Sensor Networks & Applications

Partly based on the book Wireless Sensor Networks by Zhao and Guibas

Constraints and Challenges

- Limited hardware:
  - Storage
  - Processing
  - Communication
  - Energy supply (battery power)
- Limited support for networking:
  - Peer-to-peer network
  - Unreliable communication
  - Dynamically changing
- Limited support for software development:
  - Real-time tasks that involve dynamic collaboration among nodes
  - Software architecture needs to be co-designed with the information processing architecture

Sensor vs other ad hoc networks

Sensor networks

- Special communication patterns: many-to-one, one-to-many, attribute-based
- Static sensors in many applications
- Constraints more severe than general ad hoc networks
- Distributed collaborative computing

Mobile ad hoc networks

- General-purpose communication involving mobile devices
- Devices are often mobile

Applications

- Environmental monitoring
  - Traffic, habitat, security
- Industrial sensing and diagnostics
  - Manufacturing, supply chains
- Context-aware computing
  - Intelligent homes
- Military applications:
  - Multi-target tracking
- Infrastructure protection:
  - Power grids

Why are Sensor Networks Special?

- Matchbox-sized to Shoebox-sized nodes
  - Tmote: 8 MHz, 10K RAM 48K Flash, 15 kJ, 50 m
  - Sensoria sensor: 400 MHz, 32 MB, 300 kJ, 100 m
- More severe power constraints than PDAs, mobile phones, laptops
- Mobility may be limited, but failure rate higher
- Usually under one administrative control
- A sensor network gathers and processes specific kinds of data relevant to application
- Potentially large-scale networks comprising of thousands of tiny sensor nodes

Advantages of networked sensing

- Detection:
  - Improved signal-to-noise ratio by reducing average distance between source and sensor
- Energy:
  - A path with many short hops has less energy consumption than a path with few long hops
- Robustness:
  - Address individual sensor node or link failures

Localization techniques

- Methods that allow the nodes in a network to determine their geographic positions
- Use of current GPS systems not feasible:
  - Cost
  - Power consumption
  - Form factor
  - Do not work indoors or under dense foliage
- Sensor network approach:
  - A small number of nodes are equipped with GPS receivers and can localize themselves
  - Other nodes localize themselves using landmarks
- Two ranging techniques:
  - Received signal strength (RSS)
  - Time of arrival (TOA/TDOA)

Received signal strength (RSS)

- If the source signal strength and attenuation are known, then receiver can estimate its distance from the sender

$$P_r = G \cdot P_t / d^{\alpha}$$

- Unfortunately, RSS (α) can vary substantially:
  - Fading
  - Multipath effects
- Apparently, localization to within meters is the best one can do with RSS in practice

Time of arrival

- Basic idea:
  - Measure time it takes for a signal to travel from sender to receiver
  - Multiply by signal propagation speed
- Need sender and receiver to be synchronized, and exact time of transmission known
- Exact transmission time hard to determine
- Time Difference of Arrival (TDOA)
  - Measure TDOA at two receivers
  - Can obtain difference in distances between the two receivers and the sender
- Signal speed not necessarily constant
  - Local beacons and measurements to estimate signal speed
- Apparently distance measurement to within centimeters achievable

Localization using ranging

- Obtain multiple distance measurements using multiple landmarks
- Write out equations, including measurement errors
  - Variables are the errors and the location coordinates
- Minimize the weighted total squared error to yield the desired estimate
- [SHS01]

Focus Problems

- Medium-access and power control:
  - Power saving techniques integral to most sensor networks
  - Possibility of greater coordination among sensor nodes to manage channel access
- Routing protocols:
  - Geographic routing and localization
  - Attribute-based routing
  - Energy-Awareness
- Synchronization protocols:
  - Many MAC and application level protocols rely on synchronization
- Query and stream processing:
  - Sensor network as a database
  - Streams of data being generated at the nodes by their sensors
  - Need effective in-network processing and networking support

MAC Protocols for Sensor Networks

- Contention-Based:
  - CSMA protocols (IEEE 802.15.4)
  - Random access to avoid collisions
  - IEEE 802.11 type with power saving methods
- Scheduling-Based:
  - Assign transmission schedules (sleep/awake patterns) to each node
  - Variants of TDMA
- Hybrid schemes

IEEE 802.15.3

- Two versions:
  - Basic (without beacons)
  - Beacon-based
- Basic: CSMA-CA
- Beacon-based
  - Similar to PCF of 802.11
  - Coordinator sends out a beacon periodically
  - Superframe between consecutive beacons
    - Active period
    - Inactive period
  - Active period
    - Contention Access Period
    - Contention Free Period

Proposed MAC Protocols

- PAMAS [SR98]:
  - Power-aware Medium-Access Protocol with Signaling
  - Contention-based access
  - Powers off nodes that are not receiving or forwarding packets
  - Uses a separate signaling channel
- S-MAC [YHE02]:
  - Contention-based access
- TRAMA [ROGLA03]:
  - Schedule- and contention-based access
- Wave scheduling [TYD+04]:
  - Schedule- and contention-based access

S-MAC

- Identifies sources of energy waste [YHE03]:
  - Collision
  - Overhearing
  - Overhead due to control traffic
  - Idle listening
- Trade off latency and fairness for reducing energy consumption
- Components of S-MAC:
  - A periodic sleep and listen pattern for each node
  - Collision and overhearing avoidance

S-MAC: Sleep and Listen Schedules

- Each node has a sleep and listen schedule and maintains a table of schedules of neighboring nodes
- Before selecting a schedule, node listens for a period of time:
  - If it hears a schedule broadcast, then it adopts that schedule and rebroadcasts it after a random delay
  - Otherwise, it selects a schedule and broadcasts it
- If a node receives a different schedule after selecting its schedule, it adopts both schedules
- Need significant degree of synchronization

S-MAC: Collision and Overhearing Avoidance

- Collision avoidance:
  - Within a listen phase, senders contending to send messages to same receiver use 802.11
- Overhearing avoidance:
  - When a node hears an RTS or CTS packet, then it goes to sleep
  - All neighbors of a sender and the receiver sleep until the current transmission is over

TRAMA

- Traffic-adaptive medium access protocol [ROGLA03]
- Nodes synchronize with one another
  - Need tight synchronization
- For each time slot, each node computes an MD5 hash, that computes its priority

$$p(u,t) = MD5(u \oplus t)$$

- Each node is aware of its 2-hop neighborhood
- With this information, each node can compute the slots it has the highest priority within its 2-hop neighborhood

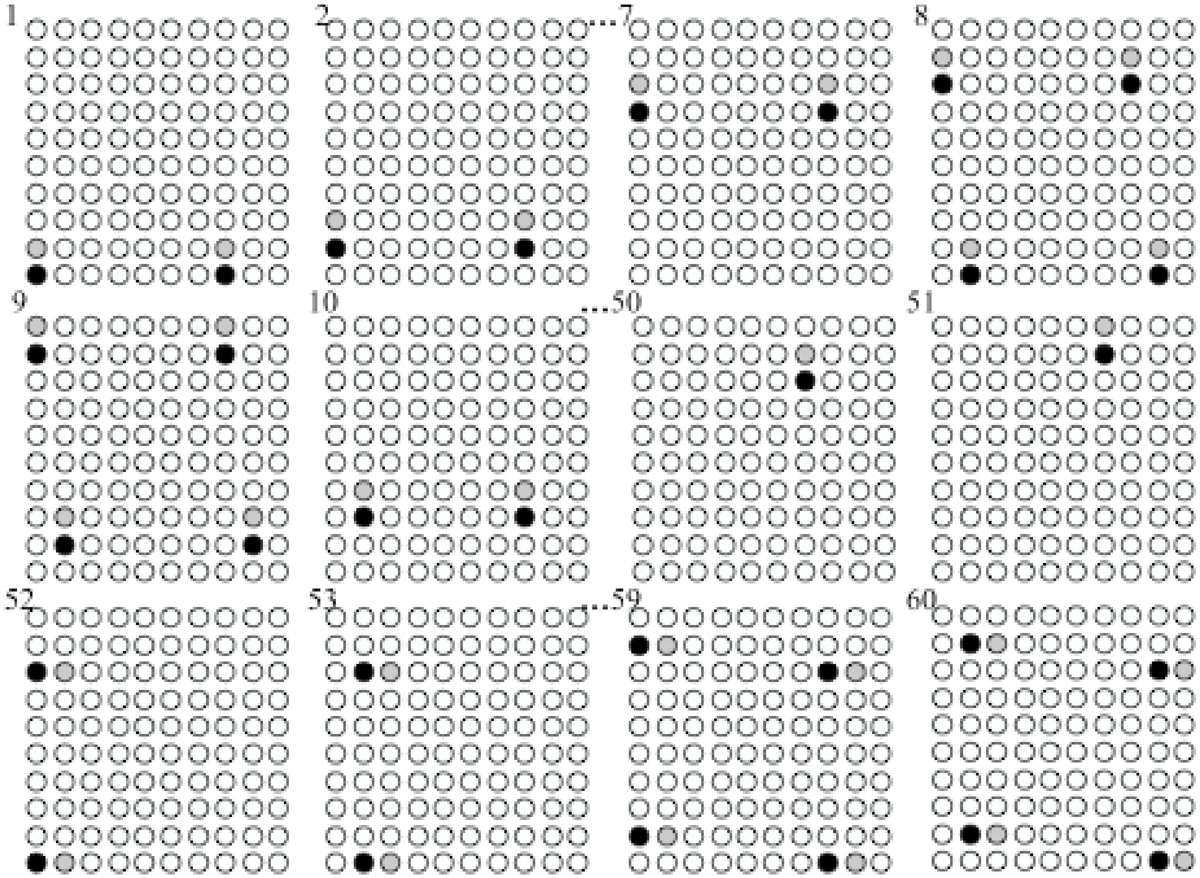
TRAMA: Medium Access

- Alternates between random and scheduled access
- Random access:
  - Nodes transmit by selecting a slot randomly
  - Nodes can only join during random access periods
- Scheduled access:
  - Each node computes a schedule of slots (and intended receivers) in which will transmit
  - This schedule is broadcast to neighbors
  - A free slot can be taken over by a node that needs extra slots to transmit, based on priority in that slot
  - Each node can determine which slots it needs to stay awake for reception

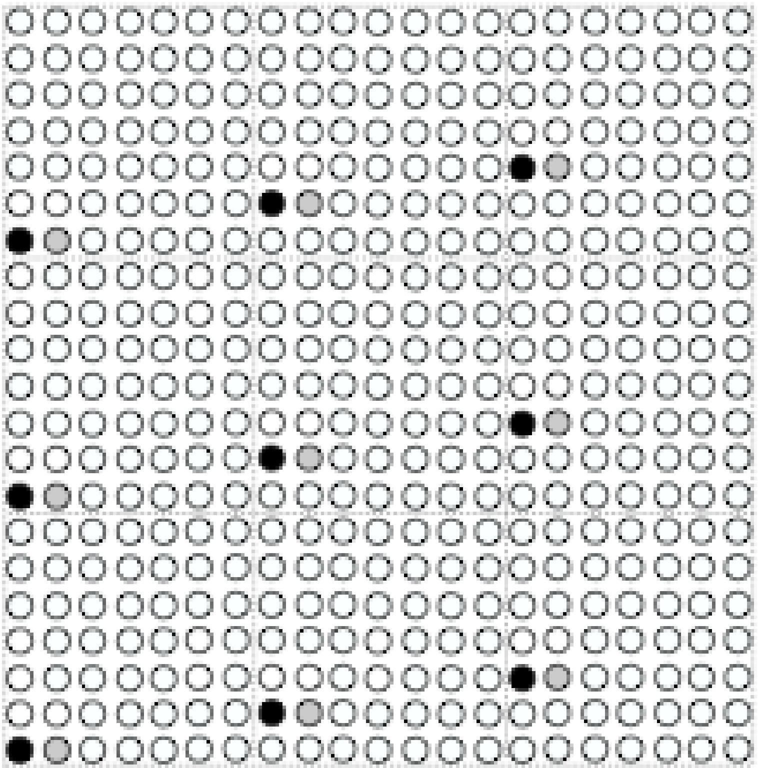
Wave Scheduling

- Motivation:
  - Trade off latency for reduced energy consumption
  - Focus on static scenarios
- In S-MAC and TRAMA, nodes exchange local schedules
- Instead, adopt a global schedule in which data flows along horizontal and vertical “waves”
- Idea:
  - Organize the nodes according to a grid
  - Within each cell, run a leader election algorithm to periodically elect a representative (e.g., GAF [XHE01])
  - Schedule leaders’ wakeup times according to positions in the grid

Wave Scheduling: A Simple Wave



Wave Scheduling: A Pipelined Wave



Wave Scheduling: Message Delivery

- When an edge is scheduled:
  - Both sender and receiver are awake
  - Sender sends messages for the duration of the awake phase
  - If sender has no messages to send, it sends an NTS message (Nothing-To-Send), and both nodes revert to sleep mode
- Given the global schedule, route selection is easy
  - Depends on optimization measure of interest
  - Minimizing total energy consumption requires use of shortest paths
  - Minimizing latency requires a (slightly) more complex shortest-paths calculation

Routing strategies

- Geographic routing
  - Greedy routing
  - Perimeter or face routing
  - Geographic localization
- Attribute-based routing:
  - Directed diffusion
  - Rumor routing
  - Geographic hash tables
- Energy-aware routing:
  - Minimum-energy broadcast
  - Energy-aware routing to a region

Geographic Location Service (GLS)

- Use sensor nodes as location servers
- Organize the space as a hierarchical grid according to a spatial quad-tree
- Each node has a unique ID (e.g., MAC address)
- Location servers for a node X:
  - One server per every vertex of the quad tree that is a sibling of a vertex that contains X
  - This server is the node with smallest ID larger than X in the region represented by the quad tree vertex (with wraparound, if necessary)
- To locate a node X:
  - Traverse up the quad tree repeatedly seeking the node that has the smallest ID larger than X
- [L+00]

Performance of GLS

- Depth of the quad-tree = O(log n), where n is the number of nodes in the network
- Therefore, number of location servers for a given node is O(log n)
- How many nodes does a given node serve?
  - If the Ids are randomly distributed, can argue that the expected number of nodes served per node is O(log n)
  - Even with high probability
- If the source and destination lie in a common quad-tree tile at level i
  - At most i steps are needed to reach the location server for the destination
  - Cost of location service distance-sensitive

Attribute-based routing

- Data-centric approach:
  - Not interested in routing to a particular node or a particular location
  - Nodes desiring some information need to find nodes that have that information
- Attribute-value event record, and associated query

type	animal
instance	horse
location	35,57
time	1:07:13

type	animal
instance	horse
location	0,100,100,200

Directed diffusion

- Sinks: nodes requesting information
- Sources: nodes generating information
- Interests: records indicating
  - A desire for certain types of information
  - Frequency with which information desired
- Key assumption:
  - Persistence of interests
- Approach:
  - Learn good paths between sources and sinks
  - Amortize the cost of finding the paths over period of use
- [IGE00]

Diffusion of interests and gradients

- Interests diffuse from the sinks through the sensor network
- Nodes track unexpired interests
- Each node maintains an interest cache
- Each cache entry has a gradient
  - Derived from the frequency with which a sink requests repeated data about an interest
  - Sink can modify gradients (increase or decrease) depending on response from neighbors

Rumor routing

- Spread information from source and query from sink until they meet
- Source information and query both follow a one-dimensional strategy
  - Random walk
  - Straight ray emanating from origin
- As agents move, both data and interest are stored at intermediate nodes
- Query answered at intersection point of these two trajectories
- Multiple sources and sinks:
  - Merge interest requests and data
- [BE02]

Geographic hash tables (GHT)

- Hash attributes to specific geographic locations in the network
  - Information records satisfying the attributes stored at nodes near location
- Every node is aware of the hash function
  - Query about records satisfying a given attribute are routed to the relevant location
- Load balancing achieved by hash function
- Information also stored locally where it is generated
- Can also provide replication through a hierarchical scheme similar to GLS
- [R+03]

GHT & geographic routing

- The nodes associated with a particular attribute are the ones that form the perimeter around the hashed location
- One of these is selected as a home node
  - Node closest to the location
  - Determined by going through the cycle
  - Recomputed periodically to allow for changes

Energy-aware routing

- Need energy-efficient paths
- Notions of energy-efficiency:
  - Select path with smallest energy consumption
  - Select paths so that network lifetime is maximized
    - When network gets disconnected
    - When one node dies
    - When area being sensed is not covered any more
- Approaches:
  - Combine geographic routing with energy-awareness
  - Minimum-energy broadcast

Geography and energy-awareness

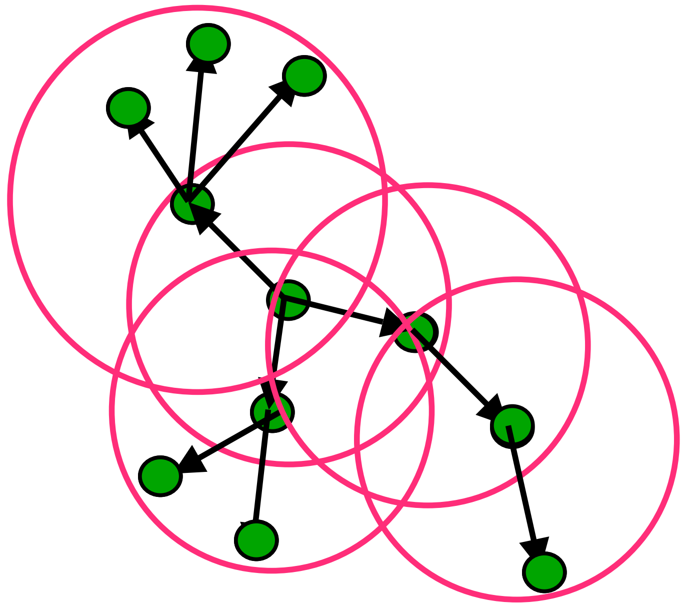
- Recall that geographic routing is divided into two steps:
  - Planarization of the underlying transmission graph
  - Routing on the planar graph using greedy and perimeter routing
- Can we obtain planar subgraphs that contain energy-efficient paths?
  - Gabriel graph and similar variants do not suffice
  - Planar subgraphs based on Delaunay triangulation have desired properties
  - Unfortunately, not necessary that greedy and perimeter routing will find such paths
- Energy available at nodes dynamically changes as different paths are used

Incorporating residual node energy

- Cost for each edge has two components:
  - Energy consumption for transmission
  - Energy already consumed at each endpoint
  - A suitable weighted combination of both
- Geographic Energy-Aware Routing (GEAR)
  - If neighbors exist that are closer with respect to both distance and cost, select such a node that has smallest cost
  - Otherwise, select a node that has smallest cost
  - Costs updated as energy of nodes change
- [YGE01]

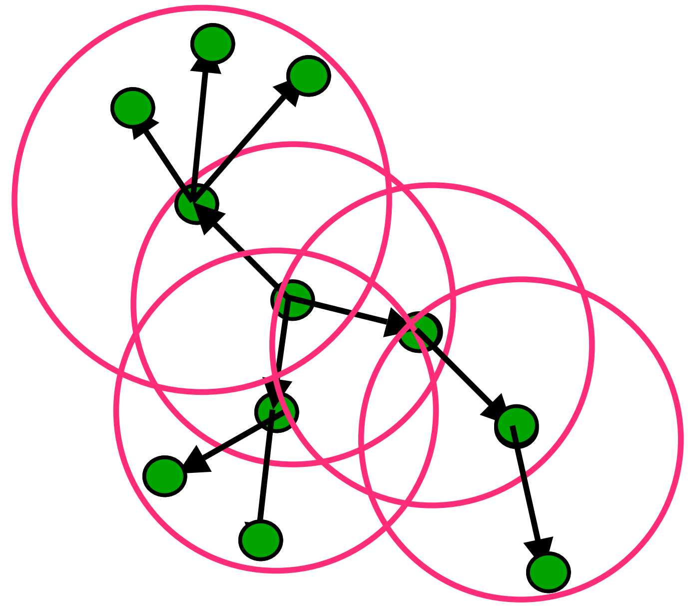
Minimum Energy Broadcast Routing

- Given a set of nodes in the plane
- Goal: Broadcast from a source to all nodes
- In a single step, a node may broadcast within a range by appropriately adjusting transmit power



Minimum Energy Broadcast Routing

- Energy consumed by a broadcast over range r is proportional to r^{α}
- Problem: Compute the sequence of broadcast steps that consume minimum total energy
- Centralized solutions
- NP-complete [ZHE02]



Three Greedy Heuristics

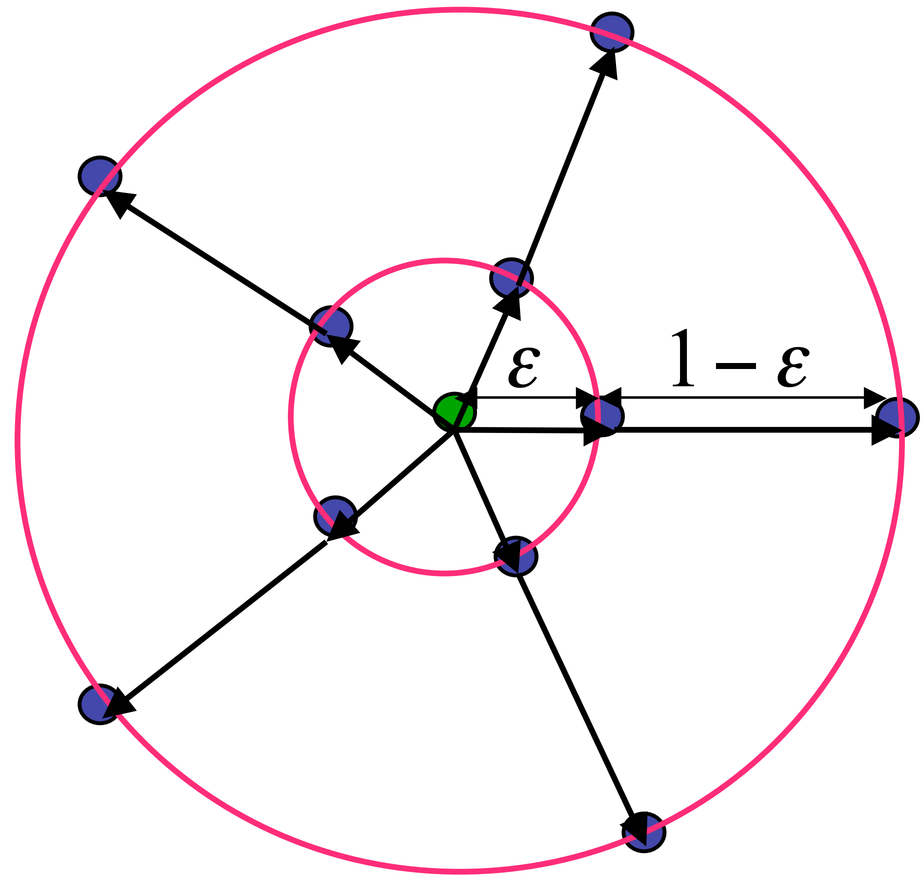
- In each tree, power for each node proportional to α th exponent of distance to farthest child in tree
- Shortest Paths Tree (SPT) [WNE00]
- Minimum Spanning Tree (MST) [WNE00]
- Broadcasting Incremental Power (BIP) [WNE00]
  - "Node" version of Dijkstra's SPT algorithm
  - Maintains an arborescence rooted at source
  - In each step, add a node that can be reached with minimum increment in total cost
- SPT is $\Omega(n)$-approximate, MST and BIP have approximation ratio of at most 12 [WCLF01]

Lower Bound on SPT

- Assume $(n-1)/2$ nodes per ring
- Total energy of SPT:

  $(n-1)(\varepsilon^{\alpha} + (1-\varepsilon)^{\alpha})/2$
- Optimal solution:
  - Broadcast to all nodes
  - Cost 1
- Approximation ratio

  $\Omega(n)$

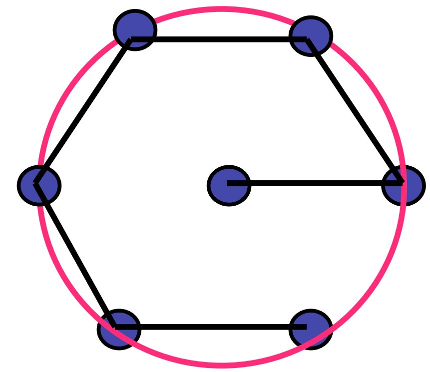


Performance of the MST Heuristic

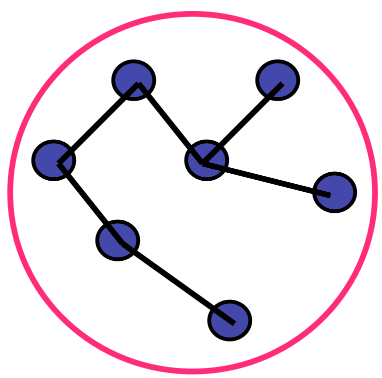
- Weight of an edge (u,v) equals $d(u,v)^{\alpha}$
- MST for these weights same as Euclidean MST
  - Weight is an increasing function of distance
  - Follows from correctness of Prim's algorithm
- Upper bound on total MST weight
- Lower bound on optimal broadcast tree

Weight of Euclidean MST

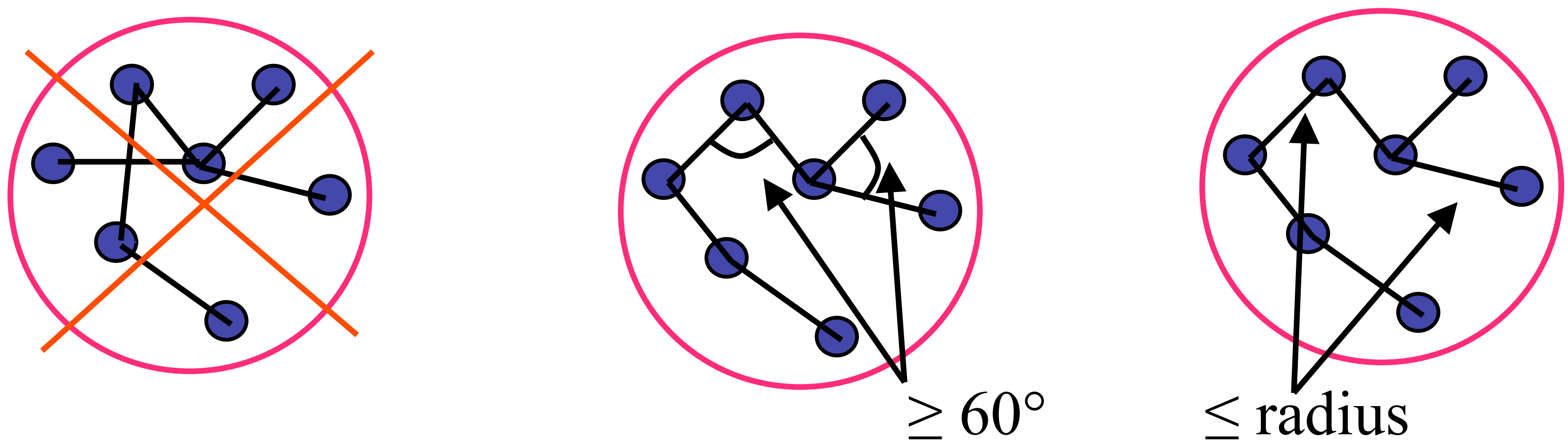
- What is the best upper bound on the weight of an MST of points located in a unit disk?
  - In [6,12]!

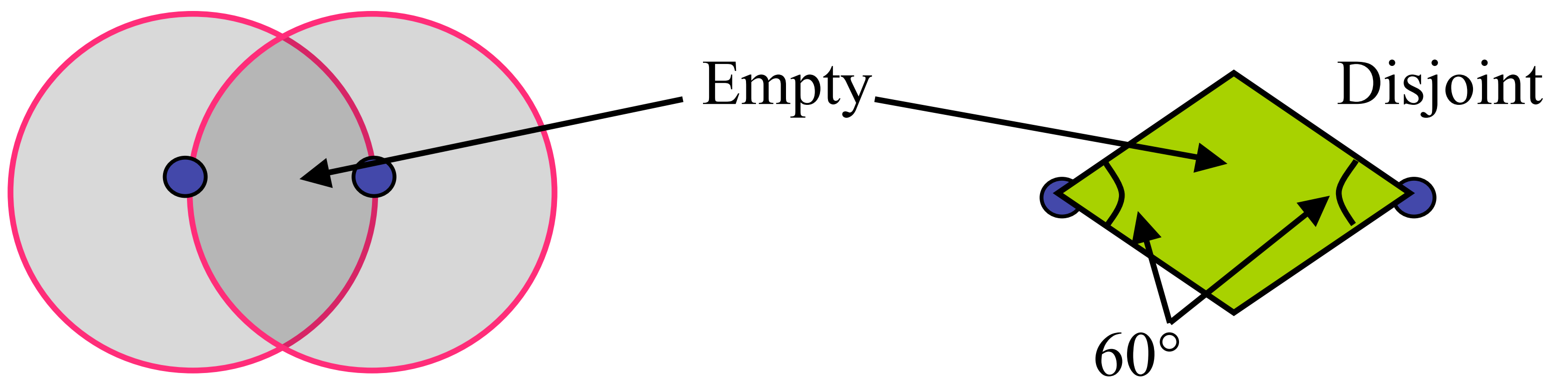


- Dependence on α
  - $\alpha < 2$: in the limit ∞
  - $\alpha \geq 2$: bounded



Structural Properties of MST

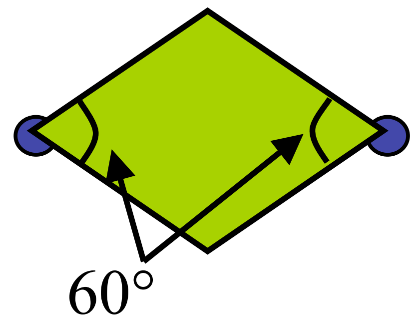




Upper Bound on Weight of MST

- Assume α = 2
- For each edge e, its diamond accounts for an area of

$$\frac{|e|^2}{2\sqrt{3}}$$



- Total area accounted for is at most $\pi(2/\sqrt{3})^2 = 4\pi/3$
- MST cost equals $\sum_e |e|^2$

$$\sum_e \frac{|e|^2}{2\sqrt{3}} \le \frac{4\pi}{3}$$

- Claim also applies for $\alpha > 2$

$$\sum_e |e|^2 \le \frac{8\pi}{\sqrt{3}} \approx 14.51$$

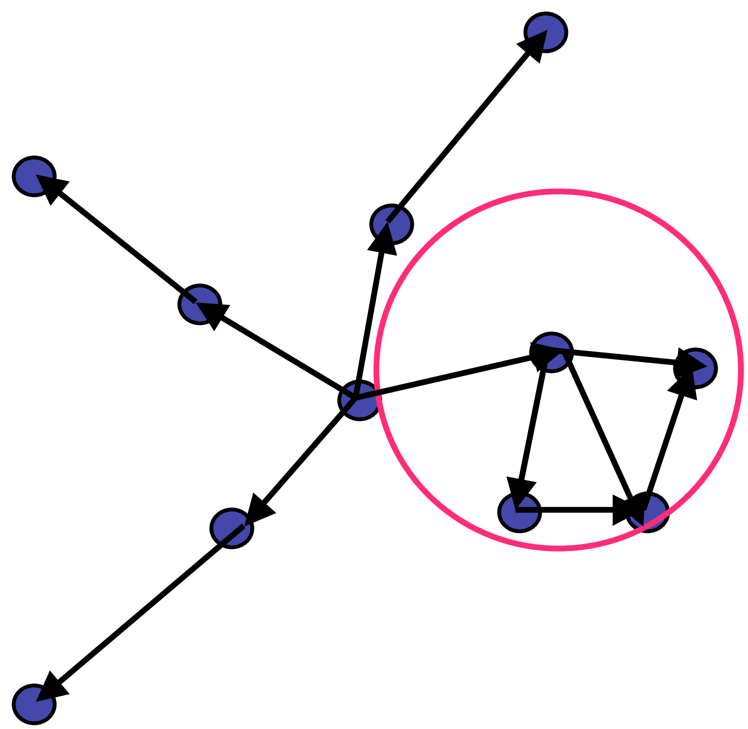
Lower Bound on Optimal

- For a non-leaf node u, let r_u denote the distance to farthest child
- Total cost is

$$\sum_u r_u^\alpha$$

- Replace each star by an MST of the points
- Cost of resultant graph at most

$$12\sum_u r_u^\alpha$$



MST has cost at most 12 times optimal

Performance of the BIP Heuristic

- Let $v_1, v_2, \ldots, v_n$ be the nodes added in order by BIP
- Let H be the complete graph over the same nodes with the following weights:
  - Weight of edge (v_{i-1}, v_i) equals incremental power needed to connect v_i
  - Weight of remaining edges same as in original graph G
- MST of H same as BIP tree B

$$
\begin{aligned}
\mathrm{Cost}_{\mathrm{G}}(B) &= \mathrm{Cost}_H(B) \\
&\le \mathrm{Cost}_H(T) \\
&\le \mathrm{Cost}_G(T)
\end{aligned}
$$

Synchronization in Sensor Networks

Need for Synchronization in Sensor Networks

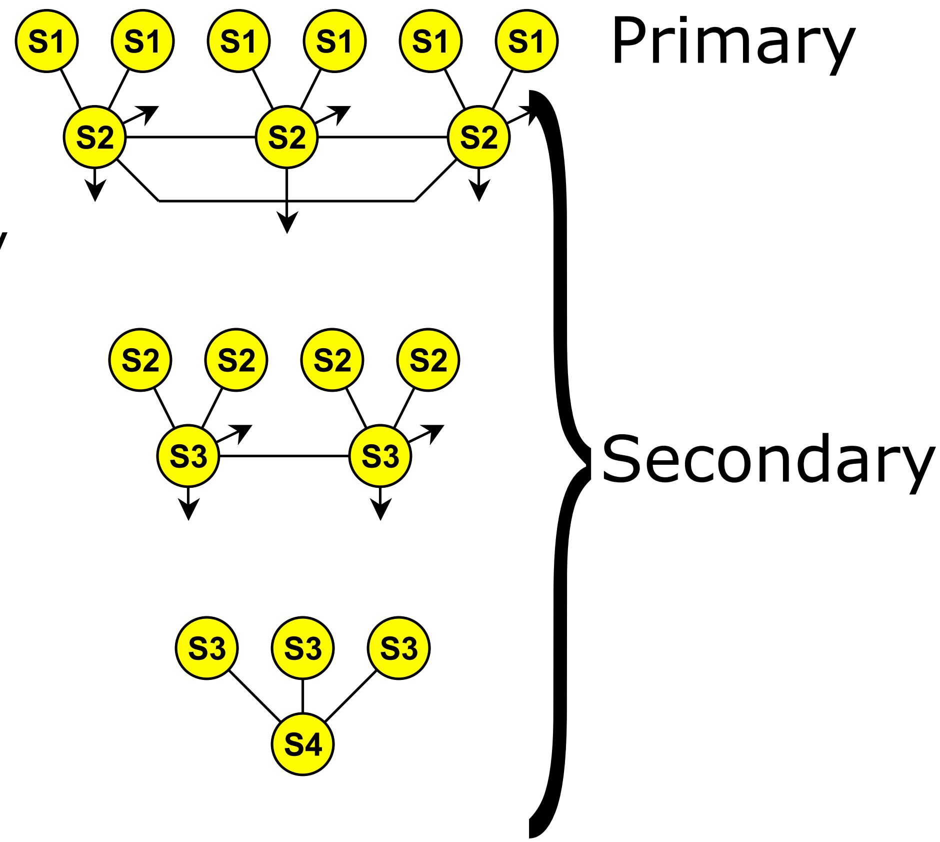
- Localization
  - Time of arrival methods require tight time synchronization between sender and receiver or among multiple receivers
- Coordinated actuation
  - Multiple sensors in a local area make a measurement
  - Determining the direction of a moving car through measurements at multiple sensors
- At the MAC level:
  - Power-saving duty cycling
  - TDMA scheduling

Synchronization in Distributed Systems

- Well-studied problem in distributed computing
- Network Time Protocol (NTP) for Internet clock synchronization [Mil94]
- Differences: For sensor networks
  - Time synchronization requirements more stringent (μs instead of ms)
  - Power limitations constrain resources
  - May not have easy access to synchronized global clocks
  - NTP assumes that pairs of nodes are constantly connected and experience consistent communication delays
  - Often, local synchronization sufficient

Network Time Protocol (NTP)

- Primary servers (S1) synchronize to national time standards
  - o Satellite, radio, modem
- Secondary servers (S2, ...) synchronize to primary servers and other secondary servers
  - o Hierarchical subnet



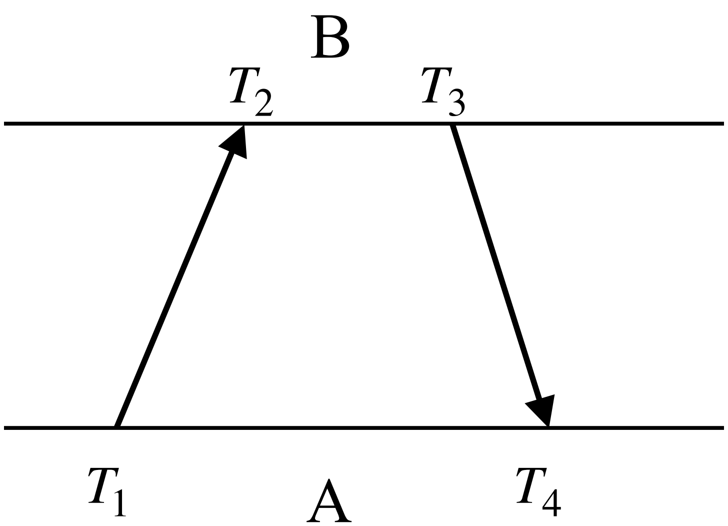
http://www.ntp.org

Measures of Interest

- Stability: How well a clock can maintain its frequency
- Accuracy: How well it compares with some standard
- Precision: How precisely can time be indicated
- Relative measures:
  - Offset (bias): Difference between times of two clocks
  - Skew: Difference between frequencies of two clocks

Synchronization Between Two Nodes

- A sends a message to B; B sends an ack back
- A calculates clock drift and synchronizes accordingly



Δ: Measured offset

d: Propagation delay

$$\Delta = \frac{(T_2 - T_1) - (T_4 - T_3)}{2}$$

$$d = \frac{(T_2 - T_1) + (T_4 - T_3)}{2}$$

Sources of Synchronization Error

- Non-determinism of processing times
- Send time:
  - Time spent by the sender to construct packet; application to MAC
- Access time:
  - Time taken for the transmitter to acquire the channel and exchange any preamble (RTS/CTS): MAC
- Transmission time: MAC to physical
- Propagation time: physical
- Reception time: Physical to MAC
- Receive time:
  - Time spent by the receiver to reconstruct the packet; MAC to application

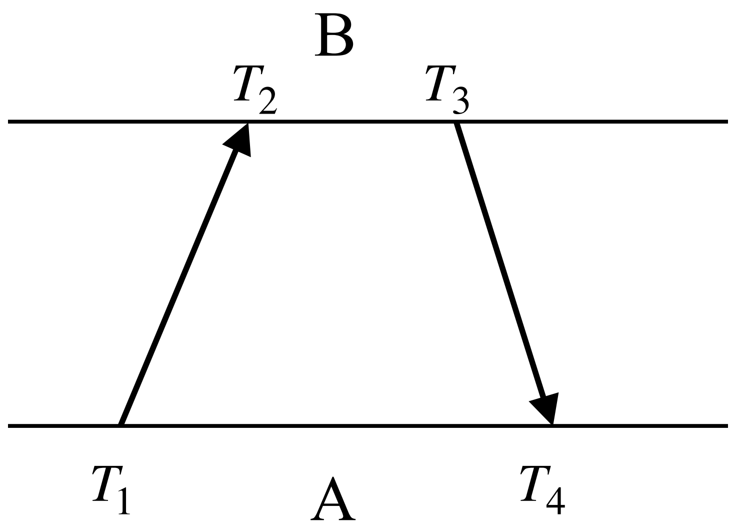
Sources of Synchronization Error

- Sender time = send time + access time + transmission time
  - Send time variable due to software delays at the application layer
  - Access time variable due to unpredictable contention
- Receiver time = receive time + reception time
  - Reception time variable due to software delays at the application layer
- Propagation time dependent on sender-receiver distance
  - Absolute value is negligible when compared to other sources of packet delay
  - If node locations are known, these times can be explicitly accounted

Error Analysis

$$\Delta = \frac{(T_2 - T_1) - (T_4 - T_3)}{2}$$

$$d = \frac{(T_2 - T_1) + (T_4 - T_3)}{2}$$



S_A : Sender time at A
R_A : Receiver time at A
$P_{A \to B}$: Prop. time for $A \to B$

S^{UC} : $S_A - S_B$
R^{UC} : $R_B - R_A$
P^{UC} : $P_{A \to B} - P_{B \to A}$

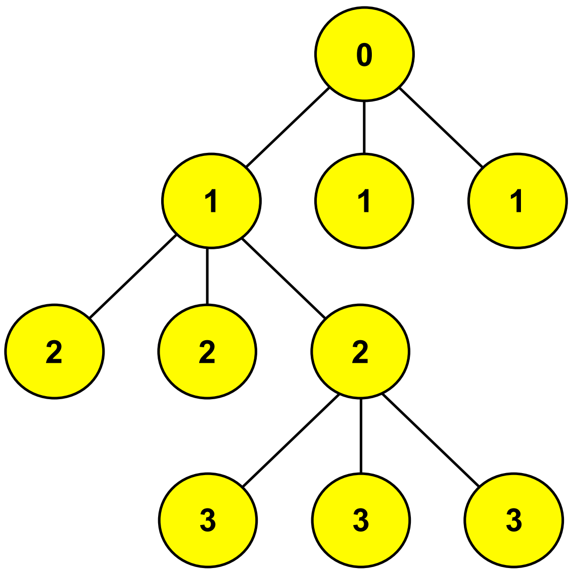
$$\text{Error} = \frac{S^{UC} + R^{UC} + P^{UC}}{2}$$

Two Approaches to Synchronization

- Sender-receiver:
  - Classical method, initiated by the sender
  - Sender synchronizes to the receiver
  - Used in NTP
  - Timing-sync Protocol for Sensor Networks (TPSN) [GKS03]
- Receiver-based:
  - Takes advantage of broadcast facility
  - Two receivers synchronize with each other based on the reception times of a reference broadcast
  - Reference Broadcast Synchronization (RBS) [EGE02]

TPSN

- Time stamping done at the MAC layer
  - Eliminates send, access, and receive time errors
- Creates a hierarchical topology
- Level discovery:
  - Each node assigned a level through a broadcast
- Synchronization:
  - Level *i* node synchronizes to a neighboring level *i-1* node using the sender-receiver procedure

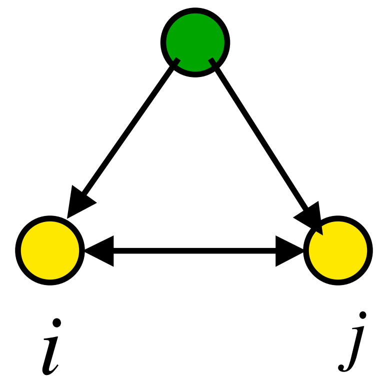


Reference Broadcast Synchronization

- Motivation:
  - Receiver time errors are significantly smaller than sender time errors
  - Propagation time errors are negligible
  - The wireless sensor world allows for broadcast capabilities
- Main idea:
  - A reference source broadcasts to multiple receivers (the nodes that want to synchronize with one another)
  - Eliminates sender time and access time errors

Reference Broadcast Synchronization

- Simple form of RBS:
  - A source broadcasts a reference packet to all receivers
  - Each receiver records the time when the packet is received
  - The receivers exchange their observations



T_i : Receive time at i

$$\Delta_{ij} = T_j - T_i$$

- General form:
  - Several executions of the simple form
- For each receiver j, receiver i derives an estimate of Δ_{ij}

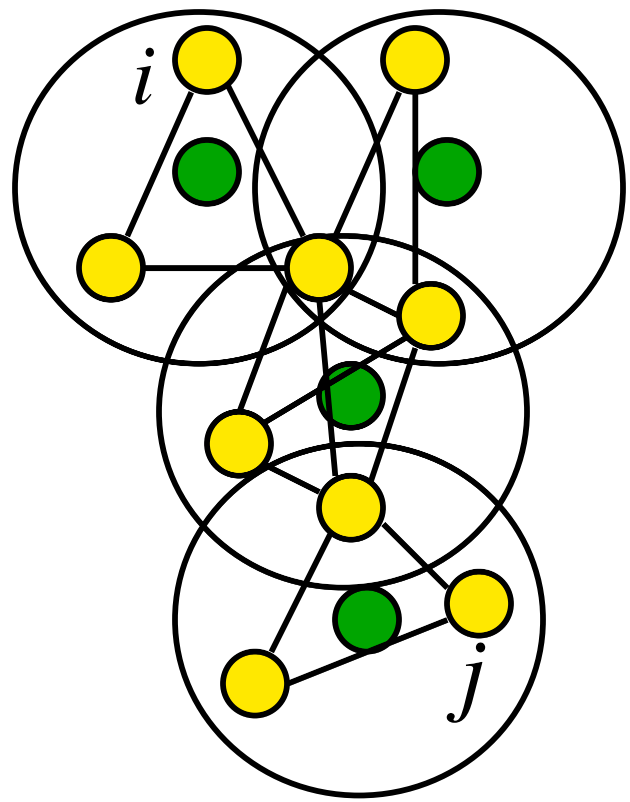
$$\Delta_{ij} = \frac{1}{m}\sum_{k=1}^{m}(T_{kj} - T_{ki})$$

Reference Broadcast Synchronization

- Clock skew:
  - Averaging assumes s_{ij} equals 1
  - Find the best fit line using least squares linear regression
  - Determines s_{ij} and Δ_{ij}

$$t_j = t_i s_{ij} + \Delta_{ij}$$

- Pairwise synchronization in multihop networks:
  - Connect two nodes if they were synchronized by same reference
  - Can add drifts along path
  - But which path to choose?
  - Assign weight equal to root-mean square error in regression
  - Select path of min-weight



Query Processing in Sensor Networks

The Sensor Network as a Database

- From the point of view of the user, the sensor network generates data of interest to the user
- Need to provide the abstraction of a database
  - High-level interfaces for users to collect and process continuous data streams
- TinyDB [MFHH03], Cougar [YG03]
  - Users specify queries in a declarative language (SQL-like) through a small number of gateways
  - Query flooded to the network nodes
  - Responses from nodes sent to the gateway through a routing tree, to allow in-network processing
  - Especially targeted for aggregation queries
- Directed diffusion [IGE00]
  - Data-centric routing: Queries routed to specific nodes based on nature of data requested

Challenges in Sensor Network Databases

- A potentially highly dynamic environment
- Relational tables are not static
  - Append-only streams where useful reordering operations too expensive
- High cost of communication encourages in-network processing
- Limited storage implies all the data generated cannot be stored
- Sensor nodes need to determine how best to allocate its sensing tasks so as to satisfy the queries
- Data is noisy
  - Range queries and probabilistic or approximate queries more appropriate than exact queries

Classification of Queries

- Long-running vs ad hoc
  - Long-running: Issued once and require periodic updates
  - Ad hoc: Require one-time response
- Temporal:
  - Historical
  - Present
  - Future: e.g., trigger queries
- Nature of query operators
  - Aggregation vs. general
- Spatial vs. non-spatial

The DB View of Sensor Networks

Traditional

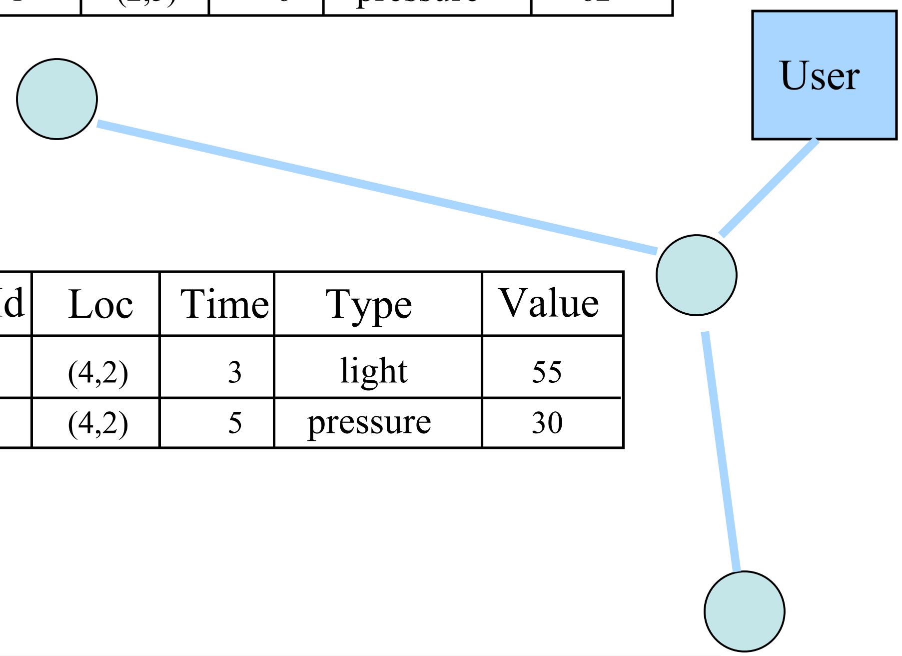
Procedural addressing of individual sensor nodes; user specifies how task executes; data is processed centrally.

DB Approach

Declarative querying and tasking; user isolated from "how the network works"; in-network distributed processing.

[TinyDB, Cougar]

SensId	Loc	Time	Type	Value
1	(2,5)	3	temperature	60
1	(2,5)	6	pressure	62



SensId	Loc	Time	Type	Value
2	(4,2)	3	light	55
2	(4,2)	5	pressure	30

SensId	Loc	Time	Type	Value
3	(3,1)	3	humidity	70

Example queries

- Snapshot queries:
  - What is the concentration of chemical X in the northeast quadrant?

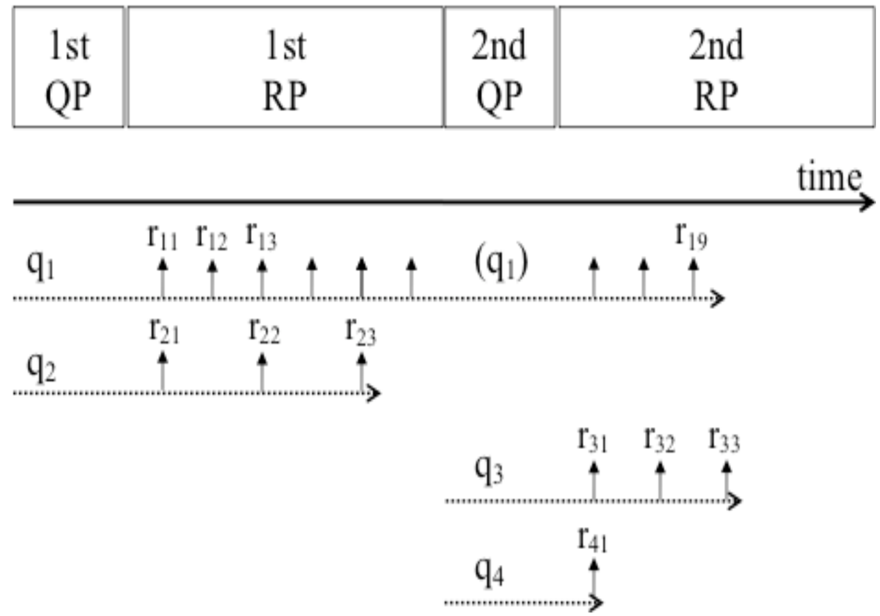
```
SELECT    AVG(R.concentration)
FROM      Sensordata R
WHERE     R.loc in (50,50,100,100)
```

- Long-running queries
  - Notify me over the next hour whenever the concentration of chemical X in an area is higher than my security threshold.

```
SELECT    R.area,AVG(R.concentration)
FROM      Sensordata R
WHERE     R.loc in rectangle
GROUP BY  R.area
HAVING    AVG(R.concentration)>T
DURATION  (now,now+3600)
EVERY     10
```

Query processing model

- ❑ Users pose queries of varying frequencies and durations at gateway
- ❑ Queries dispatched every epoch
- ❑ Query propagation phase:
  - o Queries propagated into the network
- ❑ Result propagation phase:
  - o Query results propagated up to the gateway in each round of the epoch



Query propagation

- Construct an aggregation tree
- Propagate query information for epoch along the tree
  - Query vectors, frequencies, durations
- The tree will be used for result propagation during the epoch
- What aggregation tree to construct?
- How should a node schedule its processing, listening, receiving and transmit periods?
  - Can be done on the basis of the level in the aggregation tree
- How should we adapt the aggregation tree to network changes?

Result propagation

- Given an aggregation tree and query workload, find an energy-efficient result propagation scheme
- In-network processing
  - o Sensor network is power (and bandwidth) constrained
  - o Local computation is much cheaper than communication