source of new data
    - A sensor node that only publishes when a combination of conditions arise, is a client for the triggering event data
  - True peer to peer system

Data Centric vs. Address Centric

- Address Centric
  - Distinct paths from each source to sink.
- Data Centric
  - Support aggregation in the network where paths/trees overlap
  - Essential difference from traditional IP networking
- Building efficient trees for Data centric model
  - Aggregation tree: On a general graph if k nodes are sources and one is a sink, the aggregation tree that minimizes the number of transmissions is the minimum Steiner tree. NP-complete….Approximations:
    - Center at Nearest Source (CNSDC): All sources send through source nearest to the sink.
    - Shortest Path Tree (SPTDC): Merge paths.
    - Greedy Incremental Tree (GITDC): Start with path from sink to nearest source. Successively add next nearest source to the existing tree.

Summary

- Energy is of utmost importance
  - Survey paper: other networks (cellular networks) have battery-operated devices, but user will be able to recharge/change battery
  - Here: want to run network off the battery for LONG durations (months or years)
  - Need to conserve energy shows up in all sorts of protocols
    - MAC: duty-cycle the radio
    - Topology-control: use only a subset of nodes
    - In-network processing/aggregation: do not send a message unless you have to
- Similar to MANET: networks should be self-managed and are infrastructure-less
- Different from MANETs: potentially much denser and larger scale
- Need for new or improved versions of existing protocols: topology control, localization, clock synchronization
  - Either other protocols leverage this (location-based routing, TDMA-like MAC, …
  - Applications will want to have access to that: when and where did an event happen