

## Clock Synchronization in Sensor Networks

- Fine-Grained Network Time Synchronization using Reference Broadcasts  
Jeremy Elson, Lewis Girod and Deborah Estrin  
OSDI 2002
- Optimal and Global Time Synchronization in Sensornets  
Richard Karp, Jeremy Elson, Deborah Estrin, and Scott Shenker  
CENS Technical Report #12 , April 2003
- Global Clock Synchronization in Sensor Networks  
Qun Li and Daniela Rus  
IEEE Infocom 2004

Presenter: Daniel R. Figueiredo  
May, 2004

- Based on slides from authors and Internet

## Why Clock Sync in Sensor Nets

- Physical time needed to relate events in the physical world
- Time sync is critical at *many* layers
  - Beam-forming, localization, (sound) tracking
  - Data fusion, aggregation, caching
  - fine-grained radio scheduling
- High precision sometimes required
  - order of 1 microsecond (e.g., sleep cycles)
- Low precision sometimes sufficient
  - order of 10 milliseconds (e.g., temperature readings)

## Solved Problem?

- ❑ Clock synchronization problem
  - bound differences between reading of 2 clocks
- ❑ NTP (Network Time Protocol)
  - Ubiquitous in Internet
- ❑ 802.11 synchronization
  - Precise clock sync within a cluster
- ❑ GPS, WWVB, other radio time services
  - High precision anywhere
- ❑ High-stability oscillators (Rubidium, Cesium)

YES!

## but wait... Sensor Nets is *Different*

- ❑ Important assumptions no longer hold
- ❑ Fewer resources
  - energy, network bandwidth
- ❑ No infrastructure available
  - master NTP server?
- ❑ Sensors located on hostile environment
  - no GPS signal
- ❑ Cost and size factor
  - \$50 GPS receiver or \$500 oscillator on a \$5 mote
- ❑ High precision sometimes required

## Clock Nomenclature

- ❑ Clock rate (heartbeat)
  - clock value updated once every heartbeat
- ❑ Clock stability
  - how well it maintains a constant heartbeat frequency
  - short term variations - temperature, electricity
  - long term variations - aging of oscillator
- ❑ Clock precision and accuracy
  - how well its frequency and time compares with UTC
- ❑ Offset
  - time difference between 2 clocks
- ❑ Skew
  - frequency difference between 2 clocks

## Clock Sync Spectrum

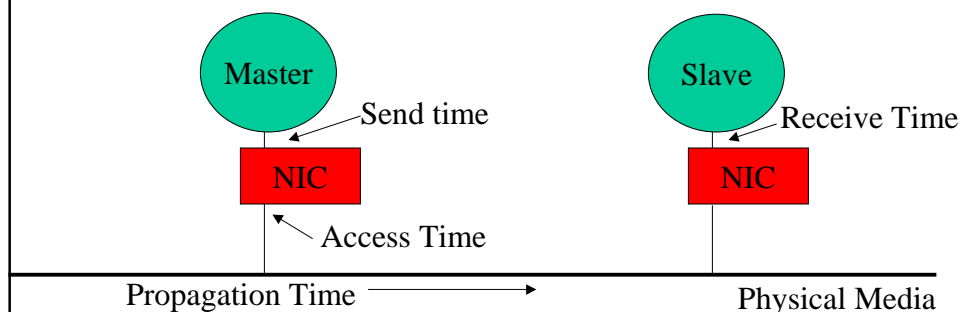
- ❑ Scope
  - global: all nodes in network have same local clock reading
  - local: a subset of nodes in network have same local clock reading
- ❑ Time-scale
  - global: time with respect to a standard (e.g., UTC)
  - local: time with respect to some clock (e.g., node 1)
- ❑ Precision (and accuracy)
  - high precision (tight bound in differences)
  - low precision (loose bound)
- ❑ Note: local sync can be used to obtain global sync (with loss in precision)

## Clock Synchronization in Sensor Networks

- Fine-Grained Network Time Synchronization using Reference Broadcasts  
Jeremy Elson, Lewis Girod and Deborah Estrin  
OSDI 2002
- Optimal and Global Time Synchronization in Sensornets  
Richard Karp, Jeremy Elson, Deborah Estrin, and Scott Shenker  
CENS Technical Report #12 , April 2003
- Global Clock Synchronization in Sensor Networks  
Qun Li and Daniela Rus  
IEEE Infocom 2004

## Traditional Local Clock Sync

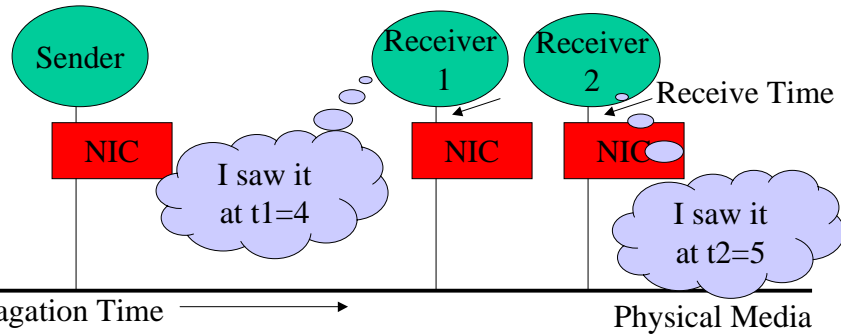
- Slave sends a message to master
- Master replies with current time
- Slave updates its local clock (removes RTT)



- **Problem:** many sources of unknown, nondeterministic latency between timestamp and its reception

## Reference Broadcasts

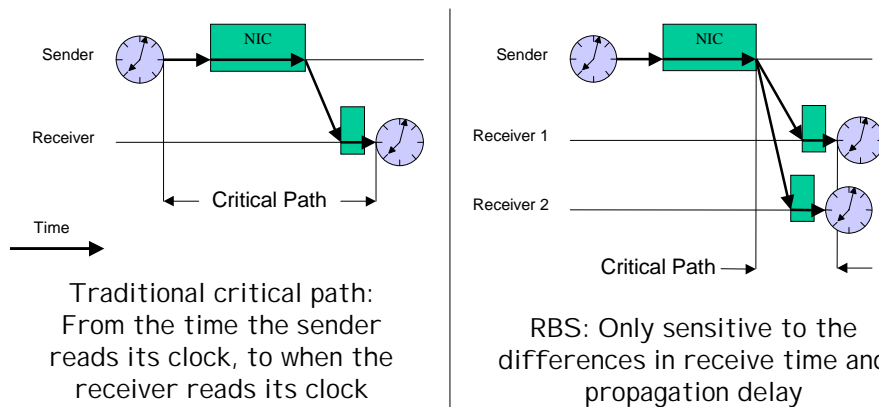
- Sender sends a broadcast reference packet
- Receivers record time of arrival
- *Receivers* exchange observations (and update clocks)



- Sync 2 receivers *with each other*, NOT sender with receiver

## Reference Broadcasts

- RBS reduces error by removing much uncertainty from critical path

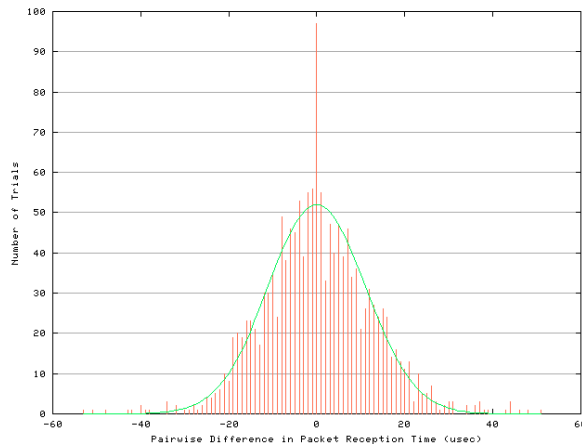


## Variations in Receive Time

- ❑ Differences in time-of-flight of packet
  - geographical distances
  - usually negligible
- ❑ Delays in recording time of packet arrival
  - read local system clock within NIC driver
  - quite deterministic
- ❑ Differences between recording time is small
  - order of transmission time of a single bit
  - can be accounted for

## Experiments with Receive Time

- ❑ Obtain exact packet arrival time (using external global clock)
- ❑ Compute differences
- ❑ Bin using 1 microsecond
- ❑ 1 bit tx time is 52 microseconds
- ❑ Error can be modeled using Gaussian distrib.



## Removing Receive Time Differences

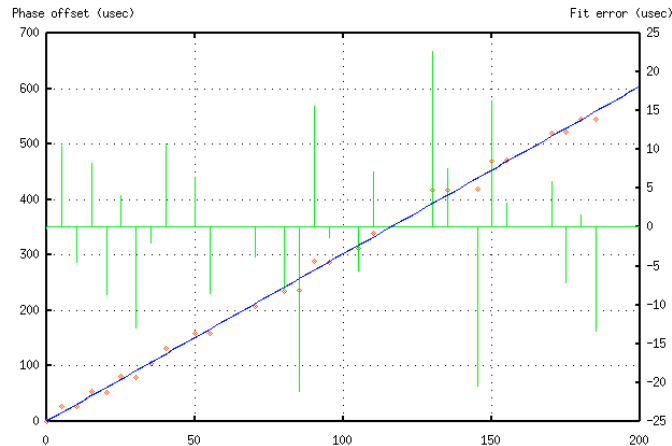
- ❑ Receive time differences are at most around transmission time of 1 bit (52 microseconds)
- ❑ Reduce this potential error by *averaging*
- ❑ Server broadcasts m reference packets
- ❑ Each of receiver records local time of each m referencen packets
- ❑ Receiver i and j exchange all m observations
- ❑ Compute  $\text{offset}[i,j] = 1/m \sum (T_{j,k} - T_{i,k})$

## Clock Skew Problem

- ❑ It takes time to send multiple reference packets
- ❑ Clocks do not have identical heartbeats
  - differences in frequency make them drift
- ❑ After collecting m reference packets, clocks will have drifted
- ❑ Direct averaging the differences will not work
- ❑ Solution:
  - Fit data to a line to estimate clock skew and offset

## Measuring Clock Skew

- Each point is difference of arrival times of reference packet between nodes  $i$  and  $j$
- Clock skew is the slope,  $y$  intercept is the offset

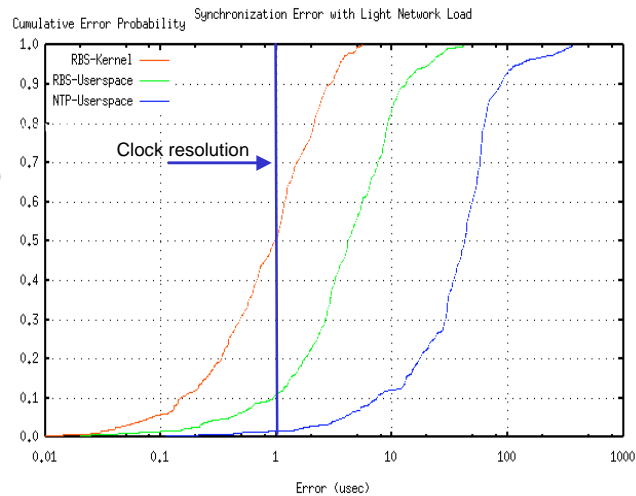


## Comparison with NTP

- Implementation:
  - Compaq IPAQs (running Linux)
  - 11mbit 802.11 PCMCIA cards
- Ran NTP, RBS-Userspace, RBS-Kernel
  - NTP synced to GPS clock every 16 secs
- In both cases, to compute error:
  - IPAQs raises a GPIO line high at an specified local time which is measured with an external analyzer and compared with the correct time

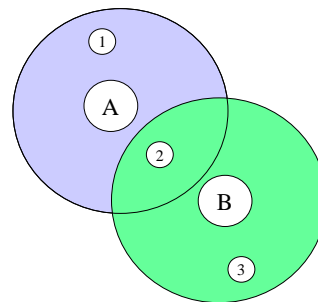
## Comparison with NTP

- Mean error (microsec)  
NTP: 51.18 (+/- 53.30)  
RBS-User: 6.29 (+/- 6.45)  
RBS-Kernel: 1.85 (+/- 1.28)
- Even better results under heavy traffic



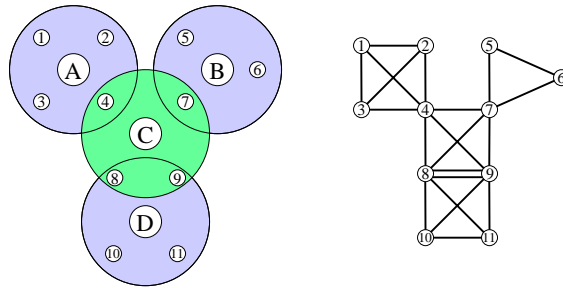
## Multi-hop RBS

- Some nodes broadcast reference packets
- Receivers within transmission range are synced using RBS
- Nodes that hear both reference packets can relate to both time bases
- Event e1 occurred in node 1 at local time t1
  - convert t1 to corresponding time in local clock of node 2 (t2)
  - convert t2 to corresponding time in local clock of node 3 (t3)



## Multi-hop RBS

- ❑ Physical topology easily converted into logical topology
  - links represent possible clock conversions



- ❑ Use *shortest path* search to find a “time route”
- ❑ Edges can be weighted by error estimates

## Clock Synchronization in Sensor Networks

- ❑ Fine-Grained Network Time Synchronization using Reference Broadcasts  
Jeremy Elson, Lewis Girod and Deborah Estrin  
OSDI 2002
- ❑ Optimal and Global Time Synchronization in Sensornets  
Richard Karp, Jeremy Elson, Deborah Estrin, and Scott Shenker  
CENS Technical Report #12 , April 2003
- ❑ Global Clock Synchronization in Sensor Networks  
Qun Li and Daniela Rus  
IEEE Infocom 2004

## Optimal and Global Clock Sync

- ❑ Line fitting in RBS provides an estimate for
  - skew
  - offset
- ❑ Clock synchronization is pairwise

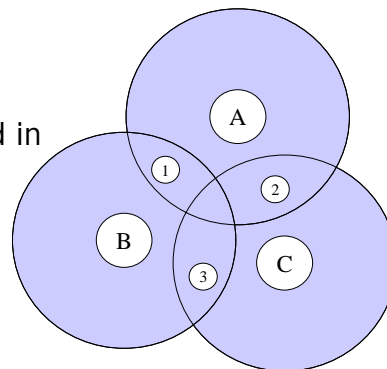
2 problems:

- ❑ Synchronization is not globally consistent
- ❑ Synchronization is not optimally precise

## Global Consistency

- ❑ Event  $e_1$  occurs at node 1 at local time  $t_1$
- ❑ Convert this time to node's 2 clock
  - directly via skew/offset relative to 2
  - indirectly via skew/offset relative to 3, then via skew/offset relative to 2

- ❑ These 2 times represented in node 2's clock may be different!
- ❑ In large networks, several conversion paths exist



## Optimal Synchronization

- ❑ Optimally precise synchronization
  - estimator (offset and skew) that has minimum variance
- ❑ Best fit line in RBS does not lead to an optimally precise estimator
- ❑ Relevant data being ignored
  - information from other broadcast sources
  - time-of-arrival of other receivers

## Global Consistency and Optimal Synchronization

- ❑ Assumptions
  - no clock drift (relaxed later)
  - variance of error known and independent
- ❑ Theorem:  
Minimum-variance estimator is globally consistent
- ❑ Theorem:  
Computing a minimum-variance estimator of the difference between offsets of two nodes is equivalent to computing the distribution of currents in an electric network

## Global Consistency

- ❑ How to compute offsets that are globally consistent?
  - maximum-likelihood estimates for offset under global consistency
- ❑ Theorem:  
The minimum-variance and maximum-likelihood estimates of the offsets coincide
- ❑ Solution to maximum-likelihood estimates is a set of least square equations (assuming independent Gaussian errors)

## Solution Method

- ❑ Least-squares equations can be solved by an *iterative distributed algorithm* which alternately updates the offset estimates
- ❑ Convergence is guaranteed
- ❑ Process can be accelerated by a well chosen initial solution and use of successive over-relaxation

## Comparisons with RBS

- Assume all error variances are 1
- Example 1: every signal is received by every receiver,  $S$  signals
  - Variance =  $2/S$  for both mechanisms
- Example 2: 2-dimensional grid
  - L: Manhattan distance between  $r_1$  and  $r_2$ .
  - Optimal variance =  $O(\log L)$
  - Variance RBS =  $L$

## From Theory to Protocol

- Convert algorithm into practical protocol
  - must consider clock skews
    - can be done assuming drift is much slower
  - nodes runs iterative solution procedure
  - must periodically broadcast information
- Many open questions
  - Broadcast rate to achieve convergence?
  - Energy consumption?
  - Precision of method in practice?

## Clock Synchronization in Sensor Networks

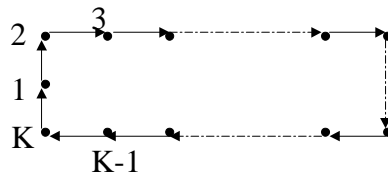
- ❑ Fine-Grained Network Time Synchronization using Reference Broadcasts  
Jeremy Elson, Lewis Girod and Deborah Estrin  
OSDI 2002
- ❑ Optimal and Global Time Synchronization in Sensornets  
Richard Karp, Jeremy Elson, Deborah Estrin, and Scott Shenker  
CENS Technical Report #12 , April 2003
- ❑ Global Clock Synchronization in Sensor Networks  
Qun Li and Daniela Rus  
IEEE Infocom 2004

## Goals and Assumptions

- ❑ Global clock synchronization
- ❑ Coarse precision (implied)
- ❑ For first 2 algorithms
  - No clock drift
  - Clock inter-tick time much *longer* than packet transmission time
  - packet transmission time deterministic
- ❑ For diffusion algorithm
  - local time-scale only (cannot sync to UTC)
  - clock drift is negligible w.r.t. diffusion time-scale

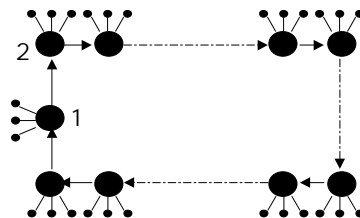
## Node-based Synchronization

- ❑ Starting from node 1, a reference packet is sent along a cycle in network
- ❑ Node 1 computes packet cycle time and informs each node (second cycle)
- ❑ Each node updates its clock based on its position on cycle and total cycle time



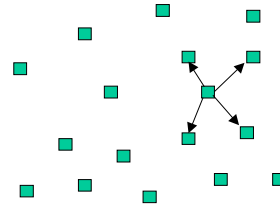
## Cluster-based Synchronization

- ❑ Previous method: requires all nodes in network to participate at same time
- ❑ Solution: divide network into clusters
- ❑ Cluster heads synchronize using previous method
- ❑ Each node in a cluster synchronizes with cluster head (using some other mechanism)



## Diffusion-based Synchronization

- Idea: average clock reading in the entire network
- Use only *local operations* at each node
  - node exchanges data only with neighbors

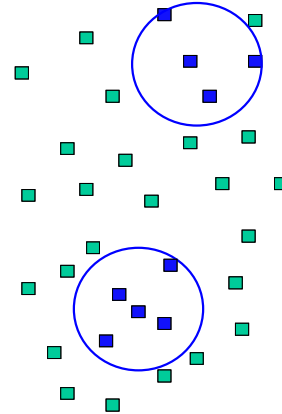


## Diffusion-based Synchronization

- 2 algorithms
  - rate-based diffusion (used in other contexts)
  - average-based diffusion
- synchronous implementation
  - requires node to interact in ordered rounds
  - theoretical proof for convergence and convergence speed (exponential on rounds)
- asynchronous implementation
  - no order required
  - theoretical proof for convergence

## Average-based Diffusion

- Each node periodically and randomly requests clock readings of its neighbors
- Compute average over received readings
- Send back to neighbors average value
- Neighbors set their clock to average value received

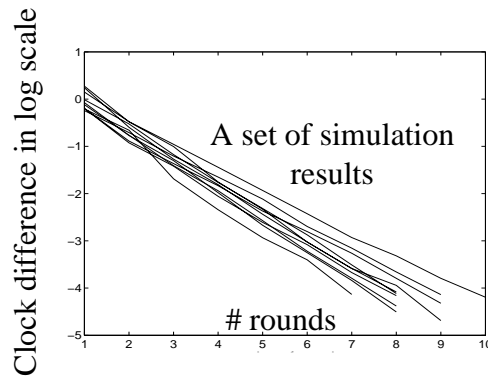


## Asynchronous and Localized Algorithm

- The average operation can be done
  - with any set of neighbors
  - at any time
  - at any order
  - under the mobile environment
  - with failing nodes
- Eventually, all nodes in network will have similar clock readings
  - theoretical proof for convergence

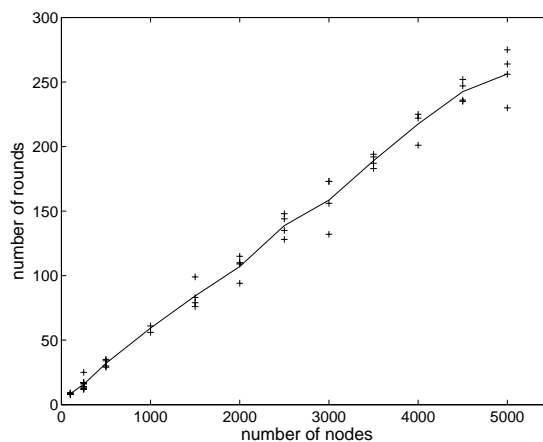
## Convergence Speed

- Maximal clock difference between any two nodes in the network
- Randomly generated network topology
- Sensor field 10x10, tx range 1.5, 400 nodes



## Convergence/# nodes

- Fixed node density
- Change the area of the network field
- # neighbors 15, simulation stops at error 0.1%



## Discussion (1)

- RBS:
  - simple idea, yields high precision
  - allows for *post-facto* synchronization
  - implemented and widely in use
    - several projects are using this mechanism
  - “reference broadcast packet” not a new idea (used in 1992)
  - no full algorithm for global clock sync
    - time routing idea not fully discussed
  - scalability for large networks
- Why not use MAC level sync?
  - 802.11 already has it available

## Discussion (2)

- Optimal and Global Sync:
  - present problems with global RBS
  - nice theoretical results
    - optimal estimator obtained via iterative method
  - feasibility of method in practice
  - convergence time
  - precision achieved in practice?

## Discussion (3)

- ❑ Node-based approach
  - too many problems...
    - it is in the paper!
- ❑ Diffusion-based
  - localized operations
  - theoretical convergence guaranteed (proofs)
  - very coarse precision
  - requires precise low-level approach (e.g., RBS)
  - local time-scale only
  - must be running all times:
    - message overhead?, Energy consumption?
  - not clear why this is better than NTP