A Hierarchical Multiple Target Tracking Algorithm for Sensor Networks

Songhwai Oh and Shankar Sastry
EECS, Berkeley

Nest Retreat, Jan. 2004

Outline

- Survey of Multiple Target Tracking Algorithms
- Tracking Multiple Objects in Sensor Networks
- Sensor Network Model
- Algorithm Overview
- Some Simulation Results

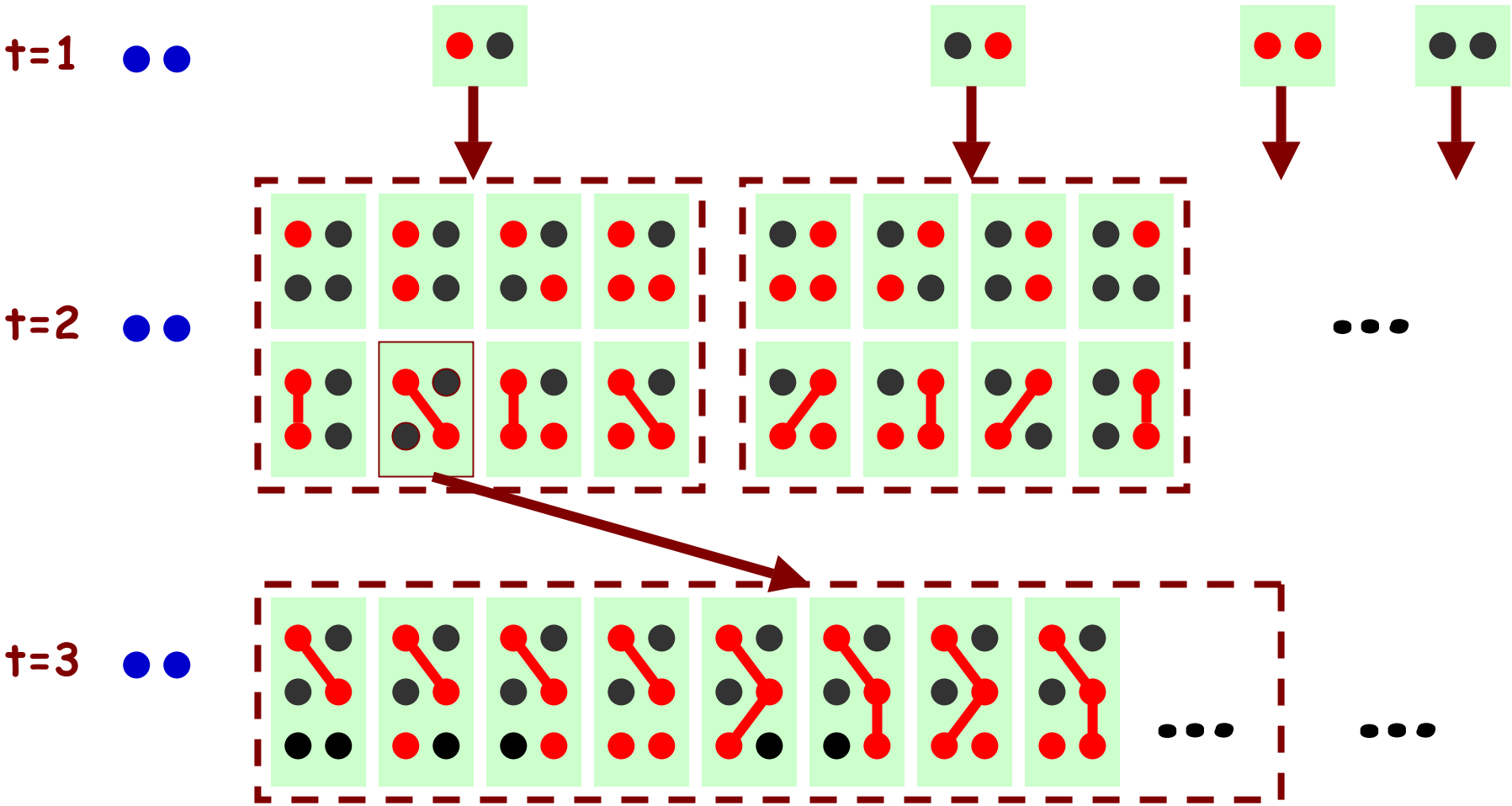
Multiple Target Tracking (MTT)

- Applications
  - Surveillance, computer vision, signal processing, etc.
- General setup (Sittler, 1964)
  - A varying number of indistinguishable targets
    - Arise at random in space and time
    - Move with continuous motions
    - Persist for a random length of time and disappear
  - Positions of targets are sampled at random intervals
  - Measurements are noisy and
    - Detection probability < 1.0 (missing observations)
    - False alarms
  - Goal: For each target, find its track!!!

MTT Algorithms

- Require solutions to
  - Data association: find a partition of observations such that each element of a partition is a collection of observations generated by a single target or clutter
  - State estimation: for each time, estimate the position of each target
  - "Chicken-and-Egg" problem
- Existing Algorithms
  - MHT (Multiple Hypothesis Tracker)
  - JPDAF (Joint Probabilistic Data Association Filter)
  - MTMR, PMHT, etc.

MHT (Multiple Hypothesis Tracker) (1)



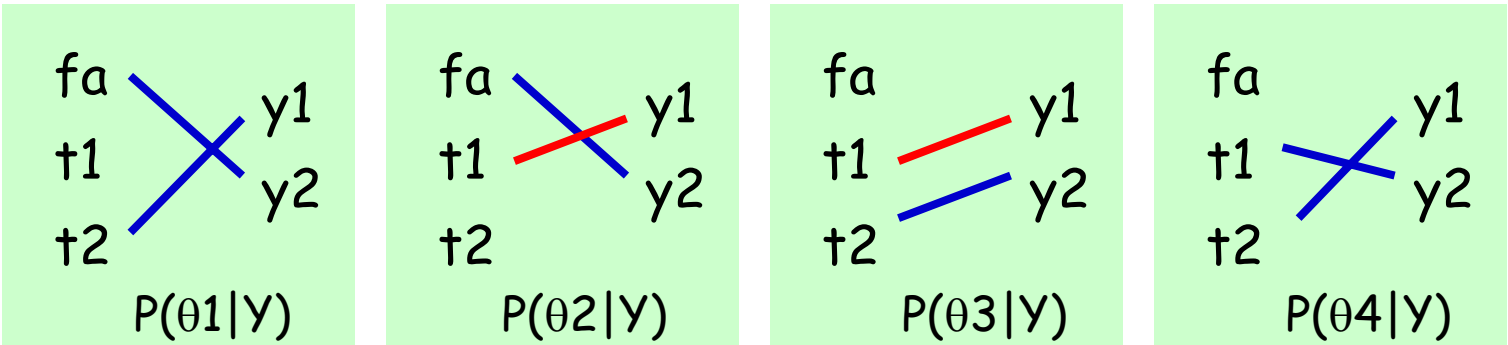
Search for a hypothesis with the highest likelihood

MHT (2)

- Pros
  - Track unknown number of targets
  - Track initiation and termination
  - Optimal (?)
- Cons
  - Exponential complexity
  - Heuristics:
    - Gating, Pruning, N-scan-back, clustering
    - Can deteriorate performance under dense environment or low detection probability
  - Running time and memory requirement not known in advance

JPDAF (Joint Probabilistic Data Association Filter)

- A **fixed** number of targets
- At each time, weight the **latest** observations (y1,y2) with all known tracks (t1,t2). E.g. $P(y1|t1) = P(\theta 2|Y)+P(\theta 3|Y)$
- Track of a target is estimated by weighted sum of conditional expectations. E.g. $E(x(t1)) = \sum E(x(t1)|yi)\, P(yi|t1)$



...

- More efficient than MHT
- But JPDAF is **suboptimal** and
  - prone to make erroneous associations
  - number of targets are fixed
  - can't initiate or terminate targets

Hardness of Data Association

- Combinatorial optimization approach (e.g. MHT)
  - NP-hard
  - Because the multidimensional assignment problem is NP-hard
- Sequential Bayesian approach (e.g. JPDAF)
  - NP-hard (at each time)
  - Because finding permanent of 0-1 matrix is #P-complete
- So we can't expect to solve a data association problem only with local information

MTT in Sensor Networks (Requirements)

- Autonomous
  - Unknown number of targets
  - Track initiation and termination
  - Can't use JPDAF, MTMR, PMHT
- Low computation and memory usages
  - Can't use MHT
- Robust against
  - transmission failures
  - communication delays
- Scalable
- Low communication load

MTT in Sensor Networks (Our Approach)

- Autonomous
- Low computation and memory usages
- Robust against failures and delays
- Scalable
- Low communication load

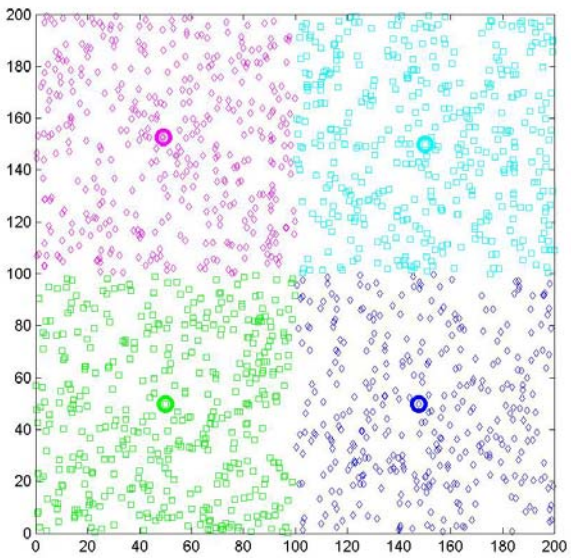
- MCMC data association

- MCMC data association
  - Optimization algorithm (stochastic search)
  - Solution space: constrained partitions of observations
  - Search for a partition with maximum posterior

- Hierarchy
- Local data fusion

Sensor Network Model

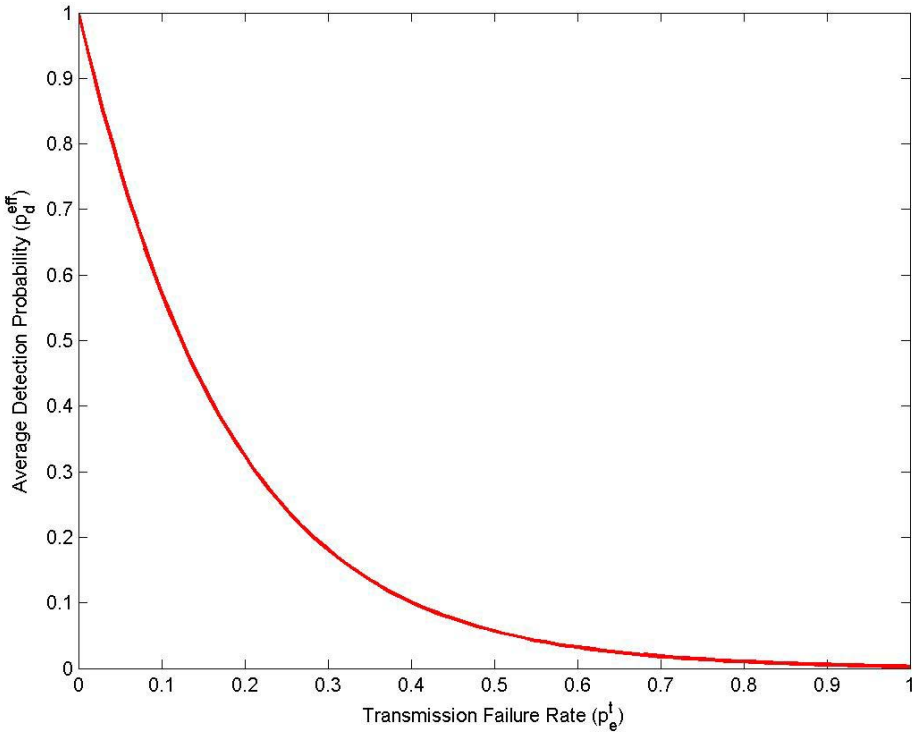
- (few) Supernodes (e.g. Stargate)
  - More computational power
  - Longer communication range
- (many) Regular nodes
  - Form a tracking group around the nearest supernode



- Region: 200x200
- 1600 nodes
  - 4 supernodes
- Rt=10, Rs=5

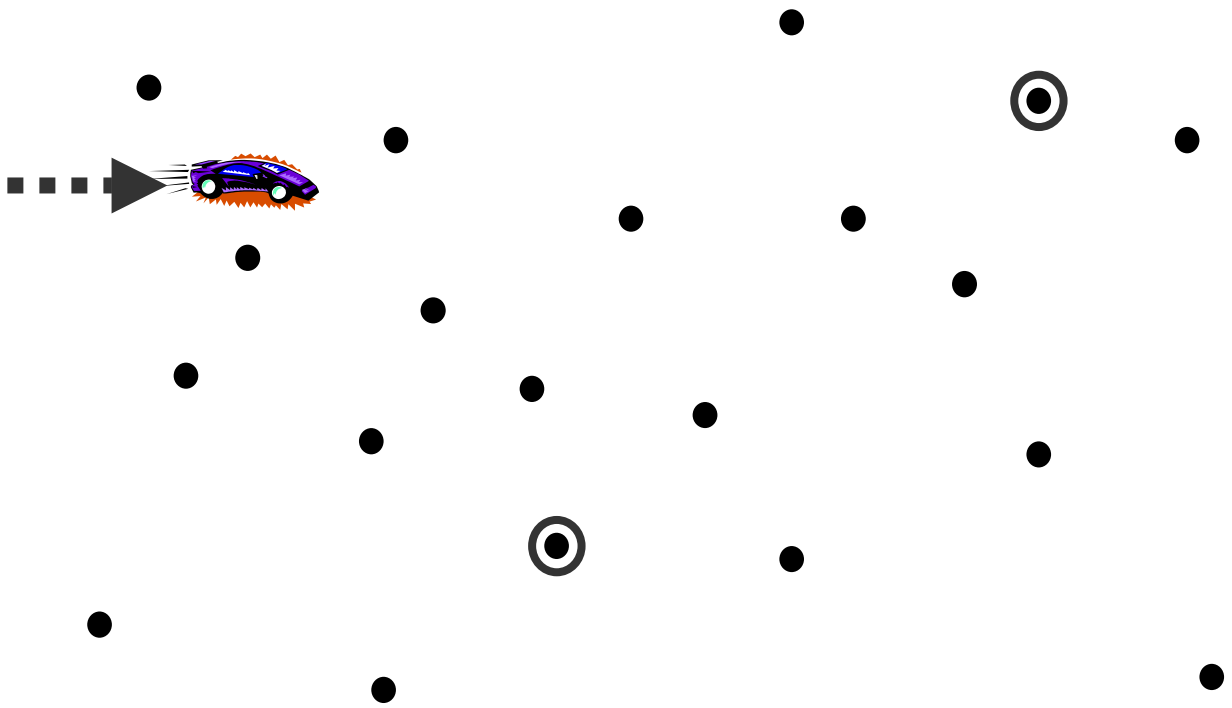
Transmission Failures

- Assumption: A transmission failure between a pair of nodes is independent and identically distributed
- Transmission failures are detection failures

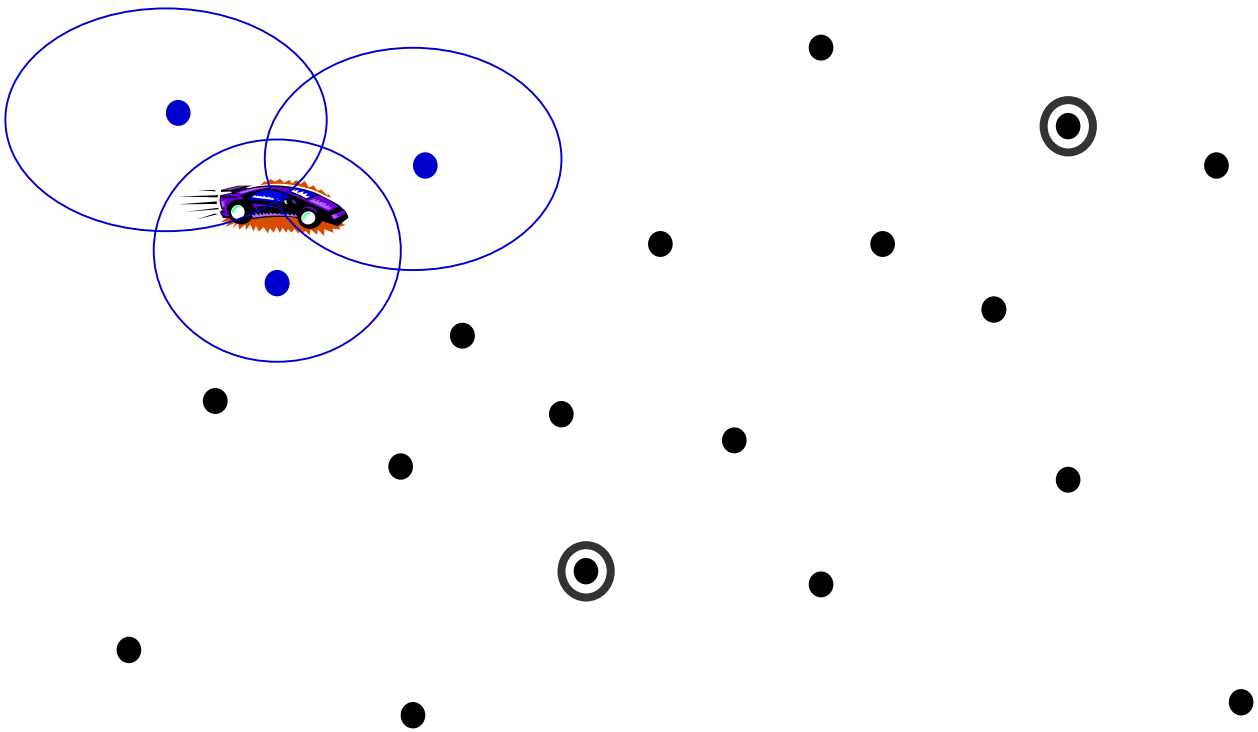


- Transmission failure vs. effective probability of detection
- Probability of detection = 1 (at each sensor)

Algorithm Overview (1)

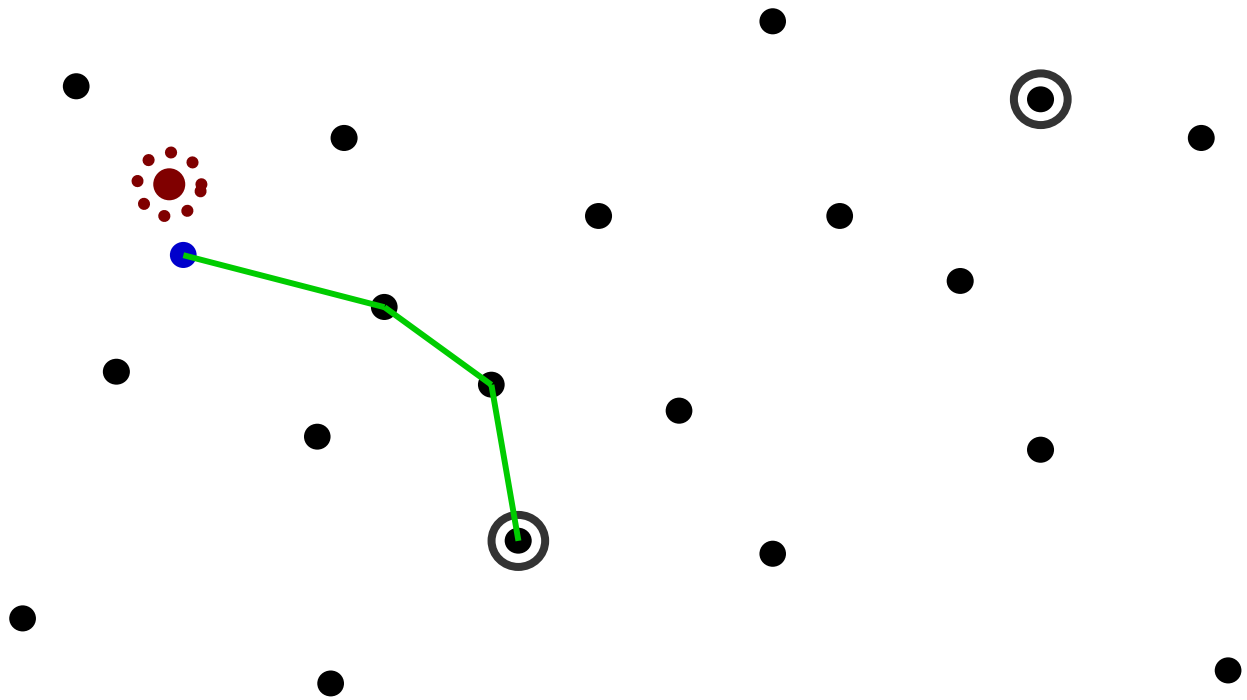


Algorithm Overview (2)



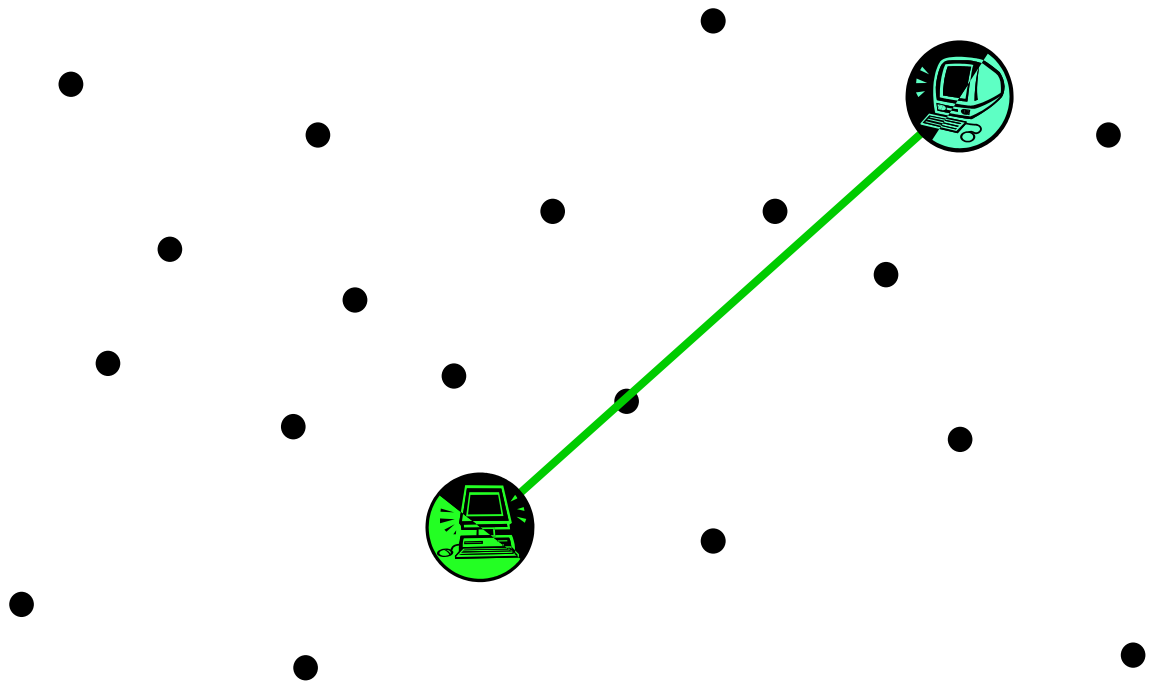
- Fuse observations from sensing neighborhood
  - E.g. weighted sum

Algorithm Overview (3)



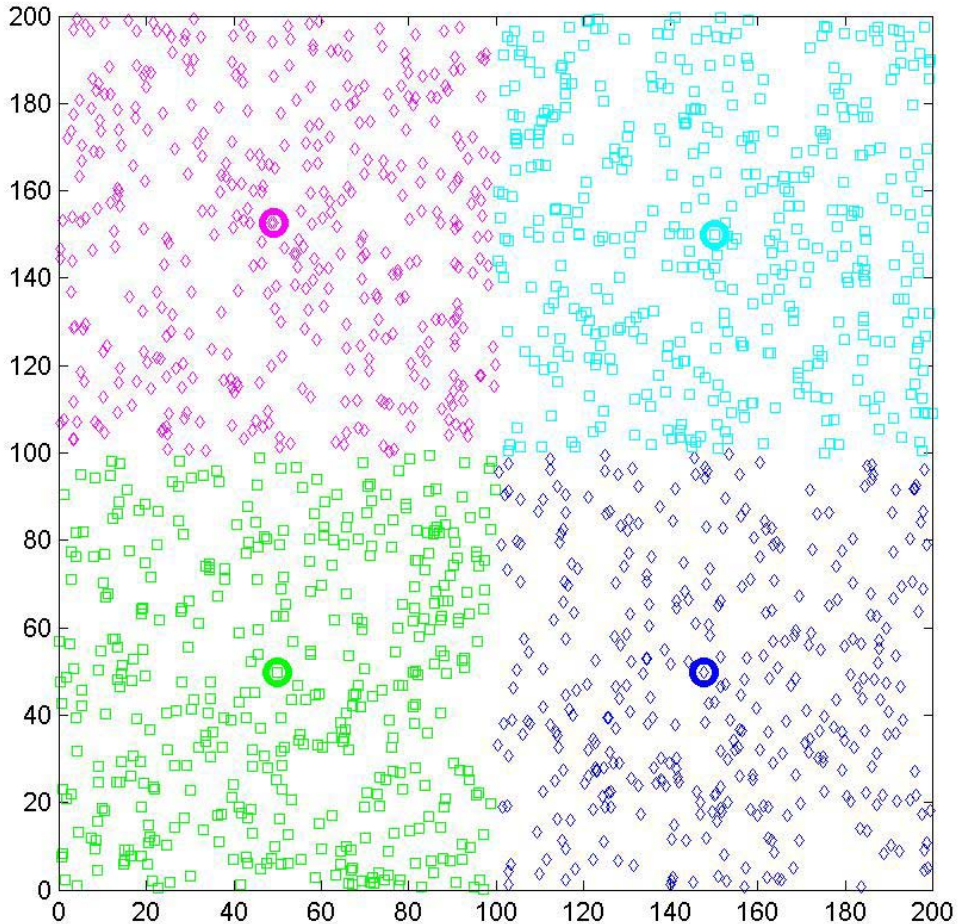
- Forward fused observations to its supernode
  - shortest-path routing
- Some observations get dropped or delayed

Algorithm Overview (4)



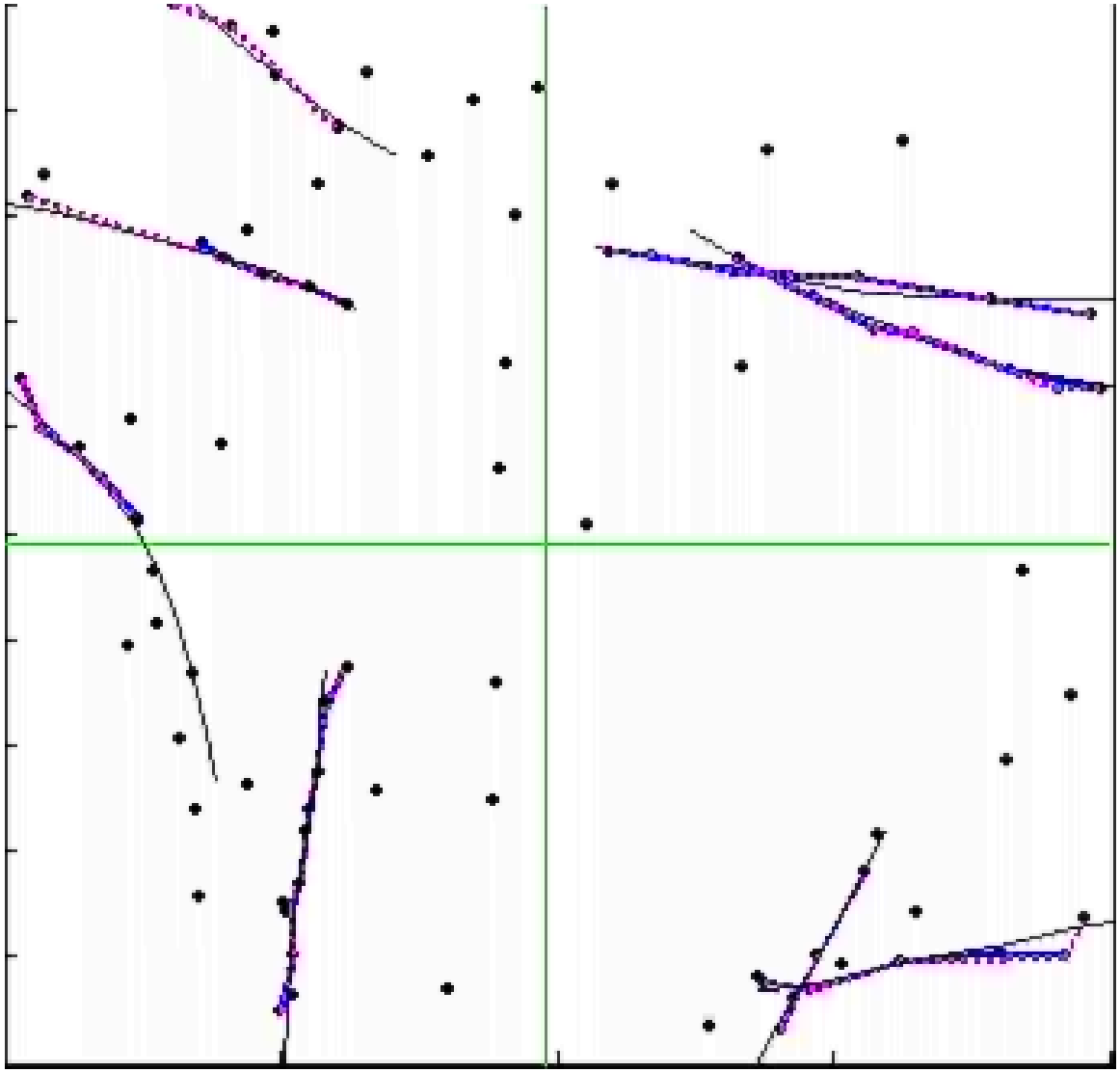
- Supernode:
  - Attach new observations into its observation window Y
  - Run MCMC data association on Y
  - Exchange track information with neighboring supernodes

Simulation



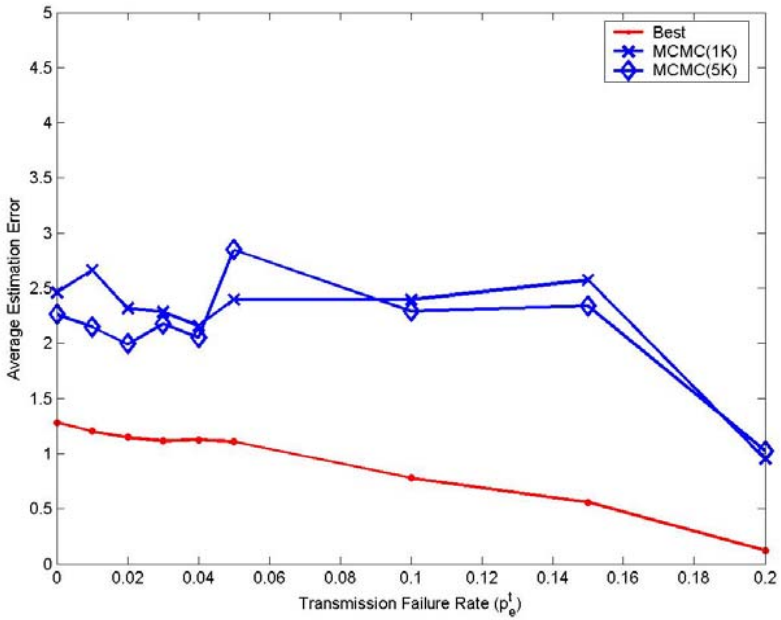
- Region: 200x200
- 1600 nodes
  - 4 supernodes
- Rt=10, Rs=5

Simulation

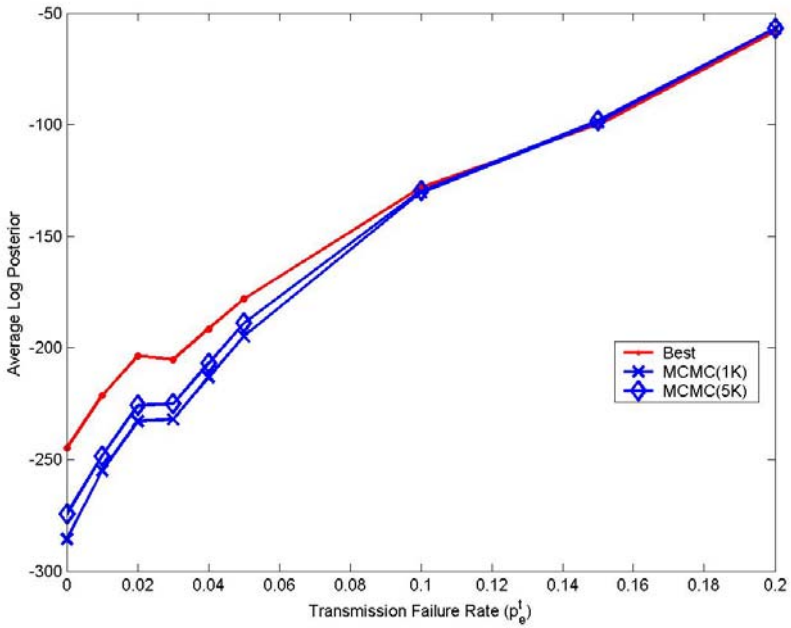


Simulation: Transmission Failures

- Average Estimation Error

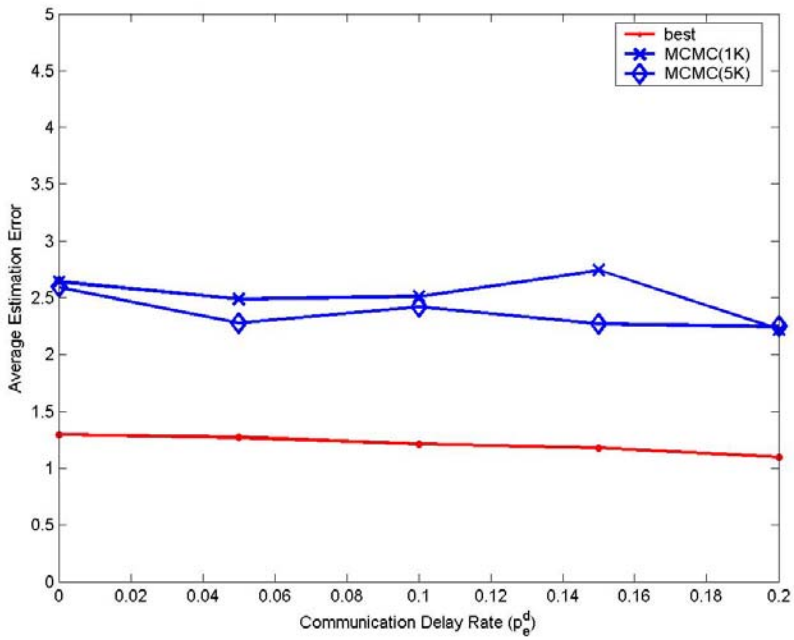


- Average Log Posterior

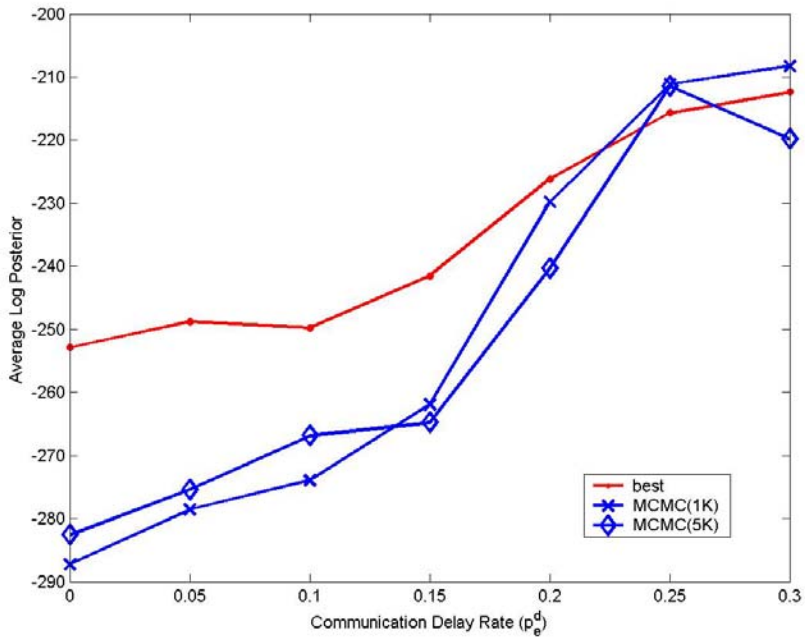


Simulation: Communication Delays

- Average Estimation Error



- Average Log Posterior



Conclusions

- Presented a hierarchical multiple target tracking algorithm for sensor networks, i.e.,
  - Autonomous
  - Low memory and computation requirement (predictable running time)
  - Robust against transmission failures and communication delays
  - Scalable
- Future works:
  - Merge tracks hierarchically
  - Find methods to reduce transmission failure rates
  - Use a better sensor network model
  - Test on a real sensor network