Infrastructure Management

References...

2

- H. Karl and A. Willing. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2005. (**Chapters 7, 8, 9, and 10**)
- C. S. Raghavendra, K. M. Sivalingam, and T. Znati, Editors. *Wireless Sensor Networks*. Springer Verlag, Sep. 2006. (**Chapters 15 and 16**)
- B. Krishnamachari. *Networking Wireless Sensors*. Cambridge University Press, Dec. 2005. (**Chapter 3**)
- J. Hightower and G. Borriello. Location Systems for Ubiquitous Systems. IEEE Computer, Vol. 34, No. 8, pp 57-66. August 2001.
- F. Zhao and L. Guibas. Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann Publishers, 2004. (**Chapter s 2 and 4**)
- N. P. Mahalik. Editor. Sensor Networks and Configuration: Fundamentals, Standards, Platforms, and Applications. Springer Verlag, 2007. (**Chapters 6, 13, and 14**)
- B. Krishnamachari. Networking Wireless Sensors. Cambridge University Press. 2005 (**Chapters 3 and 4**)

Localization...

3

Localization...

4

- As the primary function of the wireless sensor networks is to “observe” the physical world surround them, their “spatio-temporal” characteristics become important
- The term “localization” refers to the techniques and mechanisms that measure these characteristics

Location information...

5

- Important for a number of reasons:
  - **To provide location timestamps**—determine the location of the sensing
  - **To provide location and tracking of objects** in the environment
  - **To monitor the spatial evolution of a phenomenon**—important for in-network processing and routing
  - **To determine (the quality of) coverage**—to keep track of the extent of spatial coverage
  - **To achieve network management**—routing, energy conservation, clustering, ...
  - **To perform efficient spatial querying**—scoping the queries for efficiency

Key issues...

6

- What to localize?
  - Refers to identifying which nodes have a priori known locations (reference nodes) and which do not (unknown nodes)
- When to localize?
  - At the beginning of the operation or on-the-fly
- How well to localize?
  - The resolution; absolute, relative
- Where to localize?
  - The location of computation: centrally or distributed iteration; distributed within unknown nodes
- How to localize?
  - Technique(s) to use

Properties of localization mechanisms...

7

- Physical position and symbolic location
  - GPS provides physical location; symbolic location abstracts ideas of where something is: e.g., in the kitchen
- Absolute vs relative coordinates
  - Absolute system uses a "shared" reference; each object has its own frame in a relative system
- Location computation
  - Each object computes its own location; the object broadcasts or emit telemetry to be located
- Accuracy and precision
  - Within 10 meters (accuracy); 99% of the time (precision)

Properties of localization mechanisms...2

8

- Scale
  - Objects worldwide? Within a metro area? ...
- Recognition
  - Tag scanners recognize and classify items
- Cost
  - Time; space, capital; incremental
- Limitations
  - Environment; ...

Approaches…

9

- Coarse grained localization with minimal information
  - Binary proximity; centroid calculation; geometric constraints; approximate point in triange (APiT); …
- Fine grained localization with detailed information
  - Radio signal strength (RSS); time difference of arrival (TDoA); time of arrival (ToA); angle of arrival (AoA); trilateration, pattern matching (RADAR)

Binary proximity…

10

- Simple decision—whether two nodes are within the range of each other
- A set of "reference" nodes are deployed in nearly overlapping positions
- Information exchange
  - Reference nodes emit periodic beacons
  - Unknown node transmits a beacon when it needs localization
- Example
  - Active Badge System
- Larger application of localization is with RFID tags

Centroid calculation…

11

- Applicable to dense reference nodes
- In a two dimensional scenario, where *n* nodes are detected within the proximity of the unknown node
- The location (x_u, y_u) of node *n* is:

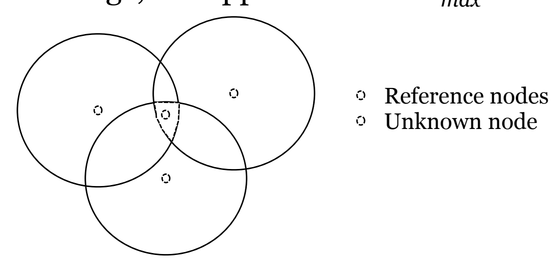
$$x_u = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad y_u = \frac{1}{n}\sum_{i=1}^{n} y_i$$

- A *4x* increase of the overlap ratio *R/d* average RMS error in localization is reduced by half ($0.5d \rightarrow 0.25d$)

Geometric constraints…

12

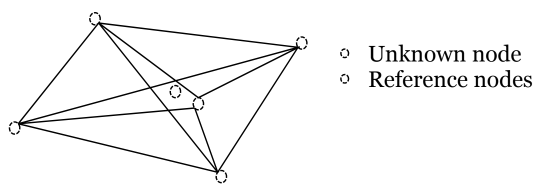
- Radio coverage of a node for a given node resembles a geometric shape
  - In a circular coverage, the upper bound is R_{max}



  - Alternative shapes include, sector, quadrants, or annulus

Approximate point in triangle…

- A similar approach to geometric constraints is the Approximate-point-in-triangle technique.
- The regions are defined as triangles between three reference nodes.



Radio signal-strength based…

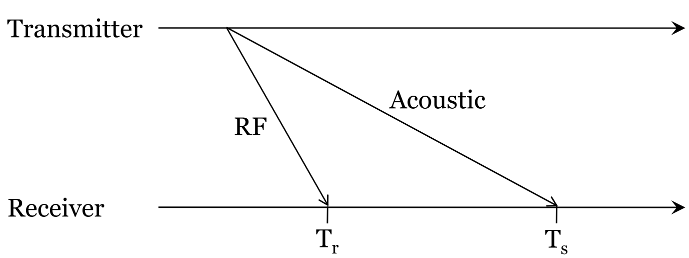
- Mean radio signal strengths diminish with distance
- The following is used to model wireless radio propagation:

$$P_{t,dB}(d) = P_{t,dB} - \eta 10 \log\left(\frac{\mathrm{d}}{\mathrm{d_0}}\right) + X_{\sigma,dB}$$

- Where
  - $P_{r,dB}(d)$ is the received power at distance d
  - $P(d_o)$ is the received power at some reference distance d_o
  - $X_{\sigma,dB}$ a log normal random variable with variance σ^2 that accounts for fading effects

Time difference of arrival…

- Use time of transmission, propagation speed, time of arrival to compute distance
- Problem: Exact time synchronization



Distance ≈ $(T_r - T_s) \cdot V_s$

Trilateration…

- Assuming distances to three points with known location are exactly given
- Solve system of equations (Pythagoras!)
  - (x_i, y_i) : coordinates of ***anchor point*** *i*, r_i distance to anchor *i*
  - (x_u, y_u) : unknown coordinates of node

$$(x_i - x_u)^2 + (y_{1i} - y_u)^2 = r_i^2 \; for \; i = 1, \ldots, 3$$

  - Subtracting eq. 3 from 1 & 2:

$$(x_1 - x_u)^2 - (x_3 - x_u)^2 + (y_1 - y_u)^2 - (y_3 - y_u)^2 = r_1^2 - r_3^2$$
$$(x_2 - x_u)^2 - (x_2 - x_u)^2 + (y_2 - y_u)^2 - (y_2 - y_u)^2 = r_2^2 - r_3^2.$$

  - Rearranging terms gives a linear equation in (x_u, y_u)!

$$2(x_3 - x_1)x_u + 2(y_3 - y_1)y_u = (r_1^2 - r_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2)$$
$$2(x_3 - x_2)x_u + 2(y_3 - y_2)y_u = (r_2^2 - r_3^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2)$$

Trilateration…2

17

- Rewriting as a matrix equation:

$$2\begin{bmatrix} x_3 - x_1 & y_3 - y_1 \\ x_3 - x_2 & y_3 - y_2 \end{bmatrix} \begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} (r_1^2 - r_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \\ (r_2^2 - r_2^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2) \end{bmatrix}$$

- Example: $(x_1, y_1) = (2,1)$, $(x_2, y_2) = (5,4)$, $(x_3, y_3) = (8,2)$, $r_1 = 10^{0.5}$, $r_2 = 2$, $r_3 = 3$

$$2\begin{bmatrix} 6 & 1 \\ 3 & -2 \end{bmatrix} \begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} 64 \\ 22 \end{bmatrix}$$

$(x_u, y_u) = (5,2)$

Pattern matching…

18

- Use a "pre-determined" map of the environment
- Use the map to "pattern match" the measurements of the unknown node to locate its position
- Requires prior collection of measurements of signal strengths

Network wide localization…

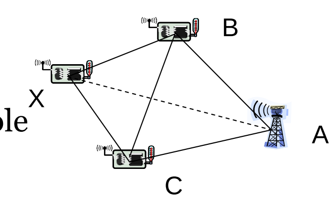
19

- A broader problem in wireless sensor systems is the localization of the network itself (multiple node localization)
- The solution depends on the resources; different scenarios:
  - No reference nodes; only relative coordinates of the nodes can be found
  - Number of the reference nodes may vary
  - Only a single reference node, which is mobile
- Centralized or distributed solutions
- An alternative is to estimate the distance of a node which is several hops away

Multihop range estimation…

20

- How to estimate range to a node to which no direct radio communication exists?
  - No RSSI, TDoA, …
  - But: Multihop communication is possible



- Three variants:
  - DV-hop: Count number of hops, assume length of one hop is known
  - DV distance: If range estimates between neighbors exist, use them to improve total length of route estimation in previous method
  - Euclidean propagation: Additionally, use geometric relations

Taxonomy of localization systems…

21

- Active—emit signals into the environment
  - Non-cooperative where the signals distorted or reflected by passive elements
  - Cooperative targets emit a signal with known characteristics; when detected the target is known
  - Cooperative infrastructure elements emit a signal and targets receive it
- Passive—monitor the existing signals (can emit, too)
  - Blind source, where the type of signal is not known
  - Passive target coherent combination of signals
  - Self location existing beacon signals from known infrastructure
- Best of the worlds: Active and cooperative techniques tend to be more accurate, more efficient, and more effective

Current location sensing technologies…

22

- Active Badge
  - First indoor badge sensing system consists of a cellular proximity system that uses diffuse infrared system
- Active Bat
  - Uses an ultrasound time-of-flight lateration technique
- Cricket
  - Complements Active Bat system uses ultrasound emitters to create the infrastructure and embeds receivers in the objects being located
- RADAR
  - A building-wise tracking system based on IEEE 802.11 WaveLAN technology
- Motionstar Magnetic tracker
  - Offers a classic position tracking method by generating axial DC magnetic-field pulses from a fixed antenna
- Easy Living
  - Explores computing vision technology to figure out the locations of things
- Smart Floor
  - Embedded pressure sensors capture footfalls and determines the position of the pedestrian
- E911
  - A new addition to 911 emergency system, where any call from a cell phone would be located within 150 mteters for 95% of time
- Sensor Fusion
  - Uses multiple technologies simultaneously to form hierarchical and overlapping levels of sensing
- Ad hoc location sensing
  - Aims at locating objects without drawing on the infrastructure or central control

Summary…

23

- Determining location or position is a vitally important function in WSN, but many errors and shortcomings:
  - Range estimates often not sufficiently accurate
  - Many anchors are needed for acceptable results
  - Anchors might need external position sources (GPS)
  - Multilateration problematic (convergence, accuracy)
- Design space of localization algorithms is large; selection depends on many key factors

Topology control and sensor tasking

24

Topology control…

- Sensors deployed in an area provide data about the environmental phenomena to the sink node(s)
- Not all sensors may be able to transmit their data directly to the sink node(s)
- Some sensors must assume roles of both data provider and router
- Generally, there are more sensors in the network at a given time to perform the task(s) than needed
- Management of sensors (infrastructure) is needed to
  - Control topology (which nodes will function as what) and
  - Sensor tasking (which nodes should sense data)

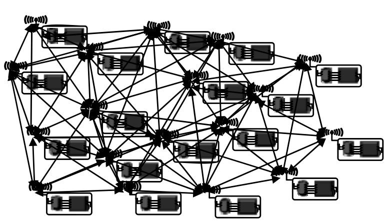
Topology control…2

- The goal of *topology control* is to design efficient algorithms that maintain network connectivity and to optimize performance metrics
- If sensors are not needed at a given time to provide data or routing services, they can save energy by powering down or halt traffic until a later time
- Maintain sufficient number of nodes to provide a connected network at all times
- MANET protocols have identical goals

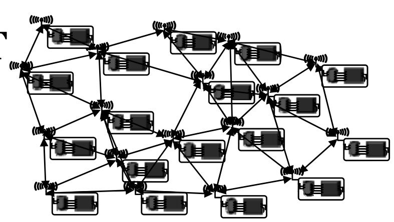
Topology control…3

- Take a graph *G=(V, E)*, representing the network and transform it to graph $T=(V_T, E_T)$, such that
  - *T* consists of all nodes in *G*, but has fewer edges
  - If nodes *u* and *v* are connected in *G*, they are still connected in *T*
  - A node *u* can transmit to all neighbors in *T* using less power than it is required to transmit to all its neighbors in *G*

G



T



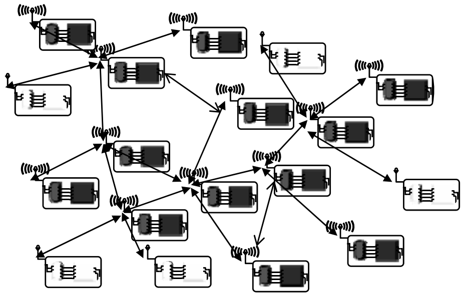
Aspects of topology control…

- **Connectivity—**If two nodes connected in G, they have to be connected in T resulting from topology control
- **Stretch factor—**should be small
  - **Hop stretch**: how much longer are paths in T than in G?
  - **Energy stretch**: how much more energy does the most energy-efficient path need?
- **Graph metrics**—small number of edges, low maximum degree for each node
- **Throughput—**removing nodes/links reduces throughput, but how much?
- **Robustness to mobility**
- **Algorithm overhead**

Topology control: Flat networks…

29

- Control "node" activity by powering off nodes with low energy reserves, and powering on others
  - Still redundancy
- Control "link" activity by discarding some links, keeping crucial ones



Topology control: Flat networks…2

30

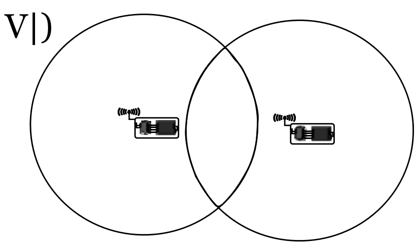
- Bounds on critical parameters:
  - Transmitting range—geometric random graphs
    - Assumes a unit disk graph model and a uniform distribution of nodes in a given area
  - Number of neighbors—expected numbers grow logarithmically

Topology control: Flat networks…3

31

The relative neighborhood graph (RNG)

- Shaded area should NOT contain another node (a witness)
- Edge between nodes *u* and *v* if and only if there is no other node *w* that is closer to either *u* or *v* than *u* an *v* apart from each other (remove the longest edge from a triangle)
- Maintains connectivity of the original graph
- Worst-case spanning ratio is Ω (|V|)
- Easy to compute locally
- Average degree is 2.6

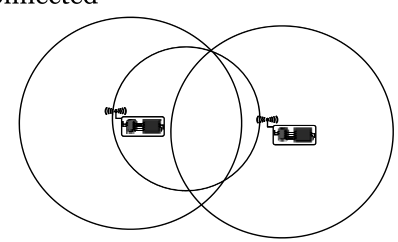


Topology control: Flat networks…4

32

The Gabriel Graph

- The shaded area must not contain a node
- Similar to RNG
  - Smallest circle with nodes *u* and *v* on its circumference must only contain node *u* and *v* for *u* and *v* to be connected

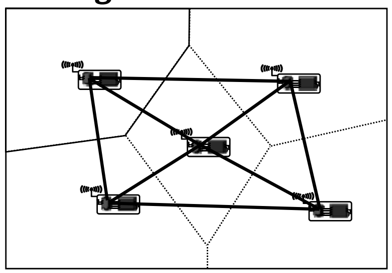


- Properties
  - Worst-case spanning ratio $\Omega(|\sqrt{V}|)$
  - Maintains connectivity
  - Worst-case degree $\Omega(|V|)$
  - Energy stretch O(1)
    - depending on precise energy consumption model

Topology control: Flat networks…5

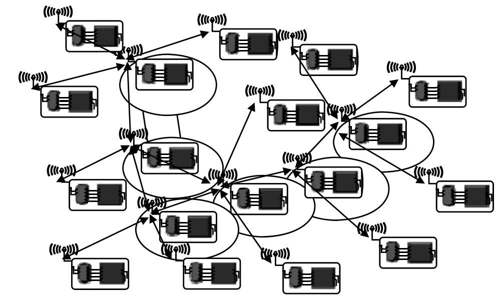
Delaunay triangulation

- Assign, to each node, all points in the plane for which it is the closest node → **Voronoi diagram**; constructed in O(|V| log |V|) time
- Connect any two nodes for which the Voronoi regions touching each other →**Delaunay triangulation**
- Problems
  - might produce very long links
  - not well suited for power control



Topology control: …by dominating sets…

- Some nodes and links can be re-arranged to form a "hierarchy", where some nodes assume special roles
- Controlling nodes form a "dominating set" (the backbone)



Topology control: …by dominating sets…2

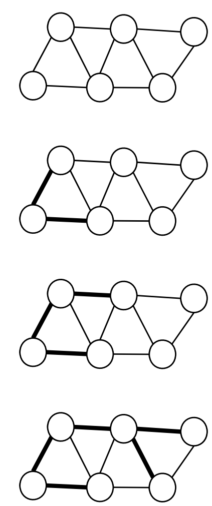
- A connected, minimal, dominating set (MDS or MCDS)
- Protocols like routing are confronted with a simple topology—from a simple node, route to the backbone, routing in backbone is simple (few nodes)
- Problem: MDS is an NP-hard problem
  - Hard to approximate, and even approximations need quite a few messages
- Examples
  - Growing a tree: naïve approach; choosing gray nodes
  - Connecting separate components: finding a non-connected dominating set; ensuring connectivity—Steiner tree

Topology control: …by dominating sets…3

- Growing a tree—a naïve approach
  - Initialize all nodes' color to white
  - Pick an arbitrary node *u* and color it gray

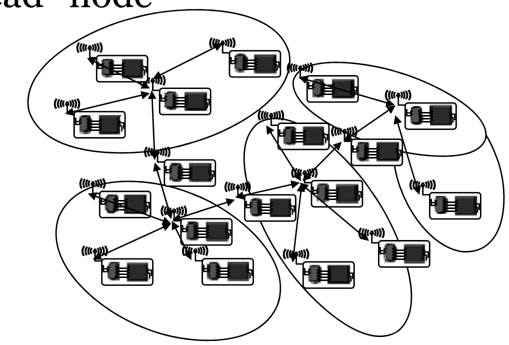
  While there are nodes
  1. Pick a gray node *v* that has white neighbors
  2. Color the gray node *v* black
  3. For each white neighbor *u* of *v*
     1. Color *u* gray
     2. Add *(v,u)* to tree *T*

  End while



Topology control: …by clustering…

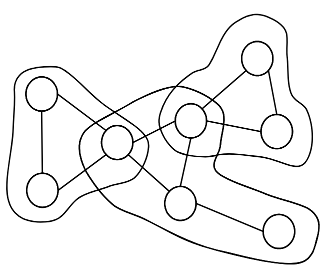
- Alternatively, the nodes can be partitioned into "clusters"
- Each node is in exactly one cluster
- Each cluster have a "head" node



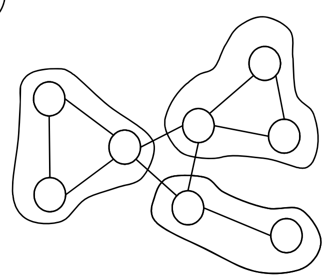
Topology control: …by clustering…2

- Given a graph *G=(V,E),* clustering is simply identifying a set of subset of nodes
- Questions:
  - Are there **clusterheads**?
    - One controller/representative node per cluster
  - May clusterheads be neighbors?
    - If no, clusterheads form an **independent set C**
    - Typically, clusterheads form a **maximum independent set**
  - May clusters overlap?
  - How do the clusters communicate?
  - How many gateways between clusters?
  - …

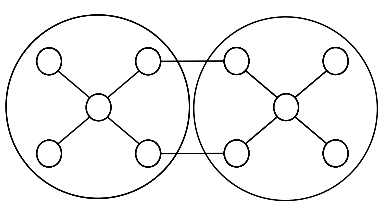
Topology control: …by clustering…3



Overlapping clusters



Non-overlapping clusters



Clusters with distributed gateways

Topology control: …by clustering…4

Rotating clusters

- Serving as a clusterhead can put additional burdens on a node
  - For MAC coordination, routing, …
- Let this duty rotate among various members
  - Periodically reelect—useful when energy reserves are used as discriminating attribute
  - LEACH—determine an optimal percentage P of nodes to become clusterheads in a network

Topology control: …by clustering…5

Multihop clusters

- Clusters with diameters larger than 2 can be useful, e.g., when used for routing protocol support
- Extend "domination" definition to also dominate nodes that are at most *d* hops away
- Find a smallest set *D* of dominating nodes with this extended definition of dominance
- Only somewhat complicated heuristics exist
- Different tilt: Fix the **size** (not the diameter) of clusters
  - Use **growth budgets**—amount of nodes that can still be adopted into a cluster, pass this number along with broadcast adoption messages, reduce budget as new nodes are found

Combined mechanisms…

- Pilot based power control
  - Similar to those found in cellular networks
  - After initial clustering is done, clusterheads use power control on both pilot signals and on data packets
    - Data packet power control assures low errors for faraway nodes
- Ad hoc network design algorithm (ANDA)
  - Allow clusterheads to control the size of their clusters
  - Assumes that positions of the nodes (both ordinary and clusterheads) are known
  - Traffic load is evenly distributed over ordinary nodes
  - The lifespan of a clusterhead is proportional to its energy, …

Adaptive mechanisms…

- Turn some nodes off deliberately
- Only possible if other nodes remain on that can take over their duties
- Examples
  - Geographic Adaptive Fidelity (GAF)
  - SPAN
  - **A**daptive **S**elf-**c**onfiguring S**e**nsor **N**etworks' **T**opologies (ASCENT)
  - Sparse topology and energy management (STEM)

Sensor tasking…

- Sometimes decisions are made not only influencing how data are routed, but also what traffic is generated
  - E.g., determine sensing frequency and data resolution
- The process of determining the subset of sensors chosen for activation is influenced by QoS (or fidelity) requirements of applications, such as
  - Coverage of entirety or just a portion of a region
  - Edge detection, where only a certain subset of nodes is expected to change state when the edge moves
  - Oversampling, where nodes adjust the resolution of their data in dense regions without significant information loss
  - Target localization or tracking, where only the subset of sensors in the current vicinity is beneficial
  - General target classification

Sensor tasking…2

45

- Probing environment and adaptive sleeping (PEAS)
  - Nodes periodically wakeup and probe; if no response, become active
- Node self-scheduling scheme (NSSS)
  - A node measures its neighborhood redundancy as the union of the sectors covered by neighboring sensors within the range; if the coverage is 360 degrees, power off
- GUR game
- Reference time-based scheduling scheme
  - Environment is divided into a grid and coverage is maintained continuously, while minimizing the number of active sensors
- Coverage configuration protocol
  - Maintain a degree of coverage

Summary…

46

- Various approaches exist to trim the topology of a network to a desired shape
- Most of them bear some non-negligible overhead
  - At least: Some distributed coordination among neighbors, or they require additional information
  - Constructed structures can turn out to be somewhat brittle – overhead might be wasted or even counter-productive
- Benefits have to be carefully weighted against risks for the particular scenario at hand

Time synchronization…

47

Introduction…

48

- In addition to the "standard" problem in distributed systems, wireless sensor networks have additional constraints:
  - The energy consumption of the algorithms
  - The potentially large number of nodes
  - Varying precision requirements
- Algorithms, protocols, (debugging) tools all need time synchrony

The time…

49

- WSN have a direct coupling to the physical world, hence their notion of time should be related to **physical time**:
  - Physical time; wall clock time, real-time, i.e. one second of a WSN clock should be close to one second of real time
    - Commonly agreed time scale for real time is UTC, generated from atomic clocks and modified by insertion of leap seconds to keep in synch with astronomical timescales (one rotation of earth)

Node clocks…

50

- Almost all clock devices of sensor nodes have
  - an **oscillator** of a specific frequency and
  - a **counter register**, which is incremented in hardware after a certain number of oscillator pulses
- The value of the **hardware clock** of node *I* at real time *t* is represented as $H_i(t)$.
- The value of a local **software clock** $L_i(t)$ can be computed as (clock adjustment)

  $L_i(t) = \theta_i \cdot H_i(t) + \varphi_i$

  where φ_i is called **phase shift** and θ_i is called **drift rate**

Synchronization…

51

- External synchronization (agreement)
  - External real time scale like UTC
  - Nodes $i=1, \ldots, n$ are accurate at time t within bound δ when $|L_i(t) - t| < \delta$ for all i
    - Hence, at least one node must have access to the external time scale
- Internal synchronization (accuracy)
  - No external timescale, nodes must agree on common time
  - Nodes $i=1, \ldots, n$ agree on time within bound δ when $|L_i(t) - L_j(t)| < \delta$ for all i,j

Properties of time synchronization algorithms…

52

- Physical time vs. logical time
- External vs. internal synchronization
- Global vs. local algorithms
  - Keep all nodes of a WSN synchronized or only a local neighborhood?
- Absolute vs. relative time
- Hardware vs. software-based mechanisms
  - A GPS receiver would be a hardware solution, but often too heavyweight, costly, and energy-consuming in WSN nodes, and in addition a line-of-sight to at least four satellites is required

Properties of time synchronization algorithms...2

- A-priori vs. a-posteriori synchronization
  - Is time synchronization achieved before or after an interesting event?
    - ➔ Post-facto synchronization
- Deterministic vs. stochastic precision bounds
- Local clock update discipline
  - Should backward jumps of local clocks be avoided?
  - Avoid sudden jumps?

Performance metrics…

- Precision
  - maximum synchronization error for deterministic algorithms, error mean, standard deviation, or quantiles for stochastic ones
- Energy costs;
  - e.g., number of exchanged packets, computational costs
- Memory requirements
- Fault tolerance
  - what happens when nodes die?

Building blocks…

- *Resynchronization event detection block*: when to trigger a time synchronization round?
  - Periodically?
  - After external event?
- *Remote clock estimation block*: figuring out the other nodes clocks with the help of exchanging packets
- *Clock correction block*: compute adjustments for own local clock based on estimated clocks of other nodes
- *Synchronization mesh setup block*: figure out which node synchronizes with which other nodes

Constraints…

- An algorithm should scale to large networks of unreliable nodes
- Quite diverse precision requirements, from ms to tens of seconds
- Use of extra hardware (like GPS receivers) is mostly not an option
- low mobility
- Often there are no fixed upper bounds on packet delivery times (due to MAC delays, buffering, ...)
- Negligible propagation delay between neighboring nodes
- Manual node configuration is not an option

Protocols…

- Based on sender/receiver synchronization
  - Lightweight time synchronization protocol (LTS)
  - Timing-synch protocol for sensor networks (TPSN)
- Based on receiver/receiver synchronization
  - Reference broadcast synchronization (RBS)
  - Hierarchy referencing time synchronization (HRTS)

Lightweight time synchronization protocol…

- LTS attempts to synchronize the clocks of a sensor network to the clocks help by certain reference nodes, which for example, may have GPS receivers
- Protocol phases
  - Pair-wise synchronization of two neighboring nodes
  - Spanning tree construction from the reference node to all nodes

LTS: Pair-wise synchronization…

- Node *i* forms a sync request packed, time stamped at t_i with time $L_i(t_i)$ and passes it to the OS and it is delayed for while
- When node *i* starts sending the first bit at time t_2, node *j* received the last bit of the packet at time $t_3=t_2+\tau+t_p$
- At time t_4, the packet arrival is signaled to node *j*'s OS and it is time stamped at time t_5 with $L_j(t_5)$
- At time t_6, node *j* has formatted its packet, time stamps it with $L_j(t_6)$ and hand it over to its OS; this packet includes previous $L_j(t_5)$ and $L_i(t_1)$
- Node *i* receives the packet at time t_7 (t_6 plus OS and NW overhead, medium access delay, propagation delay, packet transmission time, and interrupt latency) and timestamps it at time t_8 with $L_i(t_8)$
- Clock correction observations:
  - There is one propagation delay τ plus one packet transmission time t_p between t_1 and t_5
  - There is another time $\tau+t_p$ between t_5 and t_8 (assume identical propagation delays)
  - The time between t_5 and t_6 is known by node *i* as $L_j(t_6)-L_j(t_5)$
- The uncertainty about t_5 is reduced to $I=[L_i(t_1)+\tau+t_p, L_i(t_8)-\tau-t_p-(L_j(t_6)-L_j(t_5))]$

$$O = \Delta(t_5) = L_i(t_5) - L_j(t_5) = \frac{L_i(t_8)+L_i(t_1)-L_j(t_6)-L_j(t_5)}{2}$$

LTS: Network-wide synchronization…

- LTS solves the task of synchronizing all nodes of a connected network with a reference node
- If a specific node has a distance of h_i hops to the reference node, and if the synchronization error is normally distributed with $\mu=0$ and $\sigma'= 2\sigma$ at each hop, and if furthermore the hops are independent, then the synchronization error of *i* is also normally distributed with variance

$$\sigma_i^2 = 4h_i\sigma^2$$

- Based on this observation, LTS aims at constructing a spanning tree of minimum height and only node pairs along the edges are synchronized

Reference broadcast synchronization (RBS)

This protocol also consists of two phases:

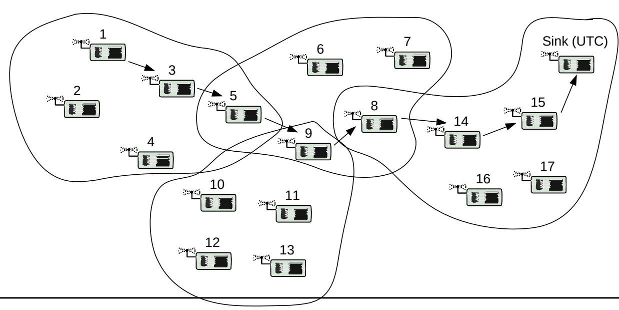
- A set of nodes within a single broadcast domain estimate their peers' clocks
  - does not modify the local clocks of nodes, but computes a table of conversion parameters for each peer in a broadcast domain
  - RBS allows for post-facto synchronization
- Relate timestamps between distant nodes with several broadcast domains between them

RBS: Synchronization in a broadcast domain…

- Two nodes *i* and *j* wants to synchronize
- At time t_o, another node *R* broadcasts a pulse packet, containing node identification and a sequence number
- Since *i* and *j* are at different distances from *R*), the last bit arrives at time t_1,i and and t_1,j, with propagation delays τ_i and τ_j, respectively
- Packet reception interrupts are generated at times $t_{2,\,i}$ and $t_{2,j}$; nodes timestamp the packets at times $t_{3,i}$, with $L_i(t_{3,i})$ and $t_{3,j}$ with $L_j(t_{3,j})$, respectively
- Nodes *i* and *j* exchange packets with their timestamps and identities
- Both nodes can calculate their relative phase shifts of their local clocks by assuming $t_{3,i} = t_{3,j}$. Specifically, node *i* stores the value $O(t_{3,i}) = L_i(t_{3,i}) - L_j(t_{3,j})$ as the phase offset in a local table without readjusting its clock
- The time between receiving the last bit and timestamping the packet is called **receiver uncertainty** and is denoted by $\delta_{r,i}$ and $\delta_{r,j}$, for nodes *i* and *j*, respectively

RBS: Synchronization over multiple hops…

- Local synchronization with broadcasting covers only a small region
- The idea is not to produce a global time scale, but to convert a packet's timestamp at each hop into the next hops nodes time scale until it reaches the final destination



Naming and addressing…

Addresses and names in sensor networks…

65

- Names and addresses are used to find "things" in a network
  - **Names** denote things (nodes, data, transactions)
  - **Addresses** supply the information to find those things
- In traditional networks, independent nodes and the data they host are named and addressed
- In sensor networks, nodes are not independent in that they collaborate to solve a given task; a shift is necessary from naming nodes → naming data
- This is done by a protocol stack
- We focus on address allocation and representation, and their proper use in wireless sensor networks

Addresses and names in sensor networks…2

66

- Unique node identifier (UID)—persistent data item unique for every node
- MAC address—used to distinguish between one-hop neighbors of a node
- Network address—used to denote a node over multiple hops
- Network identifier—used to identify identical nodes in a overlapping sensor environment
- Resource identifier—used to represent an object a user-understandable terms

Address management tasks…

67

- Address allocation
  - assignment of an address from a pool
- Address deallocation
  - allows address re-use (abrupt, or graceful)
- Address representation
  - format of an address
- Conflict detection/resolution
  - Self explanatory…
- Binding
  - mapping between different layers

Content-based addressing…

68

- Traditional fixed and ad hoc networks
  - offer services and protocols that allow a number of independent users to exchange data among themselves AND the remaining world
  - uses a naming system and binding to map names to lower level entities, such as IP addresses
- Nodes in wireless sensor networks
  - on the other hand, interact with the physical world AND they collaborate
  - levels of indirection are not needed; *data-centric addressing* is adopted