# Time Synchronization in Sensor Networks

- *Time Synchronization in Ad Hoc Networks* – Kay Roemer

- *Fine Grained Network Time Synchronization using Reference Broadcasts* – J. Elson, L. Girod, D. Estrin

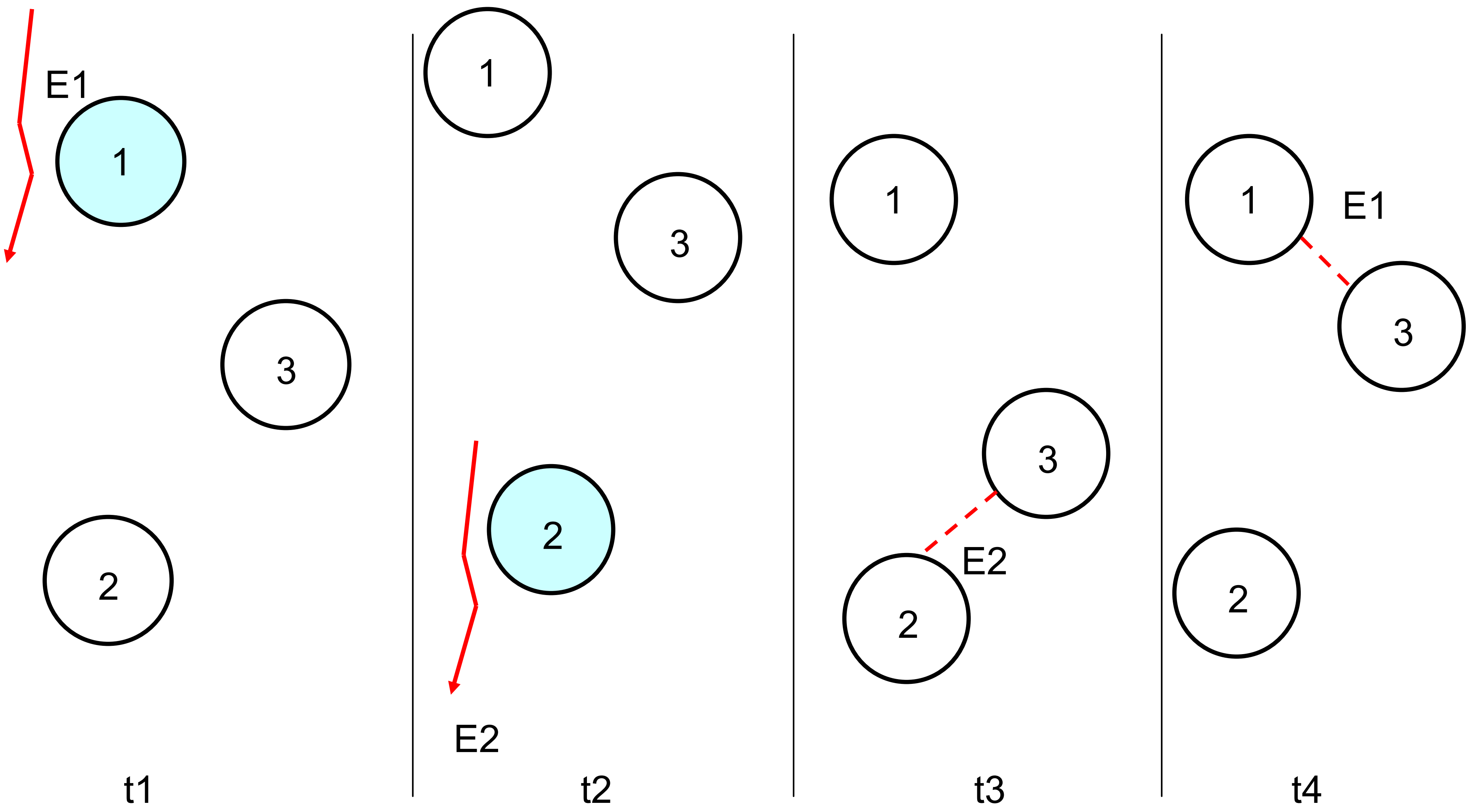
# Need For Time Synchronization

- detect direction of movement of a phenomenon
- form a TDMA radio schedule
- configure a beamforming array
- estimate velocity of a moving phenomenon
- suppress redundant transmissions by recognizing duplicate detection of same event

# Traditional Synchronization Methods

- Exchange messages between the nodes
- Estimate one way latency

# Difficulty – Topology Change



# Difficulty – Non Determinism

Kernel protocol

Context switch, system call

v non-deterministic

Latency = Send Time + Access Time + Propagation Time + Receive Time

time for network interface to receive message from Channel

- *Time Synchronization in Ad Hoc Networks* – Kay Roemer
  - – attacks problem of topology change
  - – determines upper and lower bounds for real time

- *Fine Grained Network Time Synchronization using Reference Broadcasts* – J. Elson, L. Girod, D. Estrin
  - –attacks problem of non determinism in latency
  - – uses physical broadcast to synchronize receivers

# K. Roemer – Time Transformation

- Time stamps generated by unsynchronized clocks
- Transform local time of sender to local time of receiver
- Determines lower and upper bounds for such transformations
- Temporal ordering
  - $t(E_1) < t(E_2)$
  - $t(E_1) > t(E_2)$
  - Unsure

# Clock Drift

$$1 \text{ à } \text{ú} \text{ ô } \quad \frac{dC}{dt} \quad \text{ ô } \quad 1 + \text{ú}$$

- ú is of the order of $10^{-6}$
- different computer clocks have different drifts
- Approximate as differences to get
  - $(1-\rho)\Delta t \le \Delta C \le (1+\rho)\Delta t$
  - $\dfrac{\Delta C}{1+\rho} \le \Delta t \le \dfrac{\Delta C}{1-\rho}$

# Time Transformation

$$\Delta C$$

Node 1 clock

$$\downarrow$$

$$\left[\frac{\Delta C}{1+\rho_1}, \frac{\Delta C}{1-\rho_1}\right]$$

Real Time

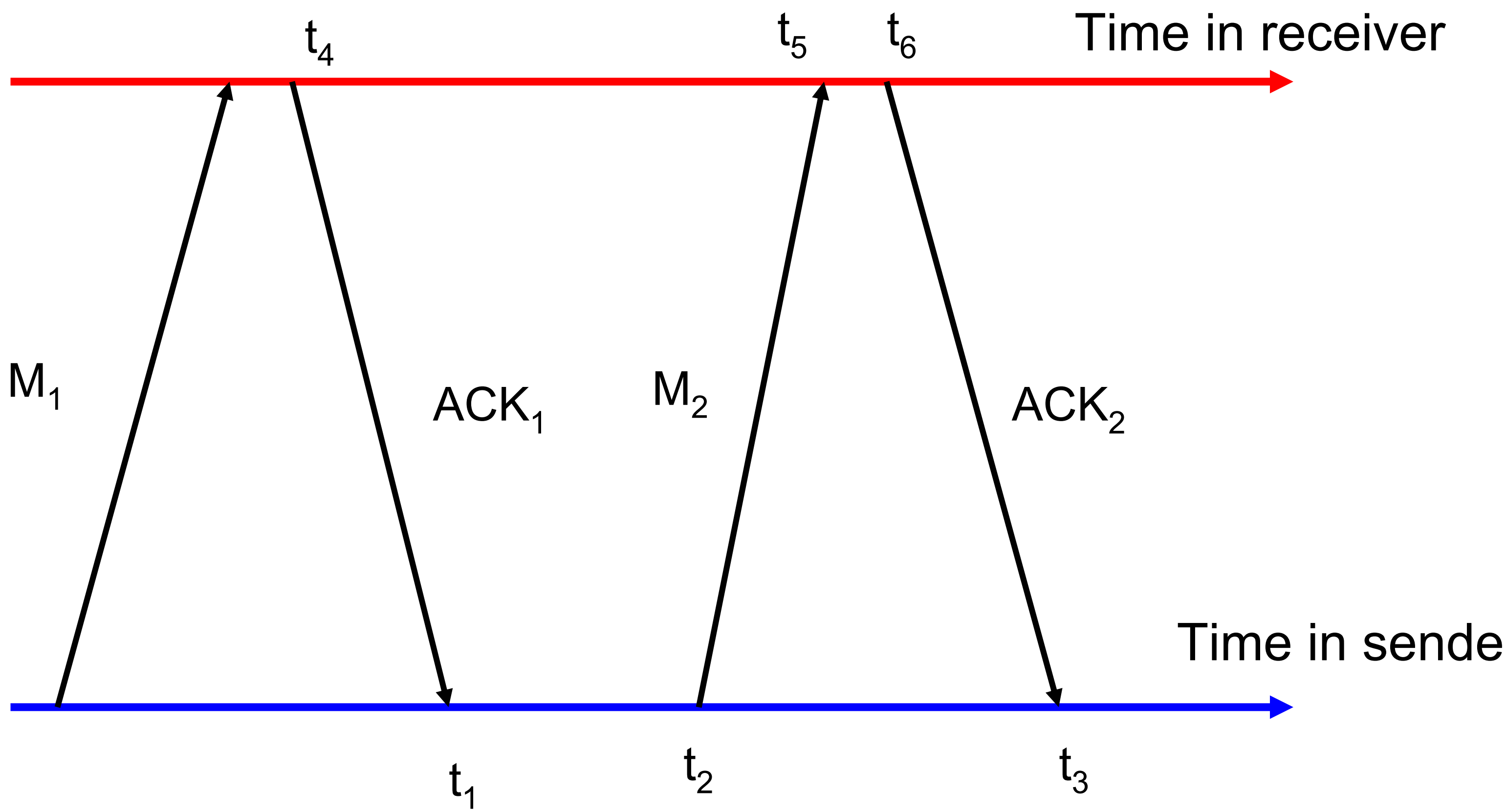
$$\downarrow$$

$$\left[\Delta C\frac{1-\rho_2}{1+\rho_1}, \Delta C\frac{1+\rho_2}{1-\rho_1}\right]$$

Node 2 clock

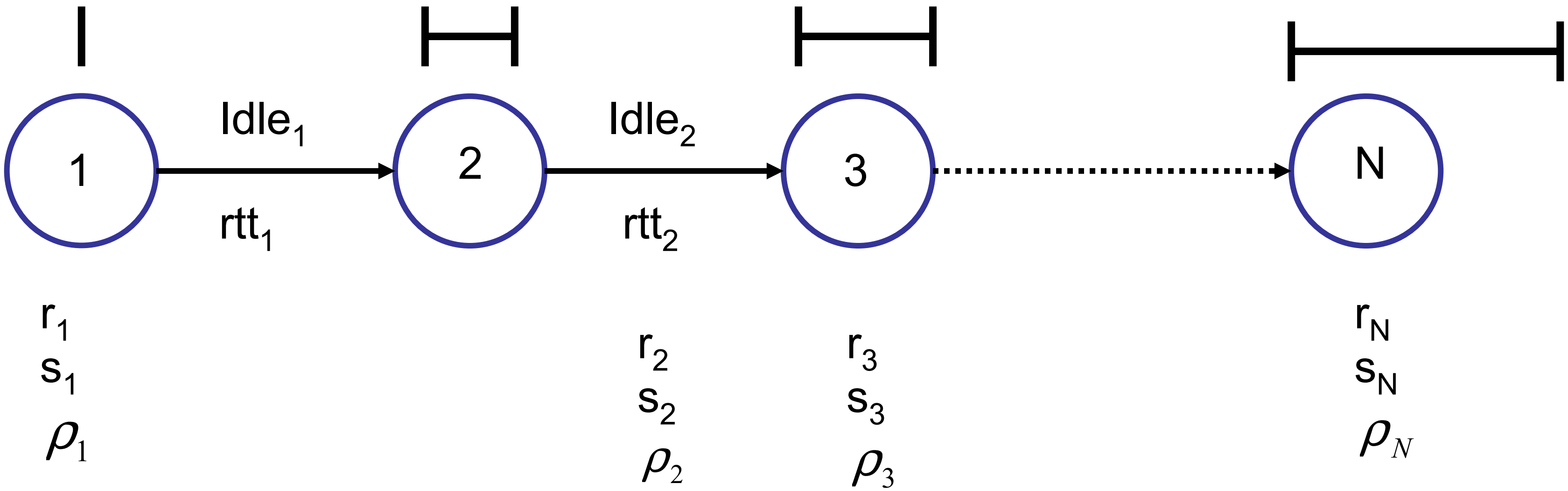
# Message Delay Estimation

$$0 \le d \le (t_5 - t_4) - (t_2 - t_1)\frac{1-\rho_r}{1+\rho_s}$$

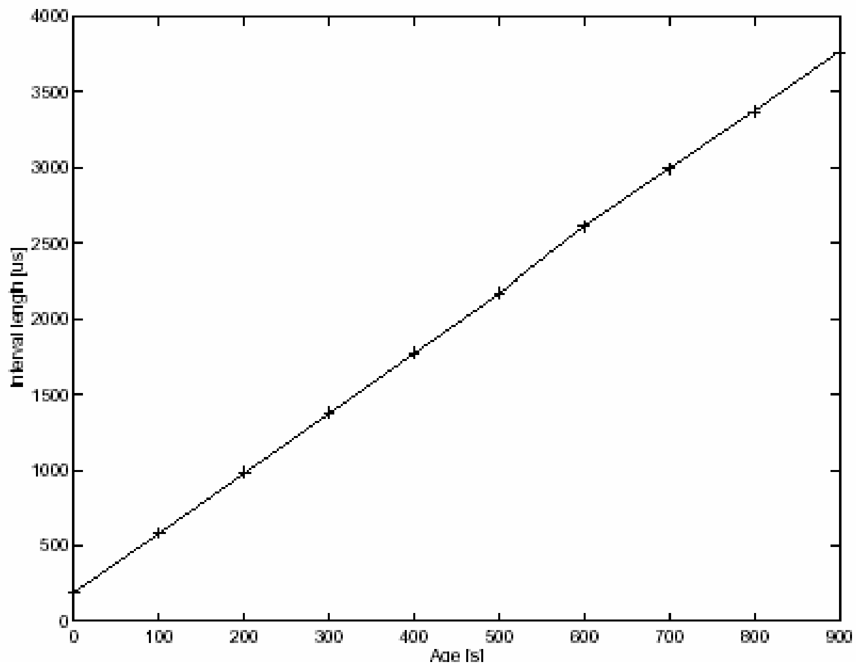


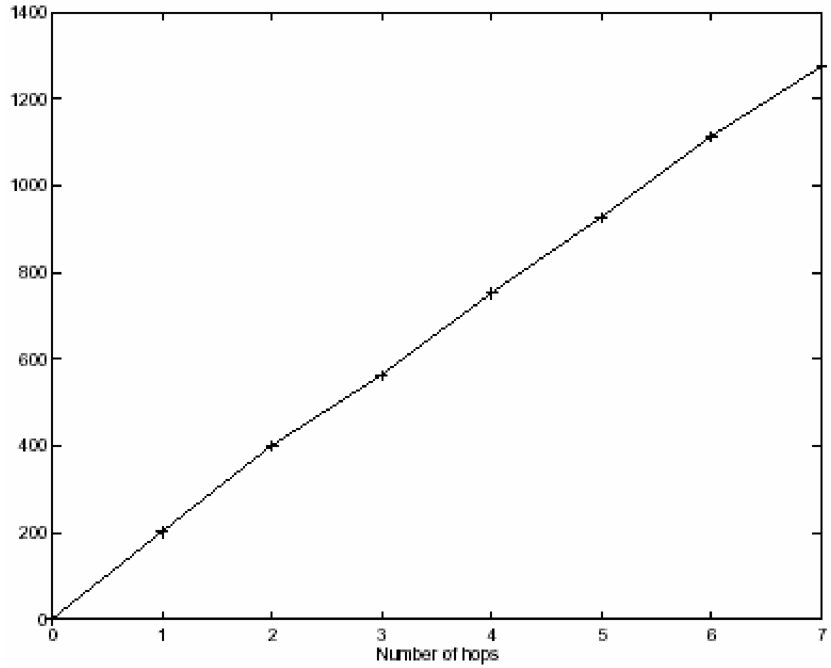

$$0 \le d \le (t_3 - t_2) - (t_6 - t_5)\frac{1-\rho_s}{1+\rho_r}$$

# Message Flow



# Accuracy

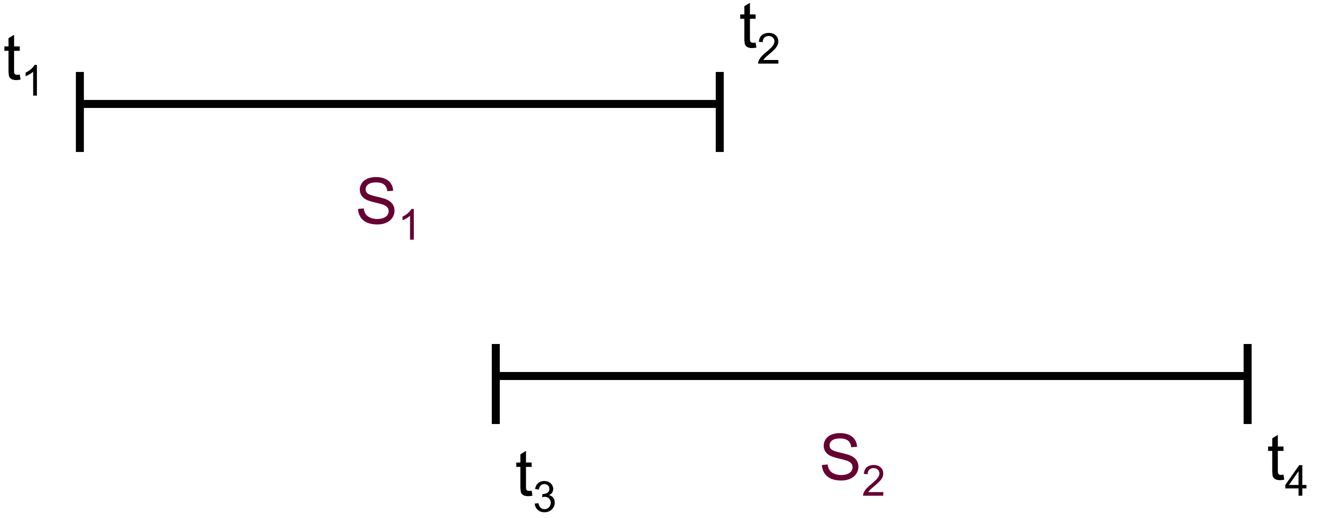



Age

number of hops

# Interval Arithmetic



# Summary

- Handles network partitioning
- Does not need special topologies
- Low message overhead
- Improvements based on probability distribution of uncertainty possible

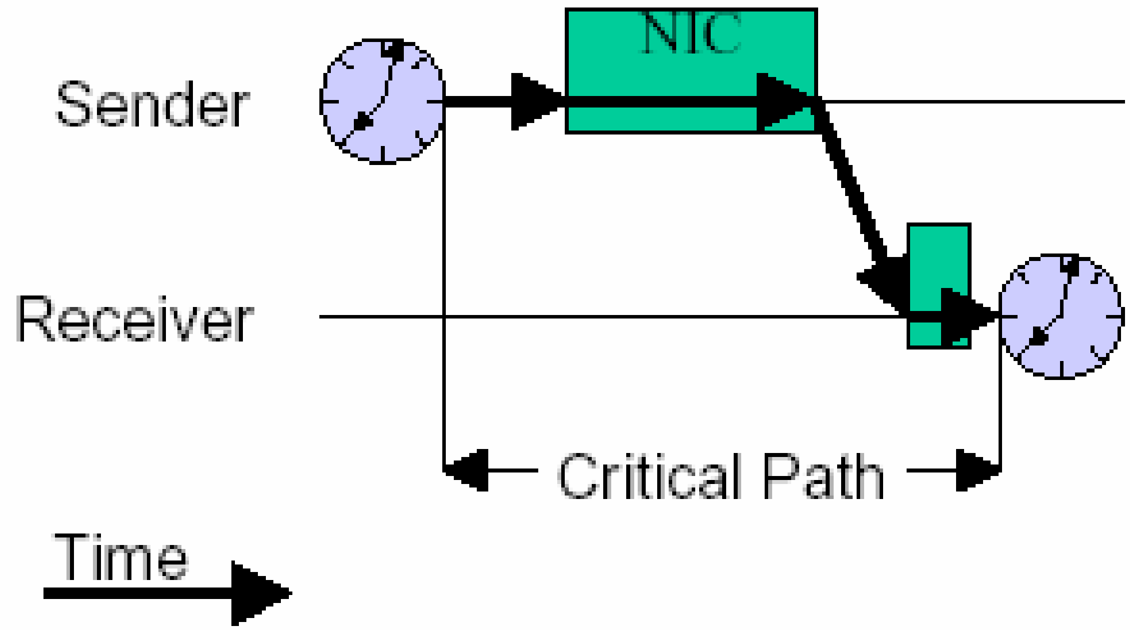
# J. Elson *et al.* – Reference Broadcasts

- Nodes send reference beacons to neighbors using physical-layer broadcasts
- Receivers use arrival time as a point of reference to compare clocks
- Multi-hop synchronization can be achieved

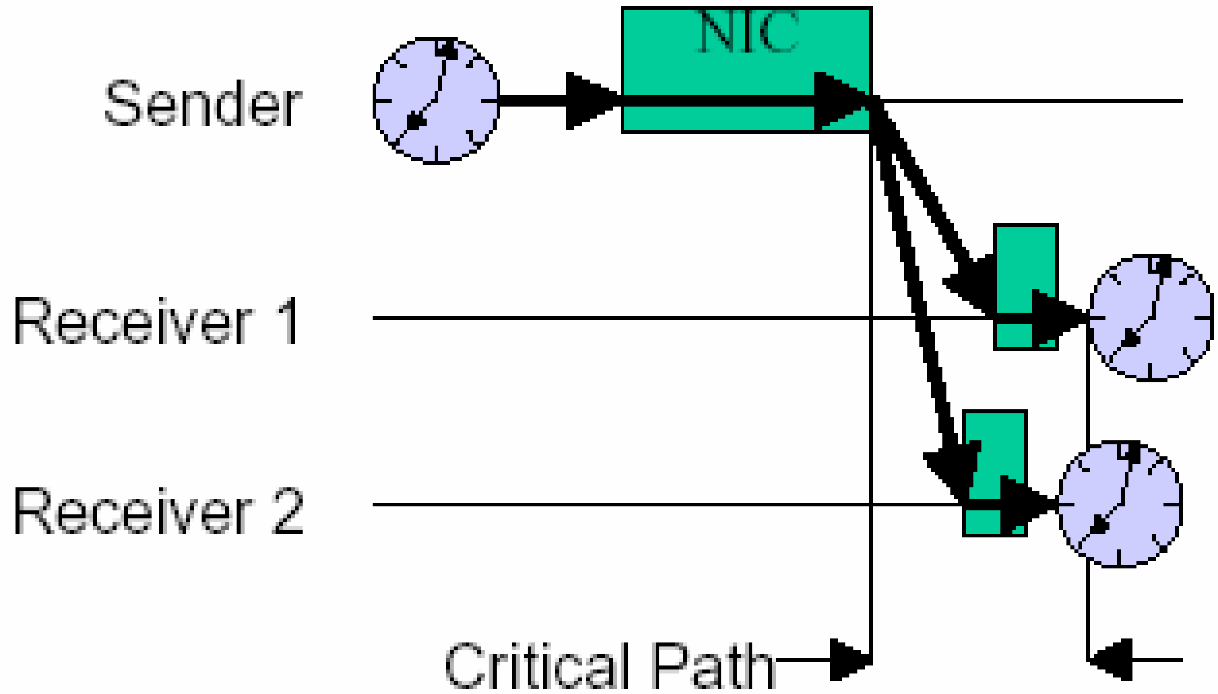
Limitation:

Need physical broadcast channel

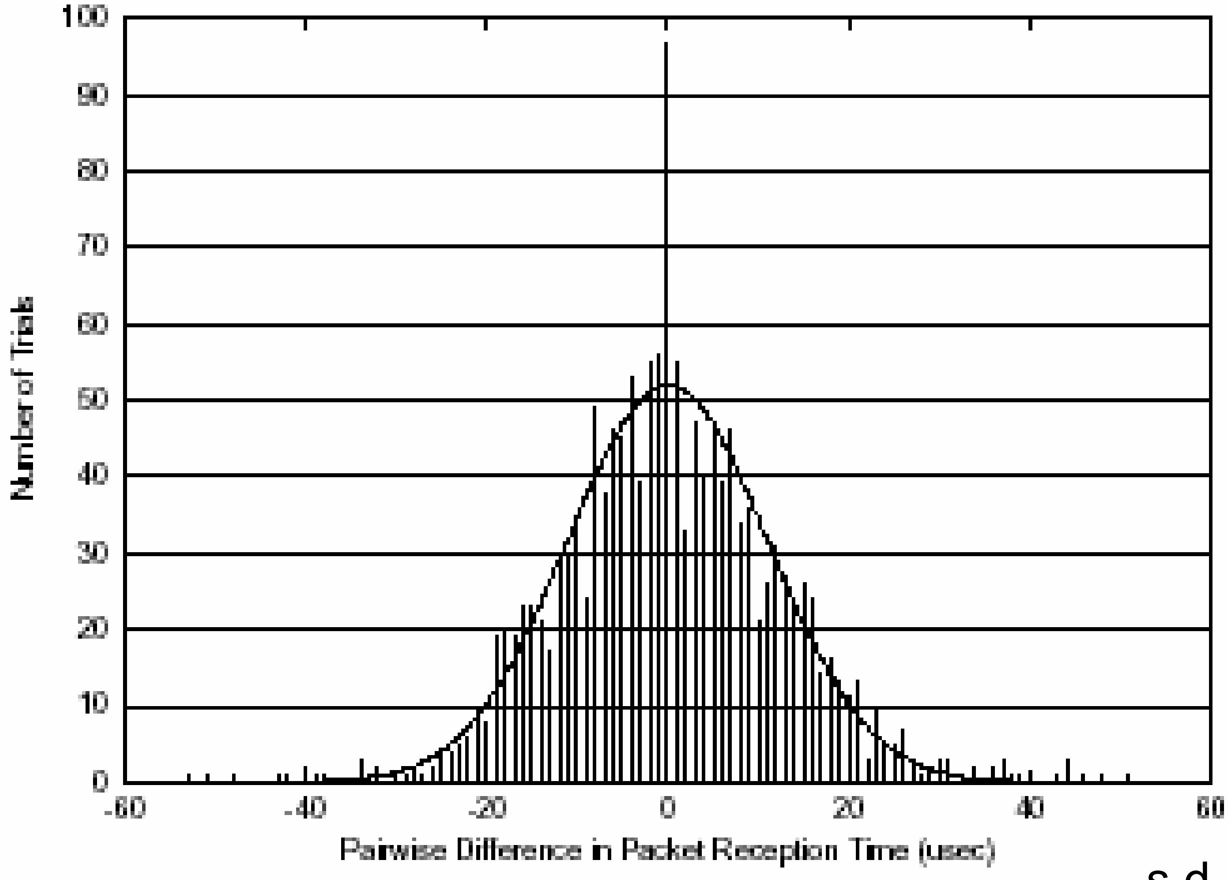
# Traditional Synchronization



# RB Synchronization



# Inter-receiver Phase Offset




s.d. = 11.1 microsec

Bit time is 52 micro-seconds

# Estimation of Phase Offset - Algorithm

- A transmitter transmits *m* reference packets
- Each of the *n* receivers records the time that the reference was observed according to the local clock
- Receivers exchange observation
- Each pair of receivers compute the phase offset by averaging the *m* readings

Group Dispersion: max. of phase error between all receiver pairs

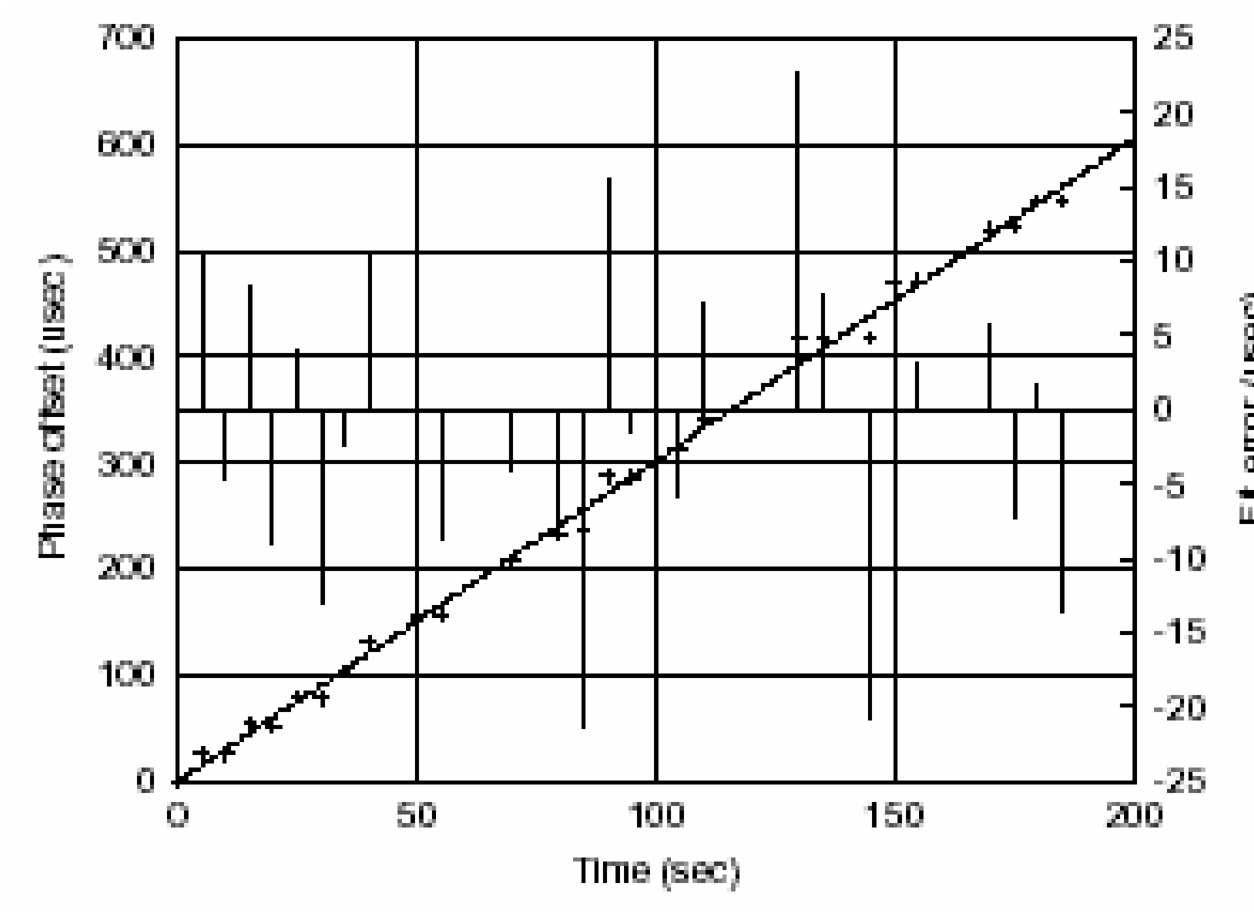
# Estimation of Clock Skew

<u>Oscillator:</u>

- Accuracy
- Stability

- Assume frequency is stable over small time window
- Use least-squares linear regression
  - Slope gives frequency offset
  - Intercept gives phase offset

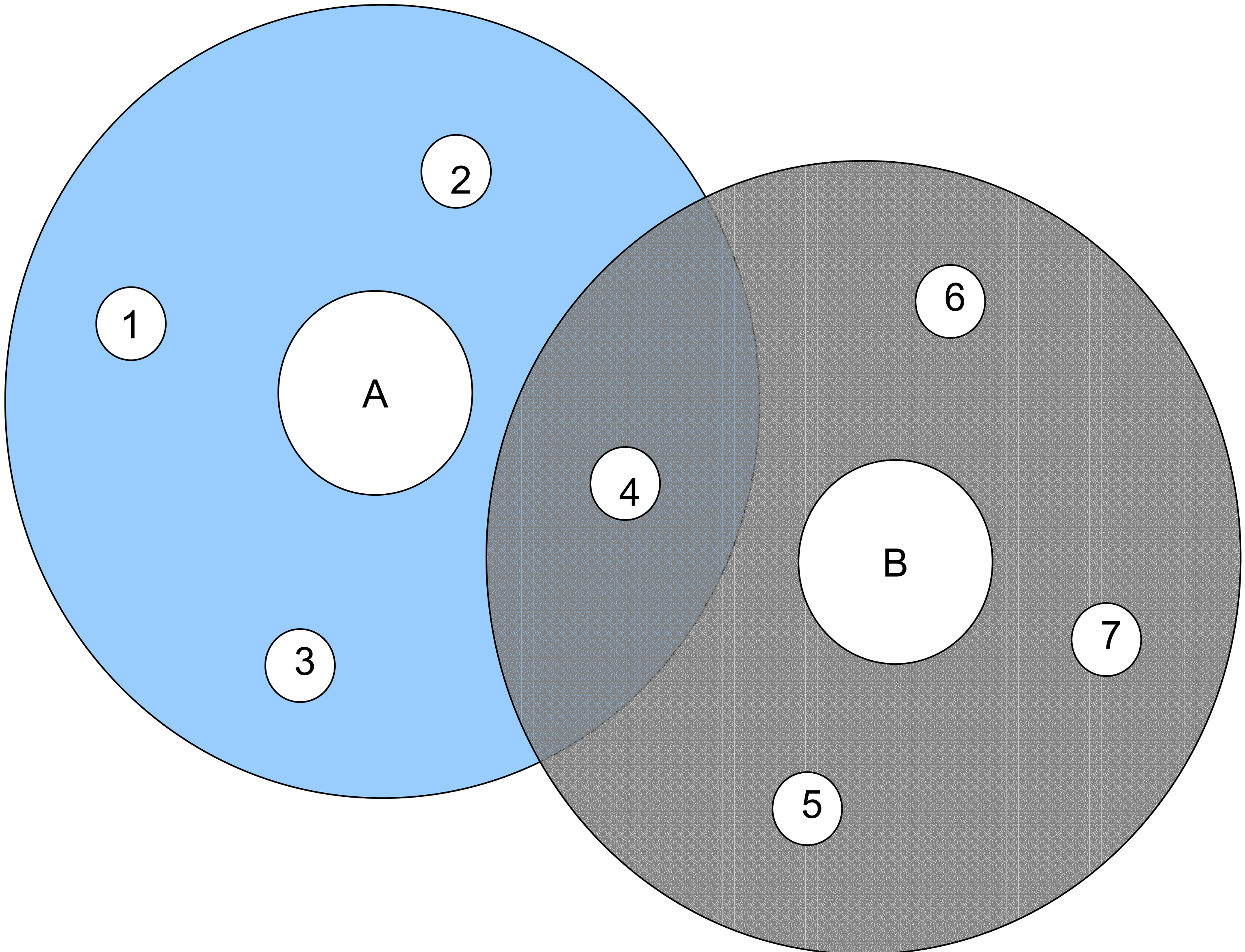
# Estimation of Clock Skew - Experiment



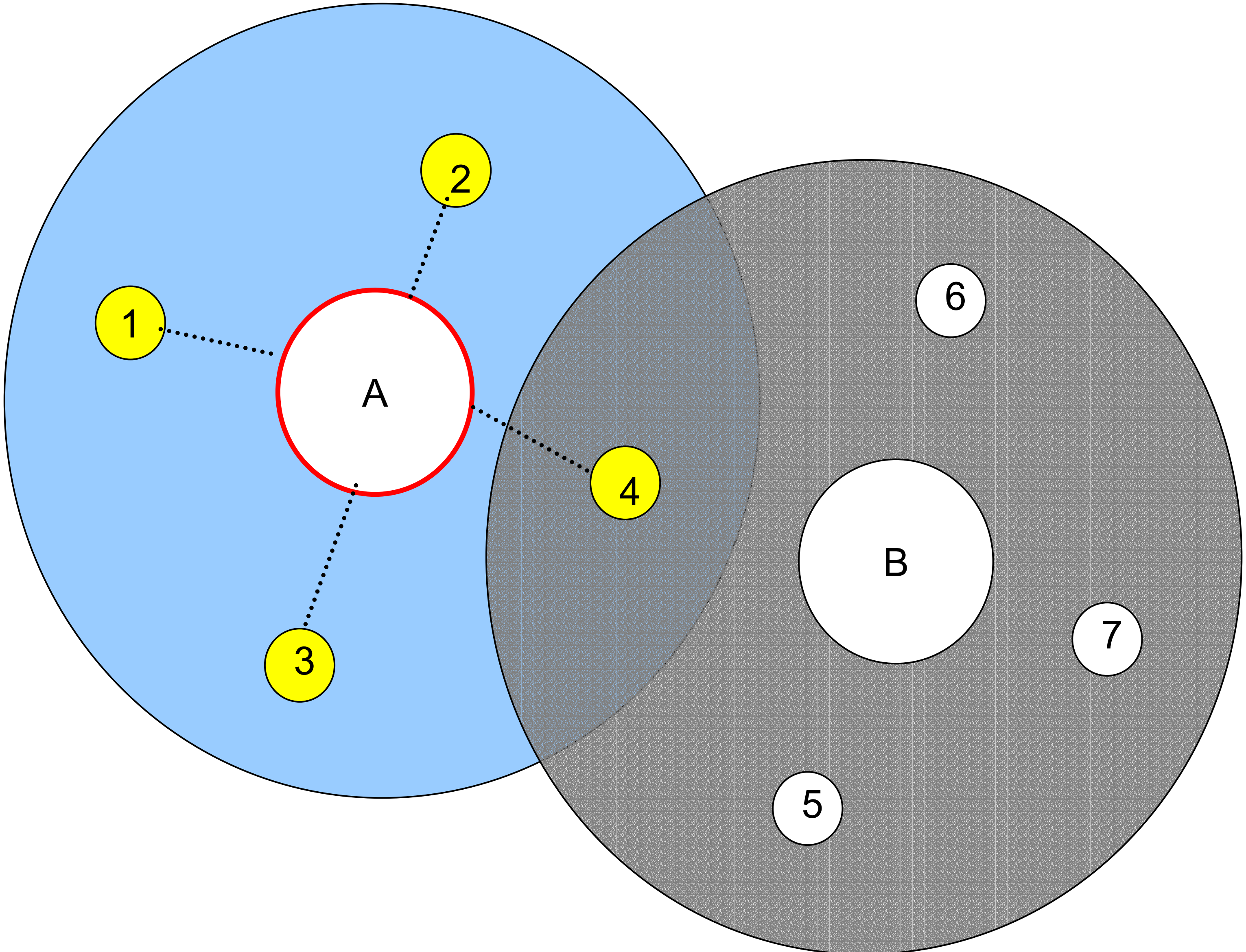
# Basic RBS Summary

- Removes largest sources of nondeterministic latency
- Multiple broadcasts decrease error, *post-facto synchronization* possible
- Outliers can be handled
- Local timescales can be constructed (absolute synchronization is a natural extension).

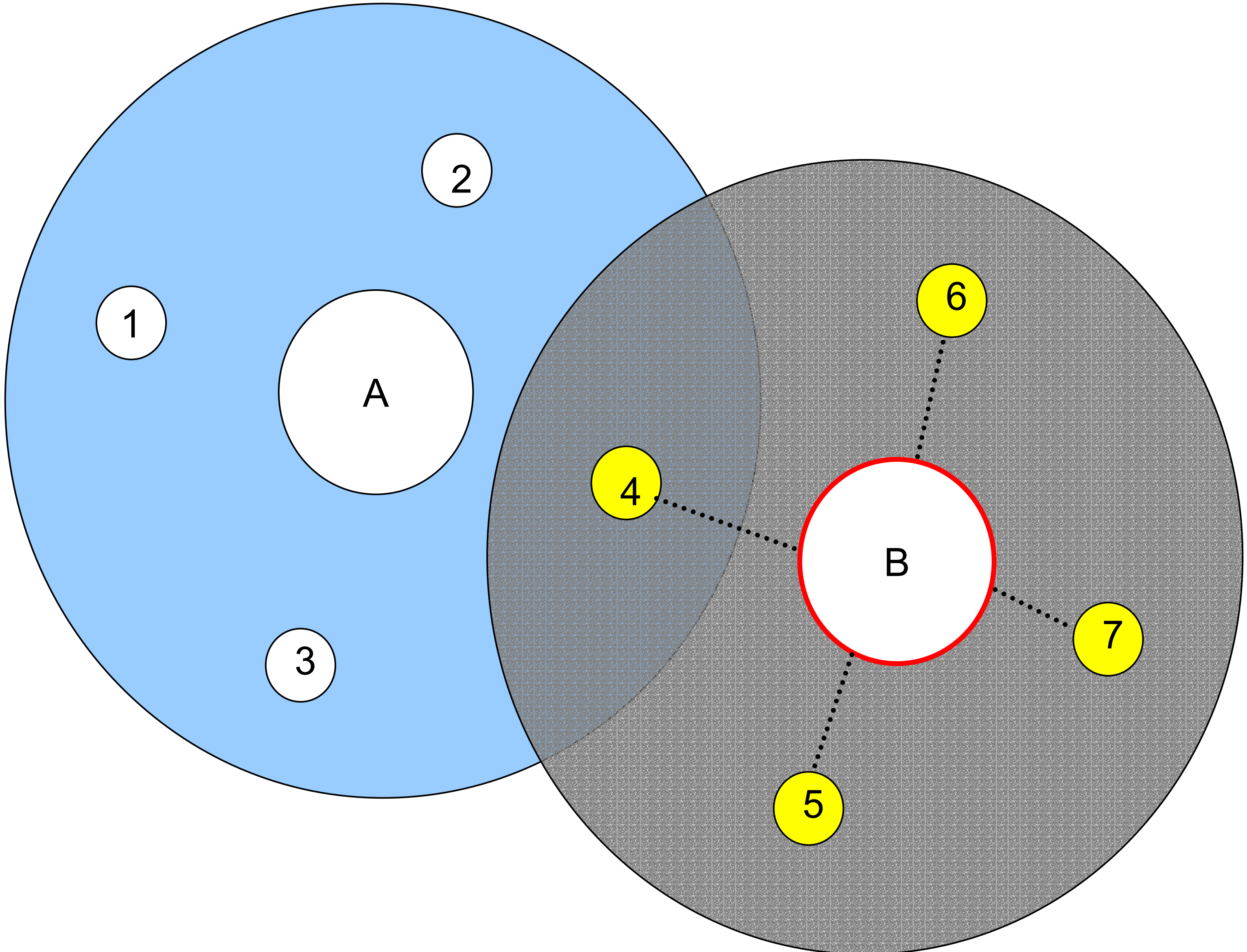
# Multihop Time Synchronization



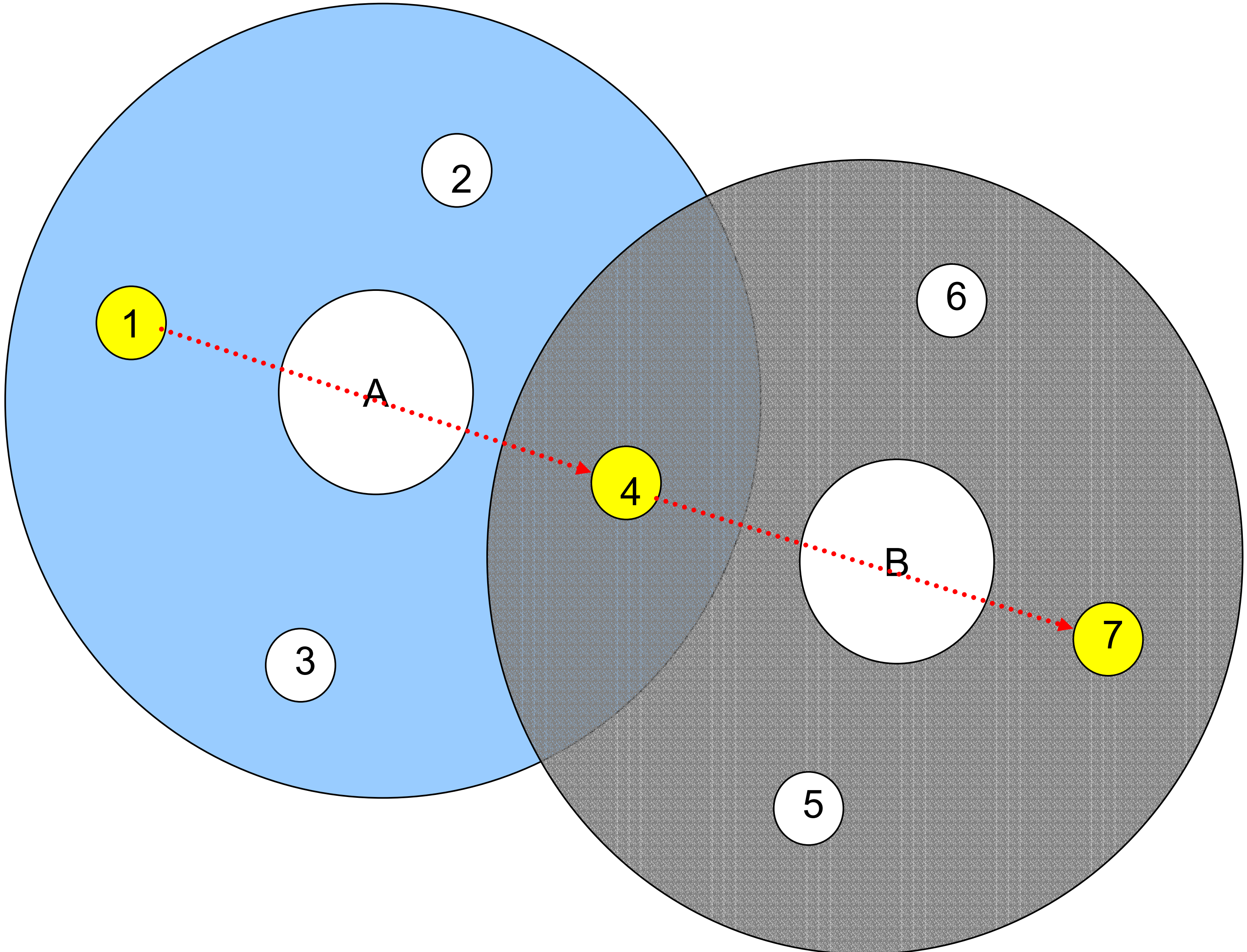
# Multihop Time Synchronization



# Multihop Time Synchronization



# Multihop Time Synchronization



# Summary

- RBS synchronizes a set of receivers with one another
- No explicit timestamp needed
- Largest sources of non determinism in latency removed
- Residual error is Gaussian (experimental observation)
- *Post-facto* synchronization possible