UBICOMP
Episode 9: Sensor Networks

Hannes Frey and Peter Sturm
University of Trier

Outline

- Introduction
- Communication Architecture
- Platform classes
- Data centric routing
- Security in sensor networks
- Sensor Information Networking Architecture (SINA)
- The Smart Dust Project

References

[1] I. F. Akyildiz et al., "A survey on Sensor Networks", IEEE Communications Magazine, August 2002
[2] C.-C. Shen et al., "Sensor Information Networking Architecture and Applications", IEEE Personal Communications, August 2001
[3] J. M. Kahn et al., "Next Century Challenges: Mobile Networking for "Smart Dust"", MOBICOM 1999
[4] Communications of the ACM, June, 2004, Vol. 47, No. 6, "Wireless Sensor Networks", p. 30-53

UBICOMP

Introduction

Introducing Sensor Networks

- Definition of sensor networks
  - Large number of small and low cost sensor nodes
    - sensing, processing, and wireless communication capabilities
  - Densely deployed inside/close to the phenomenon
  - Node position not engineered or predetermined
    - Deployment in inaccessible terrain or disaster relief
    - Protocols and algorithms with self-organization capabilities
    - Nodes have to cooperate and partially process sensed data



Introducing Sensor Networks

- Sensor Types
  - Seismic
  - Magnetic
  - Thermal
  - Visual
  - Infrared
  - Acoustic
  - Radar
- Sensing
  - Continuous
  - Event detection
  - Location sensing
  - Actuator control
- Monitored ambient conditions
  - Temperature
  - Humidity
  - Vehicular movement
  - Lightning condition
  - Pressure
  - Soil makeup
  - Noise levels
  - Presence/absence of objects
  - Mechanical stress level
  - Object speed, direction, size

Possible Sensor Network Applications

- Environmental applications
  - Biology, meteorology, geophysics
  - Agriculture
  - Forest fire detection
  - Flood detection
- Health applications
  - Interfaces for the disabled
  - Telemonitoring of human physiological data
  - Drug administration in hospitals
- Home applications
  - Home automation
  - Smart environment
- Other commercial applications
  - Environmental control in office buildings
  - Interactive museums
  - Monitoring car thefts
  - Managing inventory control
- Military Applications

Example: The Great Duck Island

- Monitoring Storm Petrel activity at Great Duck Island
- Dilemma for Biologists
  - Need multiple measurements of biological parameters at frequent intervals
  - potentially harming their subjects and biasing results
- Solution: "Mote Sensing", using small wireless probes
  - Array of individual Motes, capable of recording temperature, humidity, pressure, and other environmental data
  - Allows to follow nesting activity throughout the season with minimal impact on the birds
- Researchers will need to enter the colony only at the beginning of the study to actually insert the Motes into burrows
- Data transmitted to a base computer at Eno Station for up-link to the web.
- Potential for conservation efforts in small, isolated locations where any human presence is likely to be disruptive, or with species that are particularly sensitive to disturbance.
- http://www.greatduckisland.net

Example: Smart Buildings Admit Their Faults

- Make buildings, bridges, and other structures aware of their own health
- Matchbox-sized Motes can be built to sense numerous factors
  - light and temperature for energy saving applications
  - location to dynamic response (reveal the structural soundness)
- If sensors cost less than $1 and can be installed in minutes, "dense packs" of them can surround all critical beams and columns, providing extremely detailed structural data.
- Recent test at UC Berkeley's Richmond Field Station seismic research laboratory
  - 15 Motes installed in the wood framing of a three-story model apartment building
  - Constructed on a "shake table" that simulates earthquakes
  - During controlled quake, the Motes gathered seismic data from multiple locations in the building
  - Information was then compared to discern the way the tremors spread through the building and how the structure reacted.
- TinyOS already enables Motes to automatically establish their own network and share information as soon as they're switched on
- Eventually, Smart Dust Motes will gain enough brainpower to process the raw data they collect before it even leaves the building.
  - Goal: Let the sensors discuss the data among themselves and tell us where the problems are

UBICOMP (C) 2004 AG SYSOFT – UNIVERSITY OF TRIER

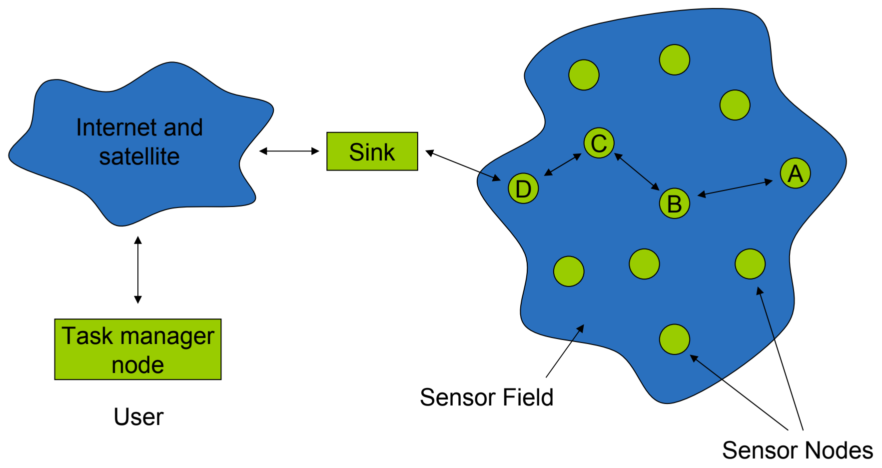
Sensor Networks compared to Ad-Hoc Networks

- Special class of ad-hoc networks
- Most ad-hoc networking techniques not well suited
- Difference to ad-hoc networks
  - Number of nodes several orders of magnitude higher
  - Sensor nodes are deployed densely
  - Sensor nodes are prone to failures
  - Frequent topology changes
  - Communication mainly based on broadcast paradigm
  - Limited power, computational capabilities, and memory
  - Sensor nodes have no global identification

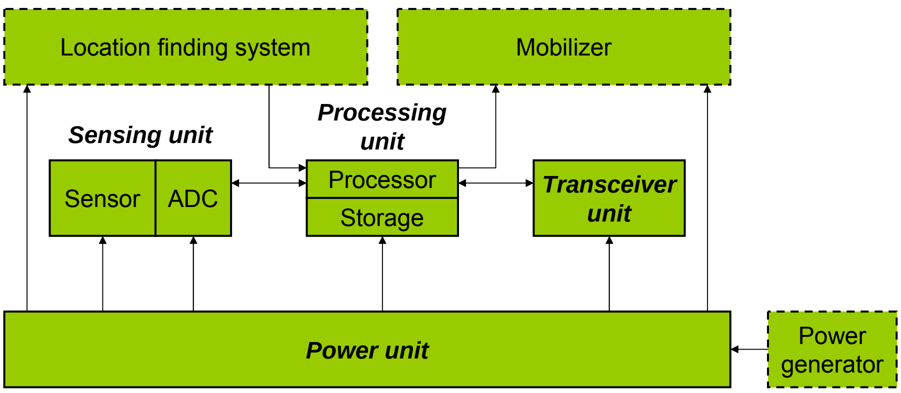
UBICOMP

Communication Architecture

A typical Sensor Network Architecture



The Components of a Sensor Node



Design Factors

- Fault tolerance
  - Sensor node may fail or be blocked
    - Lack of power
    - physical damage
    - environmental interference
  - Failure of individual node should not affect the complete network
  - (Node failure can be modeled by a Poisson process: $R(t) = e^{-\lambda t}$)
- Scalability
  - Number of nodes may reach an extreme value of millions
  - Node density may be in order of hundreds in a region
  - (Density can be calculated as: $\mu(R) = (N\pi R^2)/A$)
  - Schemes must be scalable and utilize high node density

Design Factors

- Production Cost
  - Cost of single node very important to justify cost of the network
    - Otherwise traditional tethered sensors would be the alternative
  - Sensor node should be less than 1€
    - E.g. Bluetooth 10 times more expensive than the targeted price
- Hardware constraints
  - Sensor node subunits need to fit in a matchbox-sized module
    - Required size may be smaller than a cubic centimeter
    - Light enough to remain suspended in the air
  - Additional constraints
    - Extreme energy efficient
    - Low production cost, dispensable
    - Autonomous, operate unattended, adaptive to the environment

Design Factors

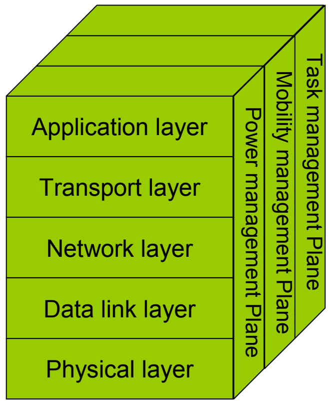
- Sensor network topology
  - Predeployment and deployment
    - Thrown in as a mass or placed one by one
  - Post-deployment
    - Topology changes due to position changes, reachability, available energy, malfunctioning, task details
  - Redeployment of additional nodes
    - Redeployment at any time to replace malfunctioning nodes or due to changes in task dynamics
- Environment
  - Home or large building, interior of a large machinery, bottom of an ocean, contaminated field, …

Design Factors

- Transmission media
  - Radio
    - Used by much of the current hardware
    - Must be available worldwide (e.g. 2.4GHz, or 916MHz)
  - Infrared, or optical media:
    - License-free, robust to interference from electrical devices, cheaper
    - Line of sight between sender and receiver
- Power consumption
  - Limited power sources: <0.5Ah, 1.2V
  - Replenishment of power source might be impossible
  - Sensor node lifetime coupled with battery lifetime
  - Power consumption in three domains: sensing, communication, and data processing

Protocol Stack

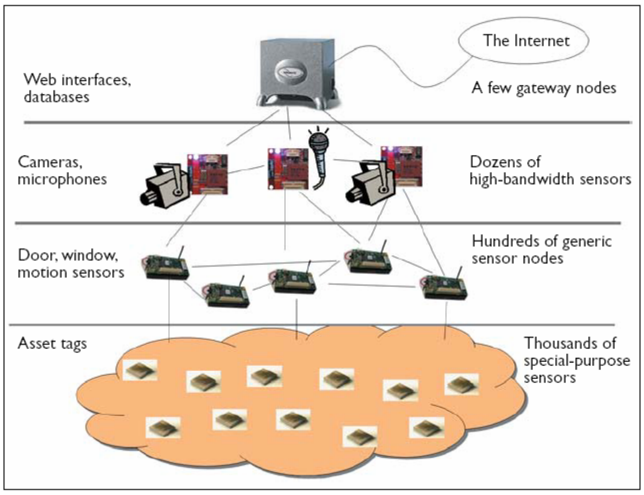
- Stack used by sink and sensors
- Management planes
  - Coordinate sensing task and lower overall power consumption
- Power management
  - E.g. turn off receiver after message receipt
  - Disconnect from routing task due to low power
- Mobility management
  - Track movement of nodes in order to maintain routes back to the user
- Task Management
  - Schedule sensing task to a specific region
  - Nodes with more power are used more frequent



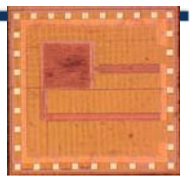
UBICOMP

Platform Classes

Sensor Node Hierarchy



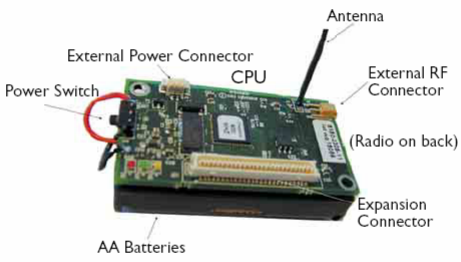
Special Purpose Sensor Nodes



- Cubic-millimeter-scale devices (see Smart Dust)
- Extremely limited energy resources
- Typical duty cycle 0.1%-0.5%
- Example scenario: track mobile assets
  - Trigger an alarm when asset leaves facility without authorization
  - Periodically report its presence for years
- Example: Spec node (Hill et al. UC Berkley)
  - Single-chip node for ultra low cost and low power consumption
  - 2.5 mm x 2.5 mm
  - Includes data RAM (< 4Kb), minimal onboard processing, and communication
  - Can interface only with simple sensors; Specialized low bandwidth sampling or advanced RF tag
  - Communicate over short distance (Bandwidth <50 kbps)
  - Current version has only transmitter (future work: transceiver)

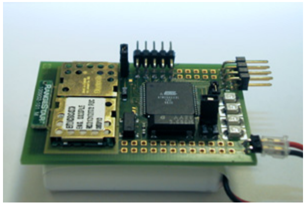
Generic Sensor Nodes

- Simple and specific function
- Require long term battery operation
- Typical duty cycle 1%-2%
- E.g. sensors placed on windows and doors for intrusion detection
- Typical operating characteristics
  - Size 1-10cm^3
  - General-purpose sensing and communications relay
  - Bandwidth <100kbps
  - Flash <0.5Mb ,RAM <10kb
- Notable example: Berkley motes → Mica2
  - Off-the-shelf components
  - Most popular sensor network research platform
  - Can be connected with a wide range of sensors
  - Can receive messages from Spec nodes
  - Processing power can easily keep track of several dozen Spec-based tags



High-Bandwidth Sensor nodes

- Handle high bandwidth of data coming from complex sensors (video, acoustic, vibration, …)
- May require battery power but often plugged into public power system for long-term operation
- Example iMote (Intel)
  - Bluetooth transceiver (~500Kbps)
  - On chip RAM ~128Kb



Gateway Sensor nodes

- End-point for mesh of sensor nodes
  - Containing database/aggregation software to process and store individual sensor readings
- Provide an interface into many existing network types
- Example platform Stargate (Intel)
  - 400 MHz X-scale architecture
  - Megabytes of RAM
  - Gigabytes of persistent storage
  - Capable to interface directly to Mica2 and iMote
  - Bridging the data to 802.11, Ethernet, …
  - Can provide a Web front-end to the sensor network

Operating Systems

- Main objective: Power management
  - Individually powered subsystems (radio, CPU, I/O, …)
  - Powered on only when in use
- TinyOS (UC, Berkeley)
  - For platforms with limited CPU power and memory (special purpose and generic sensor nodes)
- Embedded Version of Linux
  - For gateway and high-bandwidth nodes
  - Multiprocessing, preemptive task switching, virtual memory
  - Device drivers to bridge to legacy networks (Ethernet, 802.11, …)

The Need for Component-Based Architectures

- Traditional layered abstractions lead to inefficiencies in power usage
- Give applications fine-grain control over underlying hardware
  – Hardware functions exposed to applications and middleware
- TinyOS designed to allow direct access as needed
- Linux: Special-purpose drivers
  – Processor registers, general-purpose I/O lines, timing and state of peripherals
- Tradeoff: fine-grain access vs. portability
  – High-level interpreters

Platform Road Map

- Influence of Moore’s Law on device classes
  – Generic Sensor, High-bandwidth Sensor, and Gateway nodes: increase in performance (memory, communication bandwidth) for a given power and cost budget
  – Special-purpose Sensor nodes: reduce power and cost requirements while maintaining same performance
- Design of new low-power CMOS radios
  – low data rates and low power consumption
  – Specialized hardware support reduces CPU peak load
- Preferred sensor network deployment strategy (TinyOS)
  – Assemble custom protocols form building blocks
  – Start with generic protocols and customize as needed

Current Sensor Network Platforms

Node	CPU	Power	Memory	I/O and Sensors	Radio	Remarks
Special-purpose Sensor Nodes						
Spec 2003	4–8Mhz Custom 8-bit	3mW peak 3uW idle	3K RAM	I/O Pads on chip, ADC	50–100Kbps	Full custom silicon, traded RF range and accuracy for low-power operation.
Generic Sensor Nodes						
Rene 1999	ATMEL 8535	.036mW sleep 60mW active	512B RAM 8K Flash	Large expansion connector	10Kbps	Primary TinyOS development platform.
Mica-2 2001	ATMEGA 128	.036mW sleep 60mW active	4K RAM 128K Flash	Large expansion connector	76Kbps	Primary TinyOS development platform.
Telos 2004	Motorola HCS08	.001mW sleep 32mW active	4K RAM	USB and Ethernet	250Kbps	Supports IEEE 802.15.4 standard. Allows higher-layer Zigbee stardard. 1.8V operation
Mica-Z 2004	ATMEGA 128		4K RAM 128K Flash	Large expansion connector	250Kbps	Supports IEEE 802.15.4 standard. Allows higher-layer Zigbee stardard.

Current Sensor Network Platforms

Node	CPU	Power	Memory	I/O and Sensors	Radio	Remarks
High-bandwidth Sensor Nodes						
BT Node 2001	ATMEL Mega 128L 7.328Mhz	50MW idle 285MW active	128KB Flash 4KB EEPROM 4KB SRAM	8-channel 10-bit A/D, 2 UARTS Expandable connectors	Bluetooth	Easy connectivity with cell phones. Supports TinyOS. Multihop using multiple radios/nodes.
Imote 1.0 2003	ARM 7TDMI 12-48MHz	1mW idle 120mW active	64KB SRAM 512KB Flash	UART, USB, GPIO, I^{2}C, SPI	Bluetooth 1.1	Multihop using scatternets, easy connections to PDAs, phones, TinyOS 1.0, 1.1.
Gateway Nodes						
Stargate 2003	Intel PXA255		64KNSRM	2 PCMICA/CF, com ports, Ethernet, USB	Serial connection to sensor network	Flexible I/O and small form factor power management.
Inrysnc Cerfcube 2003	Intel PXA255		32KB Flash 64KB SRAM	Single CF card, general-purpose I/O		Small form factor, robust industrial support, Linux and Windows CE support.
PC104 nodes	X86 processor		32KB Flash 64KB SRAM	PCI Bus		Embedded Linux or Windows support.

UBICOMP

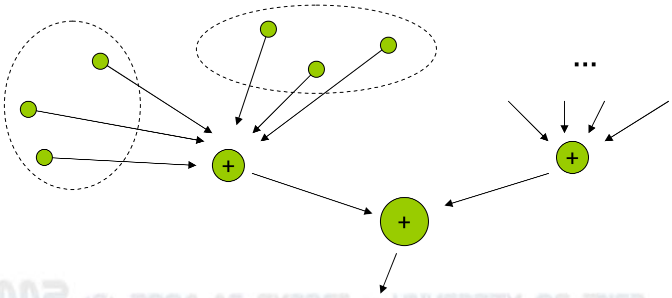
Data Centric Routing

What Means Data Centric Routing?

- Classic Communication Patterns
  - Unicast, Broadcast, Multicast
  - Addressing of individual node or set of individual nodes
- What if individual nodes disappear?
- New Data-Centric communication paradigms arise
  - Anycast, Geocast, Marketplace-Communication
- Two possible Data-Centric Routing Approaches
  - Disseminate interest about data
  - Advertise available data and wait for request
- Can be position-based and topology-based or both

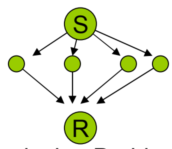
How to Address Nodes with no ID?

- Attribute-Based naming
  - Rather query an attribute than an individual node
  - E.g. the areas where the temperature is over 50°C, all available information about a running application in a certain area, …
- Data aggregation often needed to merge data received from many nodes (data fusion)
  - Some specifics may not be left out (e.g. location of the data)

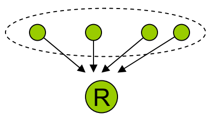


Flooding and Gossiping

- Flooding: When receiving packet for the first time, repeat forwarding, if maximum hop or destination not reached
  - Reactive technique
  - Does not require costly topology maintenance
- Deficiencies
  - Implosion
  - Overlap
  - Resource Blindness: Does not take energy resources into account
- Gossiping: forward to one random selected neighbor only
  - Avoids implosion problem
  - Message propagation takes a long time



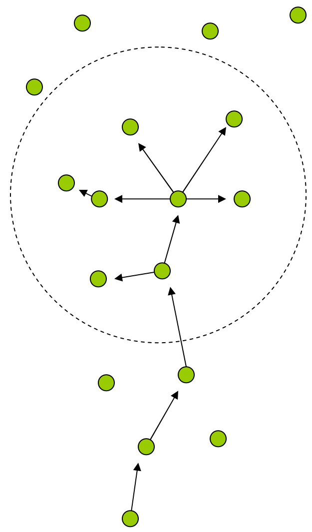
Implosion Problem



Overlap Problem

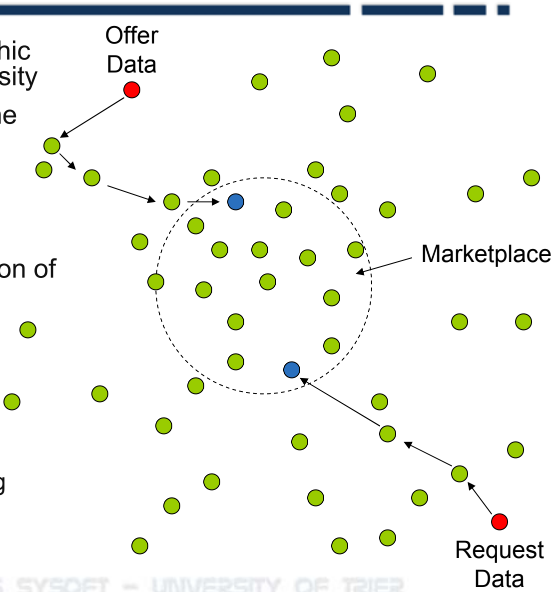
Geocasting

- Reach nodes in a certain area
- Geocasting Components
  - Routing towards the area
    - Single-path, multi-path
    - Restricted directional flooding
  - Dissemination inside the area
    - Location-aware flooding
    - Reducing redundant transmissions
- Geocast with guaranteed delivery?



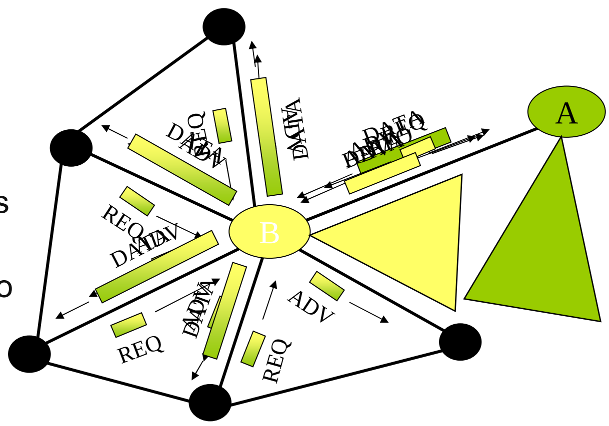
Marketplace Pattern

- Place offer in a geographic area with high node density
- Send request towards the same area
- Code execution at the marketplace to reduce message complexity
- Background dissemination of marketplace locations
- Moving towards the marketplace
  - Geographic routing
- Communication on the marketplace
  - Topology based routing
  - Efficient flooding



Sensor Protocols for Information via Negotiation (SPIN)

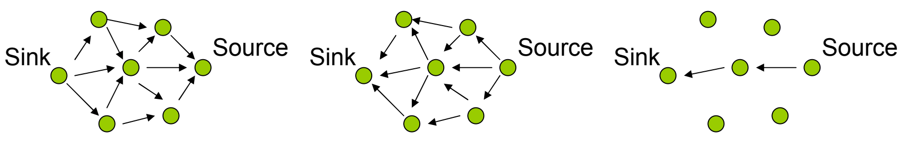
- Advertise metadata of sensed phenomenon
- Interested neighbor replies with a request
- Send full data to interested neighbors only
  - Utilizing broadcast property



© Animation from slides of Li, Huan and Liu, Junning

Directed Diffusion

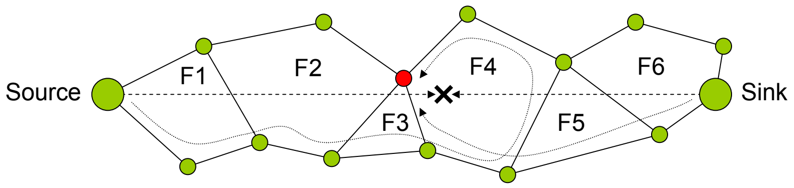
- Set up gradients for data to flow from source nodes to interested sink node
- Sink sends out interest
  - Attribute value pairs describing sensing task
- Interest Propagated through the network
  - Cached in each node to build gradients back to the sink
- Data from sources is sent back along interest gradient paths
  - Data aggregation performed locally at intermediate nodes



(a) Propagate interest (b) Set up gradient (c) Send data

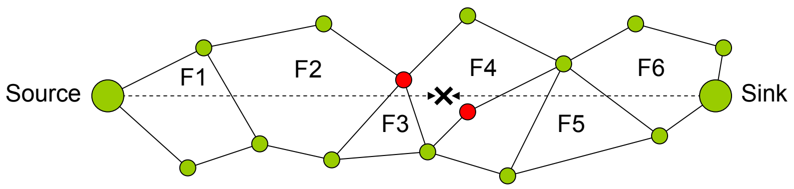
Geographic Hash Table (GHT) (1)

- Idea: hashing on geographical positions
- Put() and Get() operations map to the same device near to the hashed location
- Mapped device stores data
- Use of planar graph routing to find the same device



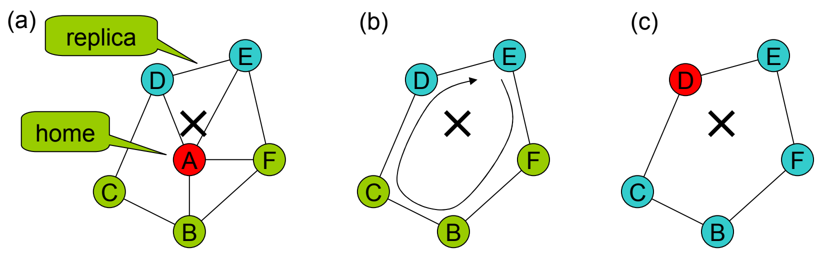
Geographic Hash Table (GHT) (2)

- Problem: changing network topology
  - Storing node might disappear
  - Put() and Get() may retrieve different storing nodes



Geographic Hash Table (GHT) (3)

- Solution
  - Replication along the face perimeter
  - Periodic refresh messages traveling along the perimeter
  - New home node selected when
    - Refresh packet is missing for a certain timeout
    - Node closer to destination receives refresh packet



UBICOMP

Security in Sensor Networks

Building a Secure System

- Traditional techniques can't be applied
  - Limited energy, computation, and communication capabilities
  - Added risk of physical attack
  - Interact closely with physical environment
- E.g. Public-key cryptography (like Diffie-Hellman)?
  - Arbitrary node pairs can set up secure key
  - Key establishment beyond sensor network capabilities
- Chance to address sensor network security from the start
  - No standalone component added to the system

Key Establishment and Trust Setup

- Network wide shared key
  - Simple solution
  - Problem: Single node may reveal the secret key
- Single shared key to establish set of link keys
  - One per pair of network nodes
  - Erase shared key afterwards
  - Problem: Does not allow addition of new nodes
- Preconfigure with unique symmetric shared key between each pair of nodes
  - Scalability? → each node stores n-1 keys (n(n-1) keys need to be established)

Key Establishment and Trust Setup

- Set up keys with others using a trusted base station
  - Each node shares a single key with the base station
  - Problem: base station is a single point of failure
- Random-key predistribution protocol
  - Large pool of symmetric keys
  - Random subset of the pool distributed to each sensor node
  - Two nodes search their pools to determine whether they share a common key
  - If existent, use key to establish a session key
  - Problem: attacker may compromise enough keys to reconstruct the complete key pool

Privacy Aspects in Sensor Networks

- Sensor technology may be used for illegal surveillance
  - Abuse of existing network
    - Node capture
  - Deployment of new networks
    - Affordable small devices
  - Data collection, coordinated analysis
    - E.g.Tracking of people and vehicles over long periods of time
  - E.g. Employers → employees, shop owners → customers, neighbors → neighbors, law enforcement agencies → public places
- Providing awareness of the presence of sensor nodes!
  - Enabled by a mix of societal norms, new laws, and technological responses

Additional Security Issues

- Robustness to communication denial of service
  - Broadcasting a high-energy signal to disrupt network's operation
  - More sophisticated: violate MAC protocol (e.g. continuously request channel access with a RTS signal)
  - Standard defense: spread-spectrum communication (cryptographically secure spread-spectrum radios not commercially available)
- Secure routing schemes needed
  - Denial-of-service attacks often possible (injecting malicious routing information)
- Resilience to node capture
  - Physical security in traditional networks
  - Sensor nodes often placed in locations easily accessible to attackers
    - Extract cryptographic secrets, modify programming, replace with malicious nodes
  - Solutions: state replication, majority voting, gather multiple redundant views before reporting an event
  - E.g. routing along multiple independent paths and checking consistency of received packets at destination node

High-Level Security Primitives

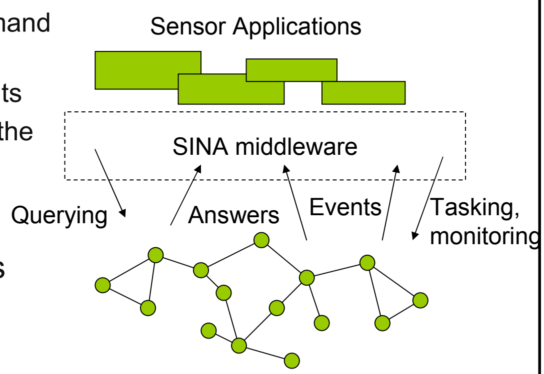
- Secure group management
  - Sensing often performed by a group of nodes (e.g. tracking a vehicle)
  - Protocols needed for securely admitting new group members, secure group communication, authentication of group's computation, …
- Intrusion detection
  - Methods from classical networks applicable?
  - Sensor networks need fully distributed and inexpensive solutions
  - Secure group management is a promising approach
- Secure data aggregation
  - E.g. randomly sampling a small fraction of nodes and checking that they behaved properly

UBICOMP

Sensor Information Networking Architecture (SINA)

SINA Middleware Concept

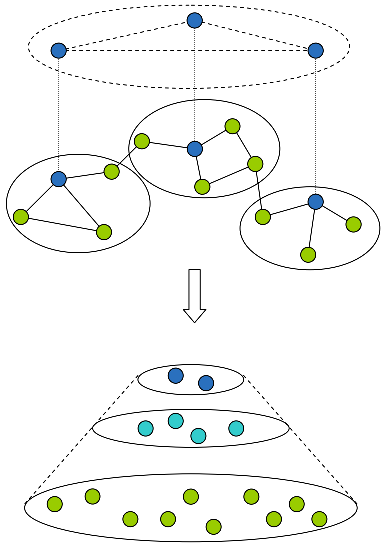
- Allows applications to
  - Issue queries and command tasks
  - Collect replies and results
  - Monitor changes within the network



- Functional components
  - Hierarchical clustering
  - Attribute-based naming
  - Location awareness

Hierarchical Clustering

- Autonomous clustering
  - Energy-efficiency
  - Scalable operations
- Form a hierarchy of clusters
- Election of cluster head
  - Performs information filtering, fusion, and aggregation
- Re-elect cluster head when current cluster head fails or runs out energy
  - Reorganize clustering structure if necessary
- Each node forms a one-level cluster when hierarchy is not applicable



Attribute-Based Naming and Location Awareness

- Sensor queries are data-centric
  - E.g. what area has a temperature above 50°C?
  - E.g. what is the average temperature in the SE quadrant?
- Attribute-based naming preferred addressing scheme
  - [type=temperature, location=N-E, temperature=50]
- Location sensing by GPS
  - Economical reason: subset of GPS equipped node function as location reference
  - Alternative: optical tracking in a small region, …
- How are the above questions performed using the three functional components?

Information Abstraction

- Network node viewed as data sheet
- Cells referred via unique attribute-based names
  - Initially small number of predefined cells
  - New cells can be created to
    - Obtain information from other cells, invoke system-defined functions, or aggregate information from other nodes
- Cell content can either be
  - Single value (e.g. remaining battery power)
  - Multiple values (e.g. history of temperature changes in the past 30 min)

Sensor Query and Tasking Language (SQTL)

- Interface between applications and middleware
- Procedural scripting language
  - Hardware access: getTemperature, turnOn, …
  - Location awareness: isNeighbor, getPosition, …
  - Communication: tell, execute, …
  - Processing of asynchronous events
    - Message receipt: receive
    - Periodical timer: every
    - Time expiration: expire
- SQTL messages
  - Executed by any node in the network
- SQTL wrapper in order to target specific nodes

Arguments of a SQTL wrapper

Argument	Meaning
sender	The sender of an SQTL message wrapper
receiver group criteria	Potential receivers specify by two subarguments: Subargument of receiver to specify group of receiver; its possible value can be one of ALL_NODES, or NEIGHBORS Subargument of receiver to specify selection criteria of receivers
application-id	A unique ID for each application in the same sensor network
num-hop	Number of hops away from a gateway node
language	Specify a language used in content
content	A payload containing a program, a message, or return values
with (optional) parameter type name value	Tuples of parameters used in the program passed from sender to receiver Repeatable subargument of with Data type of the parameter Name of the parameter Value of the parameter

Sensor Execution Environment (SEE)

- Message dispatching
  - Inspect “receiver” argument of SQTL
    - ALL_NODES: rebroadcast to every node
    - NEIGHBORS: send to one-hop neighbors only
  - Messages with matching criteria accepted only (late binding)
- “tell” message used to deliver result back to front-end node
  - Using upstream node from where the script came
- Message forwarding
  - Apply translation to unique link-layer address whenever possible
  - Use broadcast otherwise

Built-In Declarative Query Language

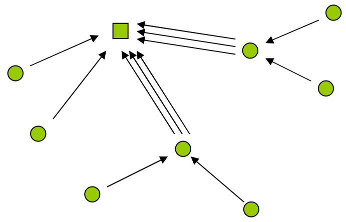
- SQL like alternative to explicitly writing procedural SQTL code
- Example: ask every cluster head to create a new attribute cell “avgTemperature” containing average temperature over all cluster members:

  SELECT avg(getTemperature())
  AS avgTemperature
  FROM CLUSTER-MEMBERS

- Technique referred as device databases

Information Gathering Methods

- Information gathering primitives
  - Sampling Operation
  - Self-Orchestration
  - Diffused Computation Operation
- Maximize quality of responses
- Minimize network resource consumption
- Avoiding response implosion problem



Sampling operation

- Introduce response probability
- Enhancement: Cluster head determines response probability depending on node density

```
SamplingOperation(ENBC Expected Number of Responses per Cluster)
  Rebroadcast this message to all neighbors;
  prevNode ← message sender node;
  if this is a cluster head then
    prob ← ENBC/# of children;
    construct a script requesting all cluster members to return value with
    probability prob;
  end if
  Upon receiving a return message, relay it back to prevNode;
```


Self-Orchestrated Operation

- Avoid collision by deferring response
  - randomly
  - Depending on distance to destination

```
SelfOrchestratedOperation(replyProb, kh)
  rebroadcast this message to all neighbors;
  prevNode ← message sender node;
  if uniformrandom(0, 1) < replyProb then
    numHops ← number of hops this message traversed;
    delay ← kh × [numHops^2 - (2 × numHops- 1) × uniform random(0,1)];
    wait for delay;
    read power level and position, then send them back via prevNode;
  end if
  Upon receiving a return message, relay it back to prevNode;
```

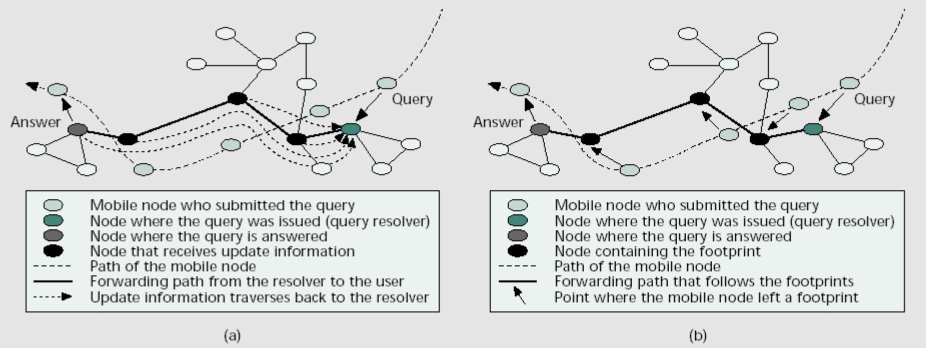
Diffused Computation

- Information gathering computed by SQTL scripts
- Collect data from child nodes
- Aggregate en route to the front end node

```
DiffusedComputation(timeout)
  confirmCount ← 0;
  prevNode ← message sender node;
  send a confirm to prevNode;
  rebroadcast this message to all neighbors;
  set timer for timeout period;
  while not timeout do
    if receive a message of type confirm then
      confirmCount ← confirmCount + 1;
    end if
  end while
  answerList ← getPowerLevel();
  while confirmCount ≠ 0 do
    if receive a message of type return then
      insert the returned value into answerList;
      confirmCount ← confirmCount - 1;
    end if
  end while
  return answerList back to prevNode;
```

Supporting a Mobile User

- (a) constantly update current location with resolver: increases traffic load
- (b) progressive footprint chaining: only inform nearby sensors about current location



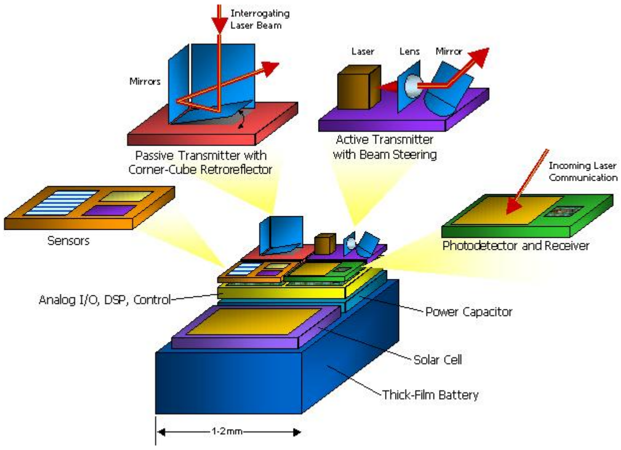
UBICOMP

Smart Dust

Berkeley's Smart Dust Project

- Explore the limits on size and power consumption in autonomous sensor nodes
  - Sensing, communication, computation, and power supply within a cubic millimeter
  - Could be small enough to remain suspended in the air
  - Last for days
- Networking and application challenges
  - Nodes must consume extremely low power
  - Communication at bit rates of kilobits
  - Need to operate at high volumetric densities

Smart Dust Motes



Major Challenge: Energy Consumption

- Power consumption limited to microwatt levels
  - Millimeter sized thick film battery stores energy in the order of 1 Joule
  - Continuous energy consumption over one day may not exceed roughly 10 microwatts
- Power management strategies needed
- Energy scavenging whenever possible
  - Solar cell and sun light: 1 Joule per day
  - Solar cell and room light: 1 millijoule per day
- Sensing and processing can be achieved at low power
- Ultra-low-power communication represents a critical challenge

Dust Mote Communication: Motivation

- Candidate communication technologies
  - Radio frequency (RF)
  - Optical transmission techniques
- RF Communication
  - Limited space for antennas → extremely short wavelength
  - Short-wavelength communication needs a lot of power
  - Radio transceivers are relatively complex circuits
    - Modulation, bandpass filtering, demodulation circuitry
    - Transmission multiplexing of multiple dust motes: time-, frequency-, code-division multiple access
- Motes are thus based on an optical communication technique

Dust Mote Communication: Optical Transmission

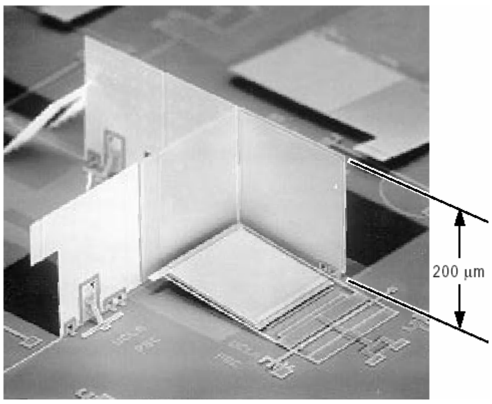
- Free-space optical transmission
  - Requires significantly lower energy than RF communication
  - Line-of-sight!
- Reasons for power advantage
  - Simple baseband analog and digital circuitry
  - No modulators, demodulators, and active baseband filters
- Short wavelength of visible or near-infrared light can be emitted by millimeter scale device
- Space-division multiplexing
  - Base-station transceiver (BTS) with image receiver can decode simultaneous transmissions at different locations

Dust Mote Communication: Optical Transmission

- Dust motes may block line-of-sight
  - Unlikely for small mote sizes
- Different dust motes must be received on different pixels at the BTS
  - E.g. covering an 17x17 meter area with 256x256 pixel camera → motes separated with a 6.6 centimeter square
- Possible communication
  - Active: laser diode and beam steering
    - Peer-to-peer communication between dust motes
    - Power consumption: long ranges (kilometers) at low data rates, high bit rates (megabits per second) over shorter distances
    - Power consumption of semiconductor laser in the order of 1 milliwatt
      - Short-duration burst-mode communication
    - Protocol to aim beam towards receiving parties needed
  - Passive: corner-cube retroreflector
    - No optical power supply needed

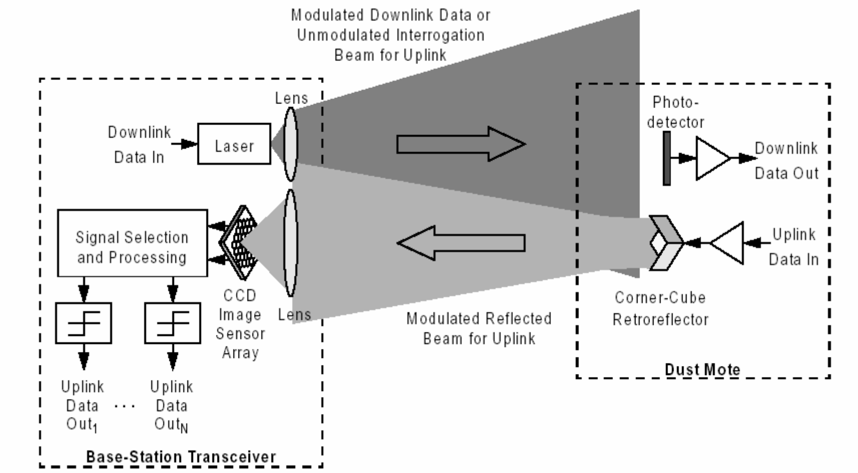
Corner Cube Retroreflector (CCR)

- Incident ray of light reflected back to source
  - Provided ray is within a certain angle to cube body diagonal
- Misaligned mirror leads to an interrupted ray
- Electrostatic actuator deflects one of the mirror at kilohertz rates
- Communication
  - 1 kilobit/sec
  - 150 meters
  - 5-milliwatt illuminating laser
- Inherently directional
  - Important implications on routing strategies
  - Apply several CCRs on a mote



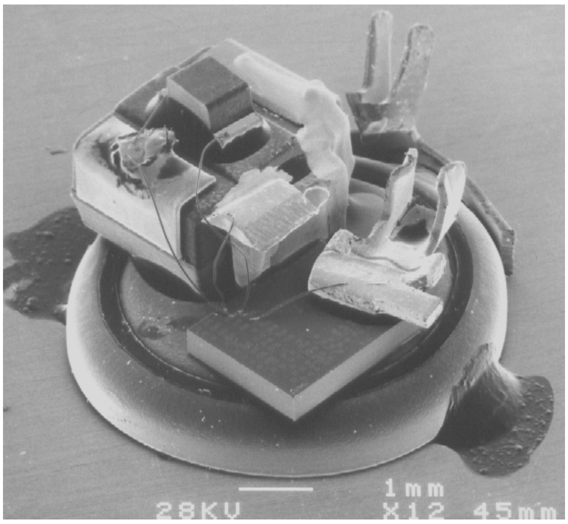
UBICOMP (C) 2004 AG SYSOFT – UNIVERSITY OF TRIER

Communication with the Base Station



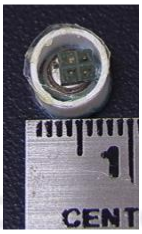
Shrinking Mote Size: Flashy Dust

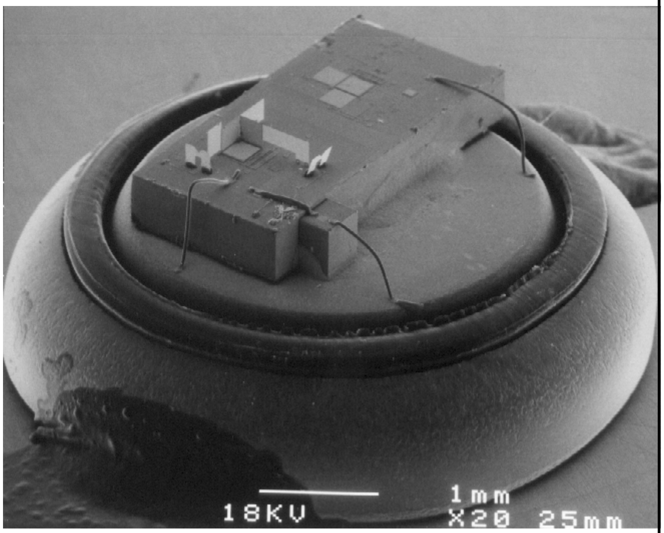
- 138 mm^3 uni-directional communication and sensing (ambient light) mote



Shrinking Mote Size: Daft Dust

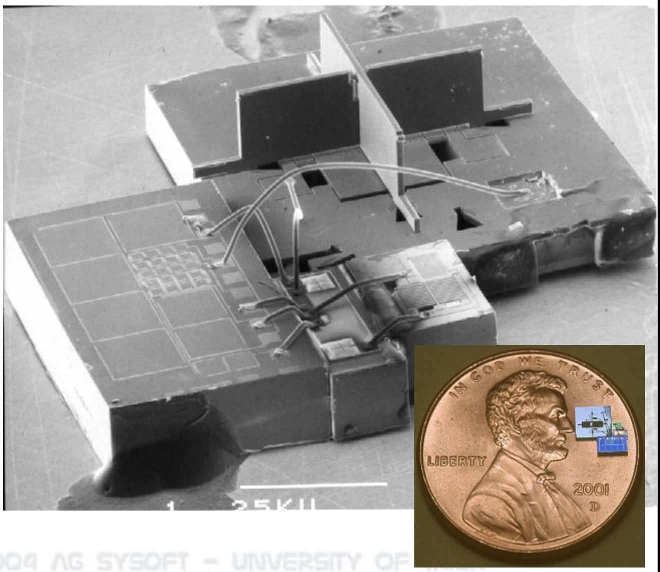
- 63 mm^3 bi-directional communication mote
- four CCR's for better hemispherical coverage





Shrinking Mote Size: Golem Dust

- solar powered mote with bi-directional communications and sensing (acceleration and ambient light)
- 11.7 mm^3 total circumscribed volume
- ~4.8 mm^3 total displaced volume



Shrinking Mote Size: Clever Dust

- Ultra-low energy microcontroller developed for Smart Dust that consumes an average of 12pJ/instruction in 0.25µm CMOS.