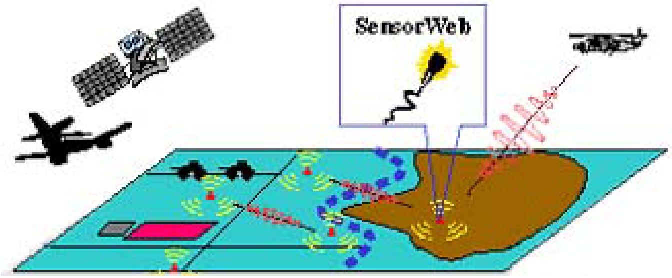
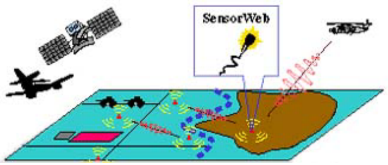


Distributed Algorithms for Estimation Tasks in Sensor Networks

Maurice Chu and Sanjoy Mitter

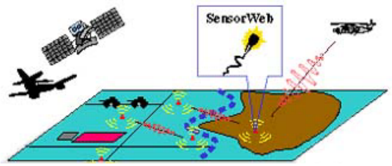
SensorWeb MURI Review Meeting

June 14, 2002



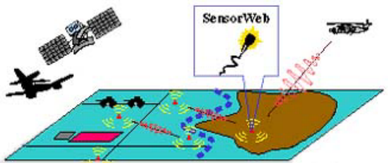
SensorWeb and Beyond

- Data Organization, Information and Estimation [IT-2]
- The Role of Communication
- The Gluing Together of Systems
  - The Central Role of the Cohomology Group
- Dynamic Gluing



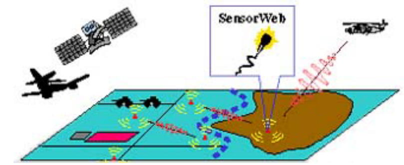
Distributed Algorithms Outline

- Estimation in Sensor Networks (RCA-5&6)
- Information-Driven Sensor Querying (IDSQ) Algorithm
- Problem Formulation
- Distributed Algorithm
- Discussion and Future Work



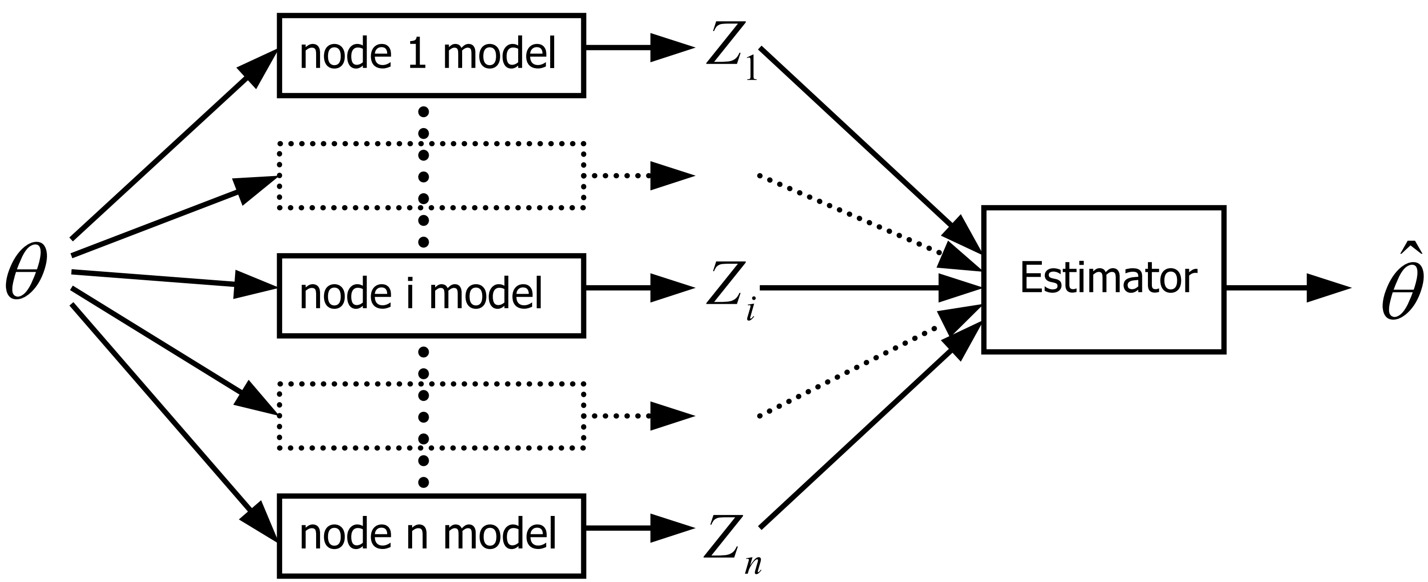
Nature of Information in Sensor Networks

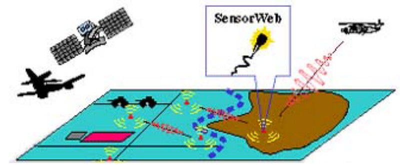
- Measurement Types
  - Acoustic – amplitude, direction of arrival
  - Seismic
  - Magnetic
  - Visual
- Characteristics
  - Local – sensors capable of measuring quantities in a local region
  - Distributed – measurements from several sensors must be incorporated for a decent estimate
  - Redundant – all sensor measurements unnecessary



Estimation Task in Sensor Network

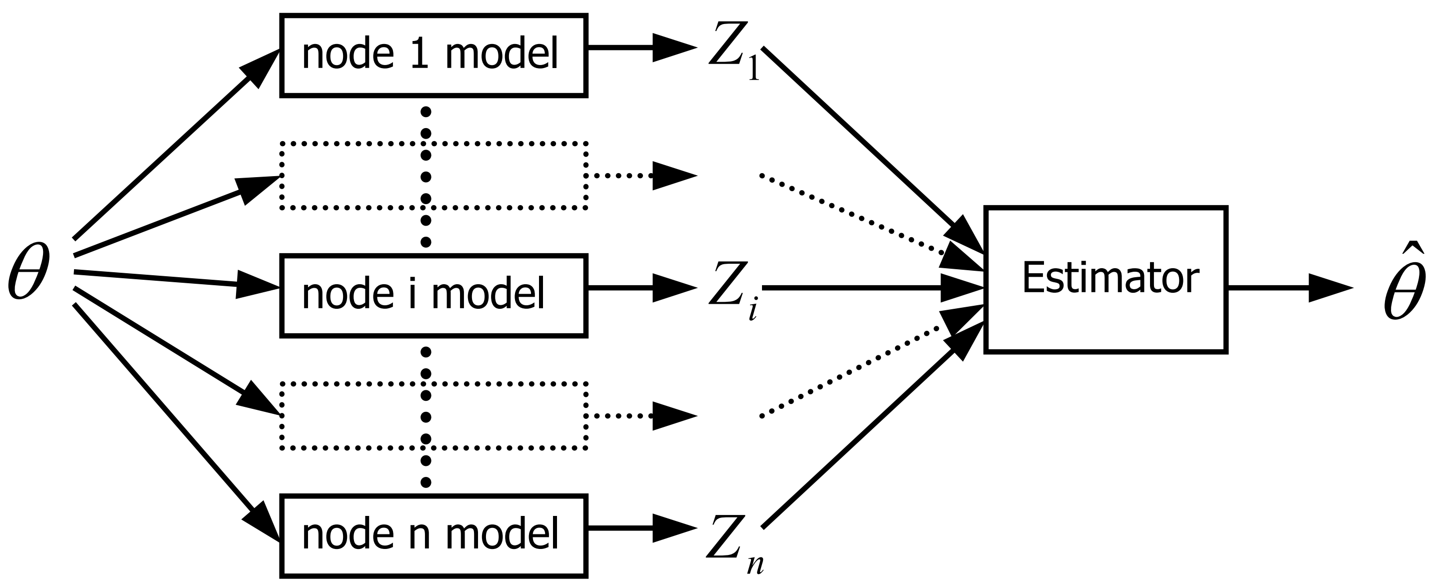
- Usual Parameter Estimation
  - Given model of parameters to measurements, estimate the parameter when given measurements.

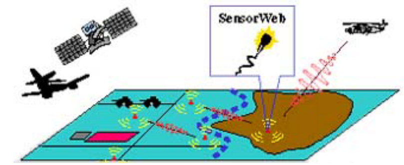




Estimation Task in Sensor Network

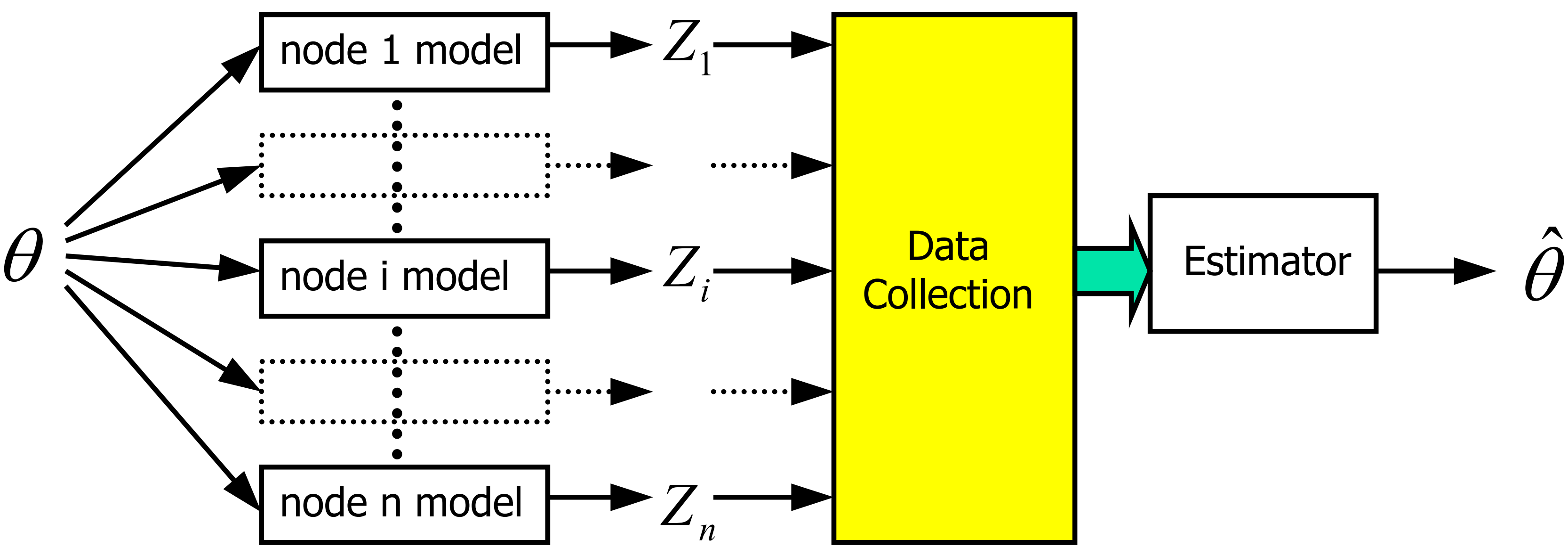
- Unique aspect in Sensor Network
  - Measurements to estimate parameters are distributed throughout different sensor nodes.
  - Cost of communicating measurements to a single node for processing is significant due to power constraints.

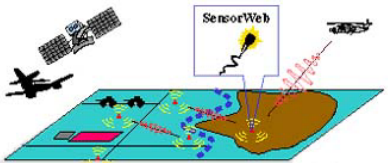




Estimation Task in Sensor Network

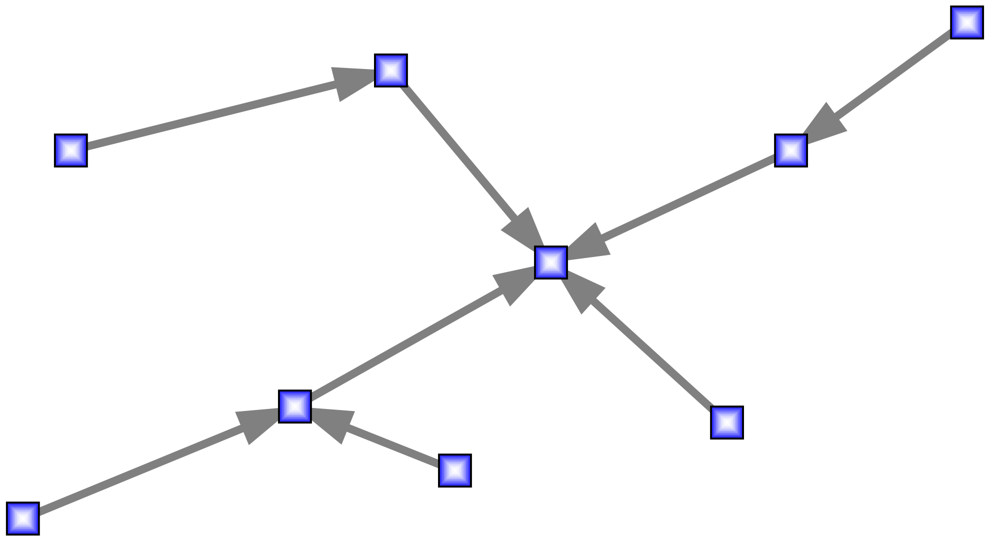
- Unique aspect in Sensor Network
  - Measurements to estimate parameters are distributed throughout different sensor nodes.
  - Cost of communicating measurements to a single node for processing is significant due to power constraints.



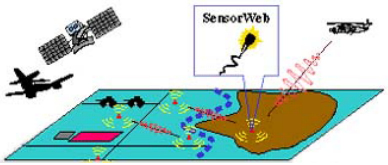


Static Data Collection

- Communication cost proportional to distance.
- Minimal spanning tree.

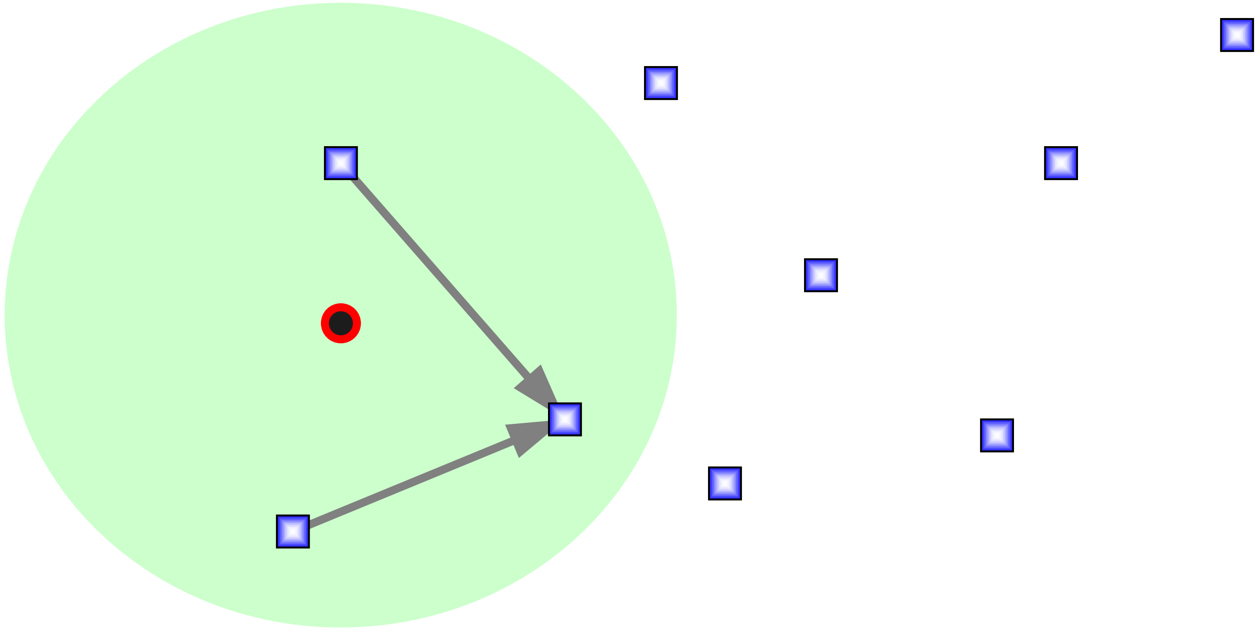


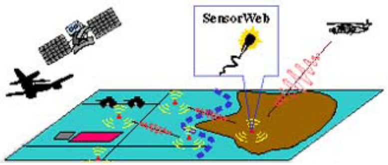
Communication cost is the same regardless of what parameter generated data.



Dynamic Data Collection

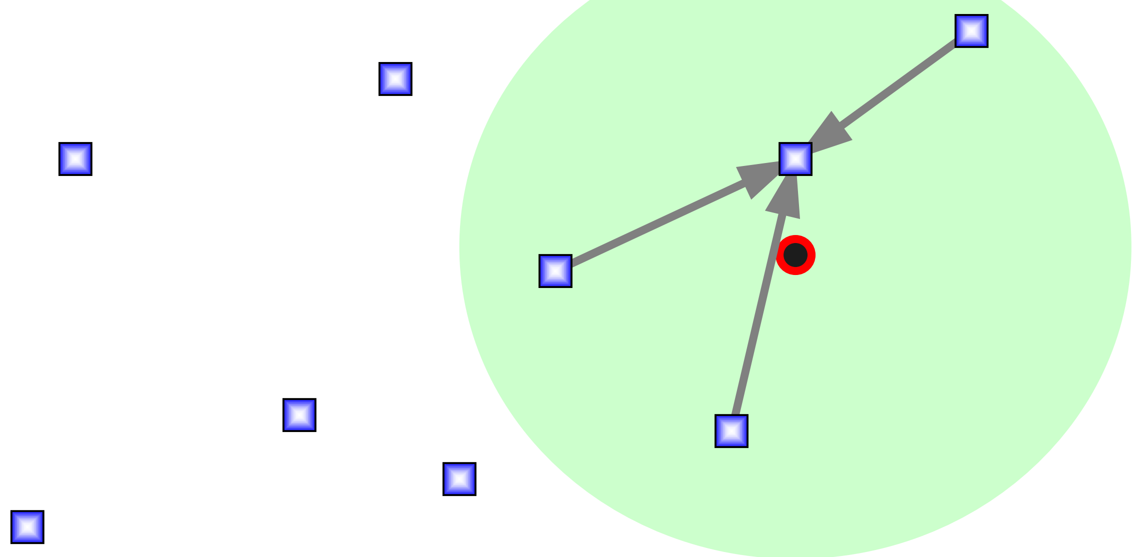
- Due to locality of sensor measurements, knowledge of the parameter generating the data implies only a subset of the sensors’ measurements need to be collected.



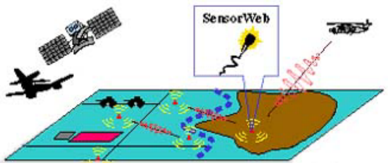


Dynamic Data Collection

- Due to locality of sensor measurements, knowledge of the parameter generating the data implies only a subset of the sensors' measurements need to be collected.

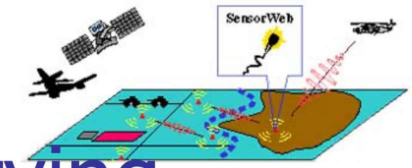


- We do not know what parameter generated data until data is collected.
- However, knowing a few of the sensor measurements tells us what subset of parameters could have generated data.



Communication Protocols

- Point-to-Point
  - "pull" type - node queries for a remote node's data
    - Auxiliary communications required
    - Based on local information of querying node
  - "push" type – node sends information to remote node
    - No auxiliary communications
    - Based on local information of sending node
- Broadcast-to-a-Region
  - More appropriate for wireless communication
  - Less refined than point-to-point (no receiver specified)



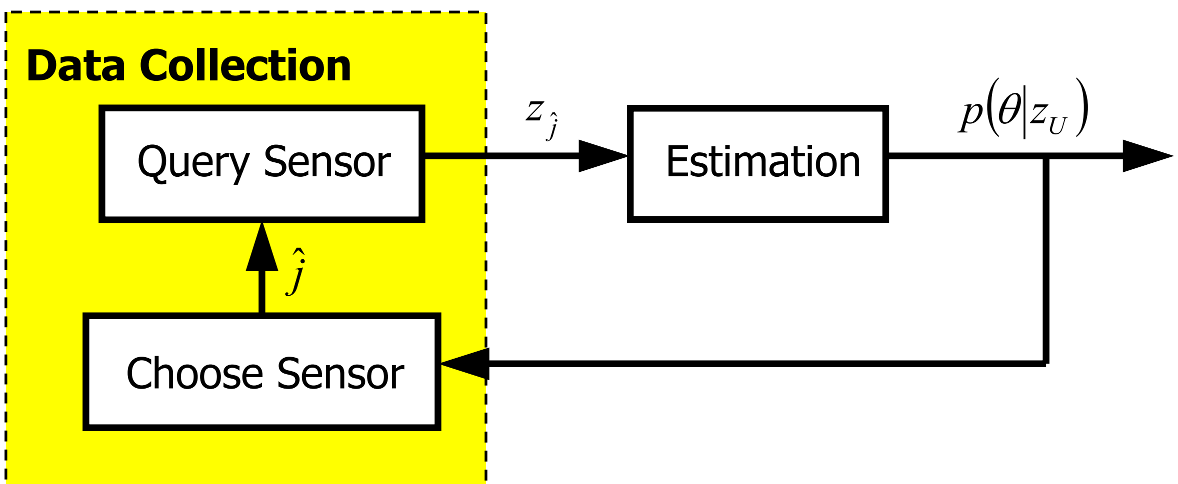
Information-Driven Sensor Querying (IDSQ)

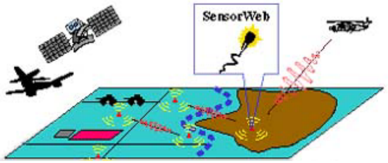
Work funded by DARPA: SensIT program (Sri Kumar, PM)
Joint work with F. Zhao, H. Haussecker

- **Idea:** Choose next measurement to incorporate into posterior distribution based on maximal information content.

$$\hat{j} = \arg\max_{j \in \{1,\ldots,n\} - U} E_{z_j}\left[\mathrm{I}\left(p\left(\theta \middle| z_U \cup \{z_j\}\right)\right)\right]$$

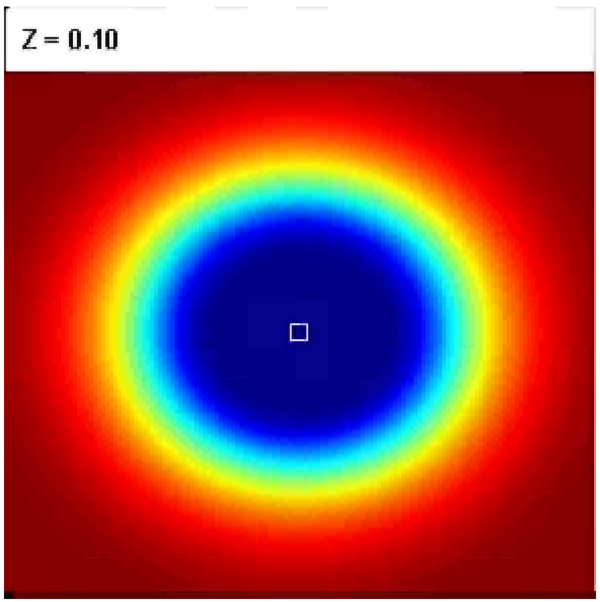
U - set of sensor node measurements already incorporated





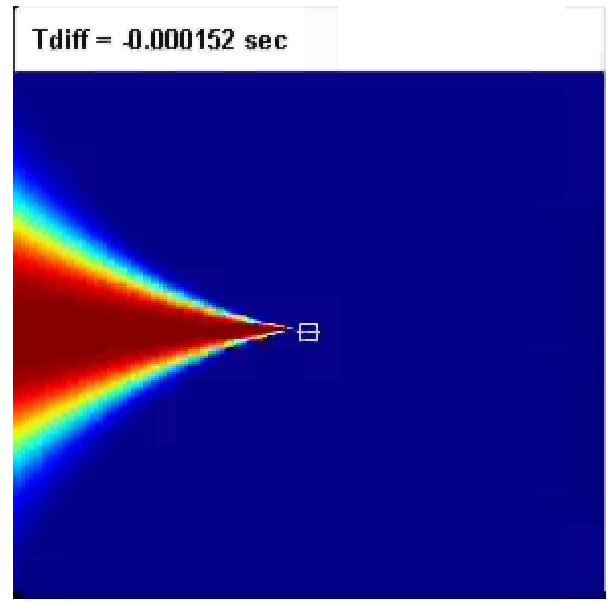
Measurement Models

Amplitude



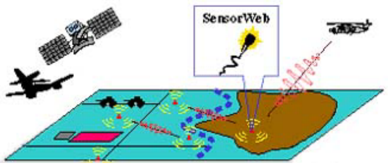
$$Z_i = \frac{A}{\|\theta - x_i\|^{\alpha}} + N_i$$

Time Difference of Arrival (TDOA)

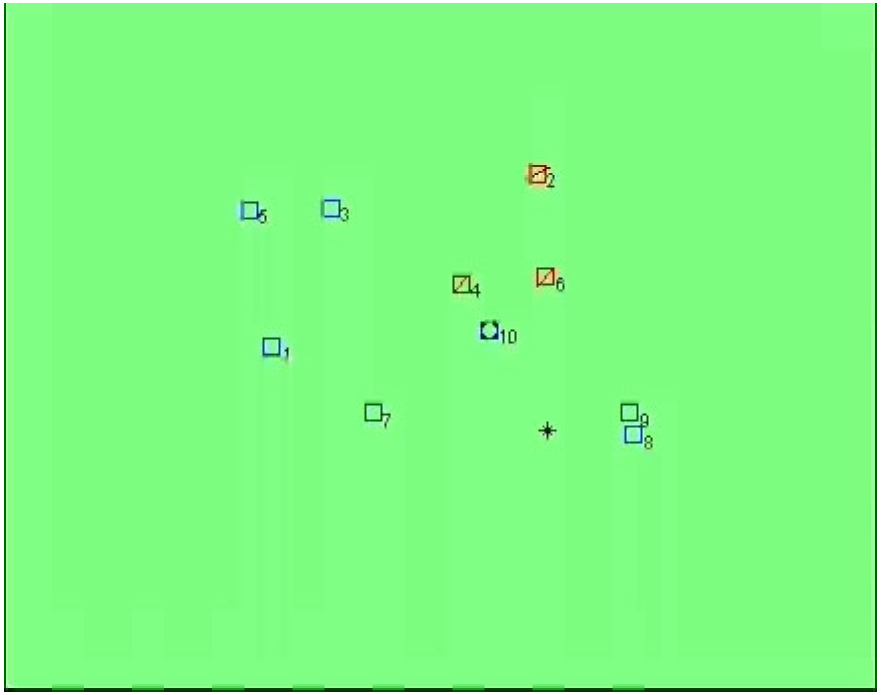


$$\Delta T_i = T_{i,1} - T_{i,2}$$

$$T_{i,j} = \frac{\|\theta - x_{i,j}\|}{c}\left(1 + N_{i,j}\right)$$



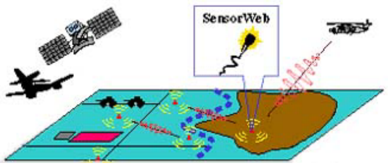
IDSQ for Stationary Target Localization



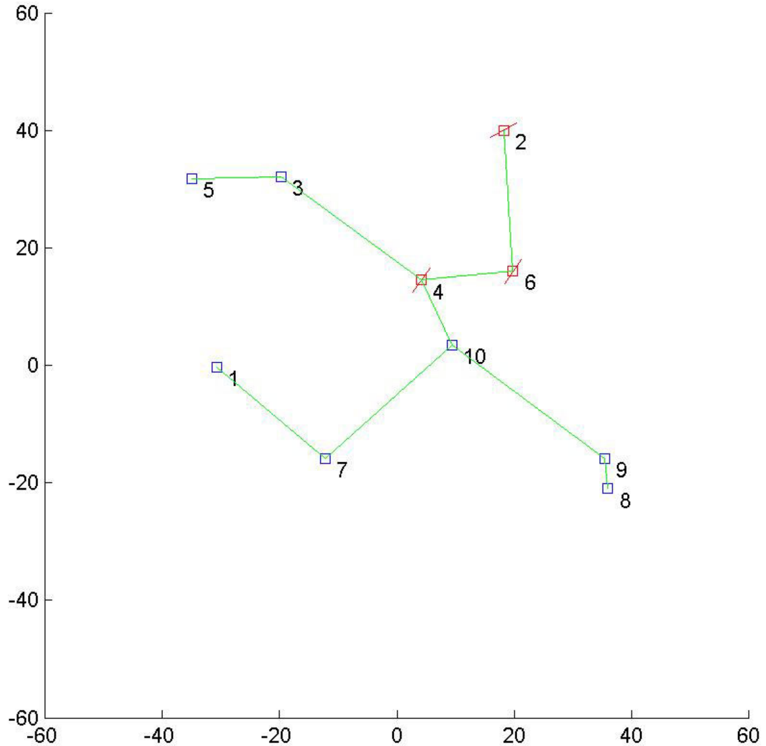
Communications
blue – query request
green – measurement value

Sensor Type
Blue square – amplitude
Red square – TDOA dipole

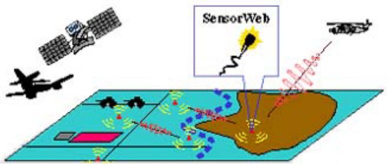
Target
asterisk – target location
green area – posterior distribution



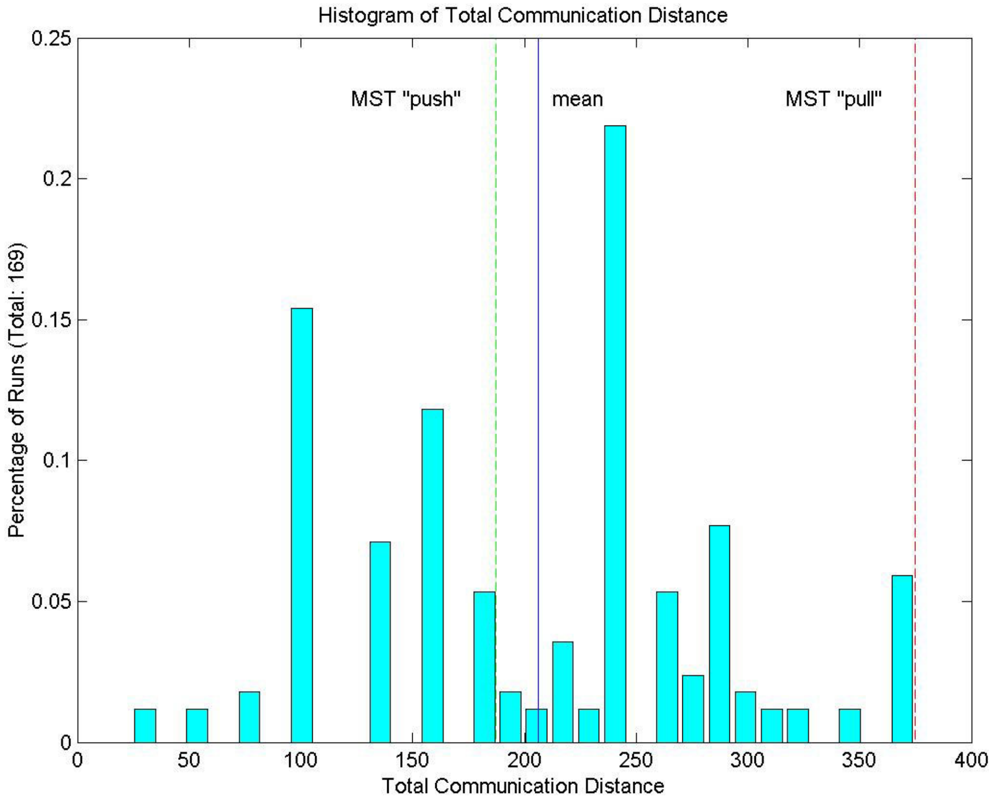
Minimal Spanning Tree

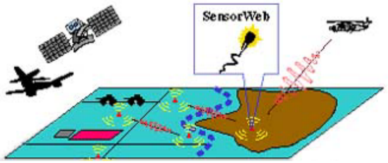


Total communication distance = 187.37 ft



Comparison
IDSQ vs. Minimal Spanning Tree



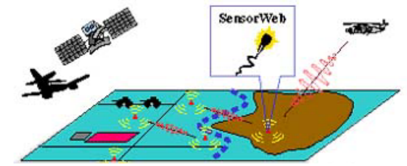


IDSQ for Target Tracking

50 sensors, randomly placed in 100x100 ft square

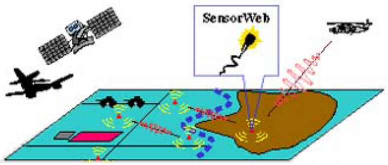
Questions

- Choosing leader node.
- Allowing concurrent communications.

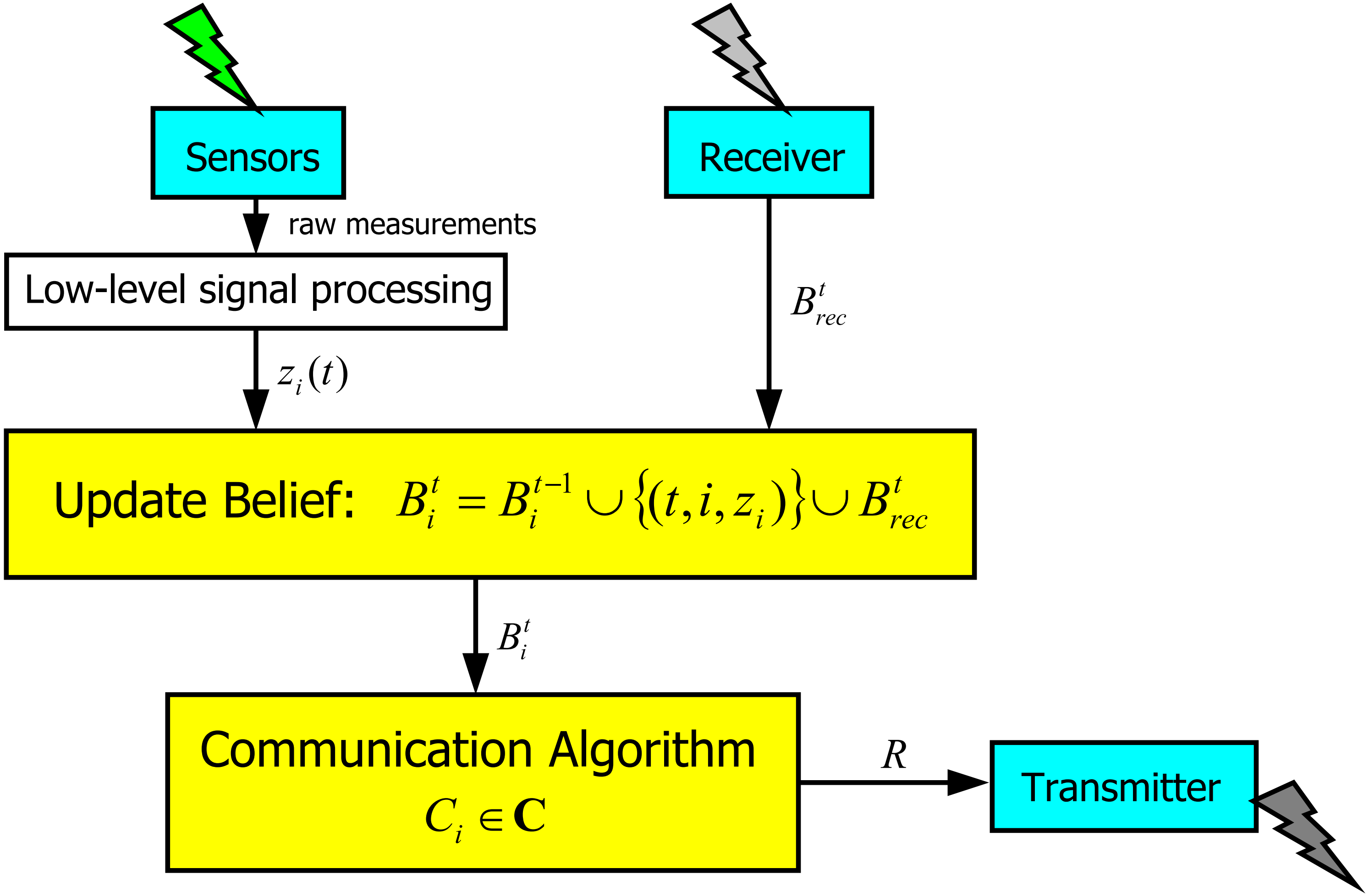


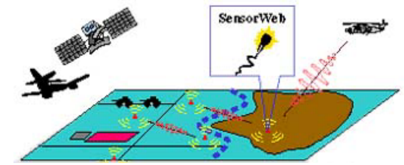
Data Collection Problem Formulation

- Sensor Node Architecture
- Definitions
  - Belief
  - Communication rule
  - Communication algorithm
  - Communication history
- Problem Statement
- Construction



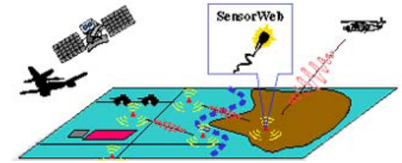
Sensor node architecture





Belief

- A belief is a collection of triples $\{(t, i, z_i)\}$
  - Time $t \in [0, \infty)$
  - Sensor ID $i \in \{1, \ldots, n\}$
  - Measurement Value $z_i \in Z_i$
- Set of Beliefs: $\mathbf{B}$
- Posterior distribution is calculated from data in a belief.
- Practical representation of belief need not be a collection of data.
  - Approximate by family of parameterized distributions.
  - Approximate by samples from distribution like particle filters.

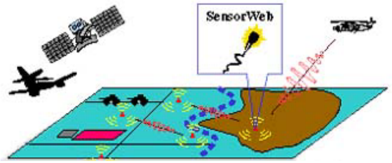


Communication Rule and Algorithm

- A communication rule is a pair (B, R)
  - Precondition: belief $B \in \mathbf{B}$
  - Action: send belief to subset of sensors $R \subset \{1, \ldots, n\}$
- A communication algorithm is a function

$$C : \mathbf{B} \rightarrow 2^{\{1, \ldots, n\}}$$

  - Collection of communication rules
  - Action: Evaluate communication rule for the current belief. If non-empty, transmit belief to appropriate sensor nodes.
- Set of communication algorithms: $\mathbf{C}$



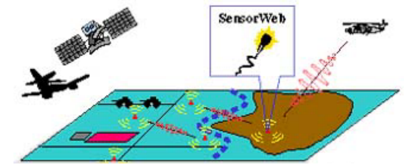
Communication History

- A *communication history* is a record of communications that have occurred from a set of communication algorithms.

$$\left(H_t\right)_{t\in[0,\infty)}$$

$$H_t \subset \{1,\ldots,n\}\times\{1,\ldots,n\}$$

- depends on
  - time series of measurements $\{z_i(t)\}_{i=1,t\in[0,T]}^{n}$
  - communication algorithms $\{C_j\}_{j=1}^{n}$



Problem Statement

- Choose a set of communication algorithms

$$\{C_j\}_{j=1}^{n}$$

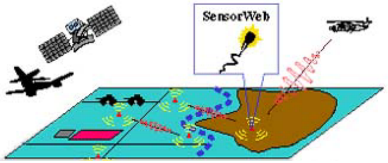
such that

  - *(information constraint)* some sensor node has in its belief enough data to compute an estimate and

$$p\left(\theta \mid [B_i^{\infty}]_t\right) \approx p\left(\theta \mid z_1(t), \ldots, z_n(t)\right)$$

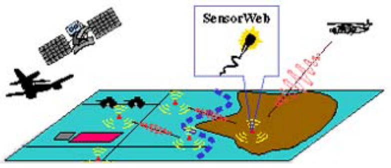
  - *(communication optimization)* the average communication cost is minimized.

$$\mathbf{E}_{Z_1^n}\left[\mathbf{cost}((H_t)_{t\in[0,\infty)})\right]$$

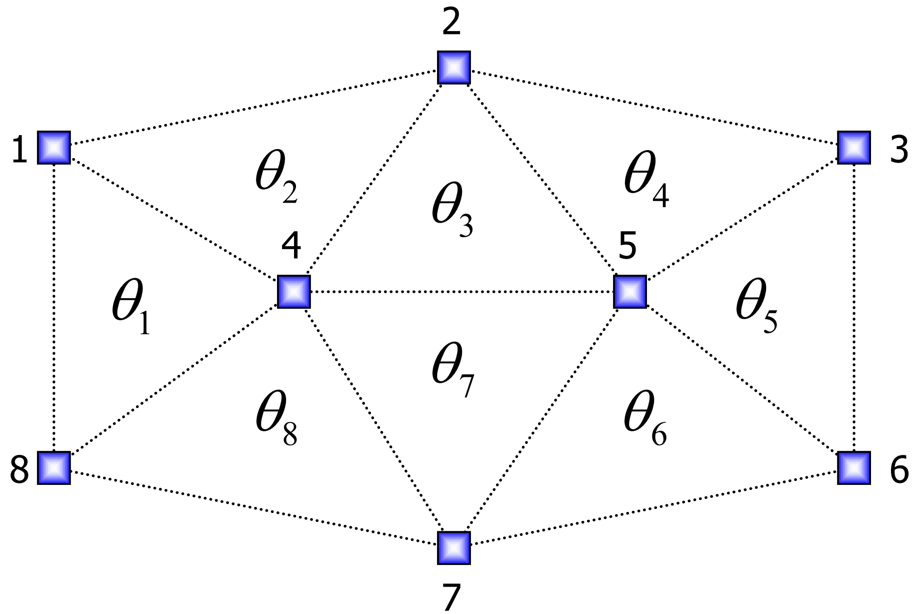


Construction of Distributed Communication Algorithms

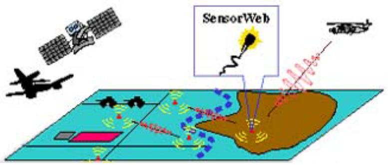
- Bipartite graph representation for capturing information constraints
- Construct a hierarchy from bipartite graph
- Convert hierarchy to communication algorithms



Simple Sensor Network Example

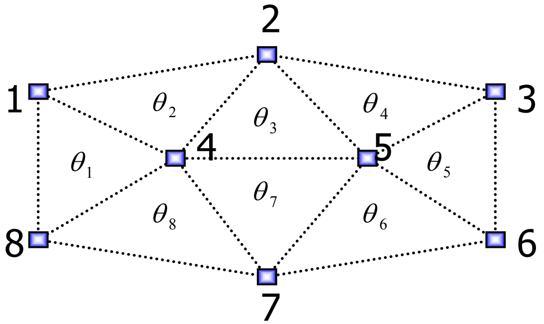


- Sensor nodes measure two values {0,1}.
- Estimation task is to determine whether high valued sensors are in triangular formations.



Bipartite Graph Representation

- Information constraints



$\theta_1 : Z_1 = 1, Z_4 = 1, Z_8 = 1$

$\theta_2 : Z_1 = 1, Z_2 = 1, Z_4 = 1$

$\theta_3 : Z_2 = 1, Z_4 = 1, Z_5 = 1$

$\theta_4 : Z_2 = 1, Z_3 = 1, Z_5 = 1$

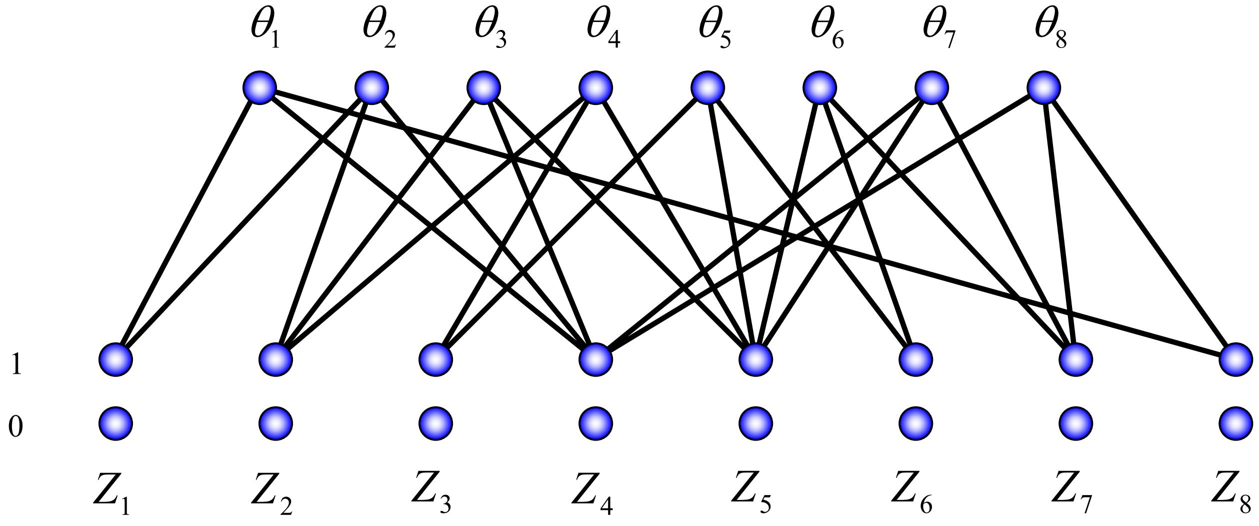
$\theta_5 : Z_3 = 1, Z_5 = 1, Z_6 = 1$

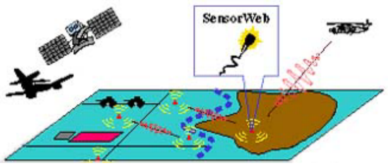
$\theta_6 : Z_5 = 1, Z_6 = 1, Z_7 = 1$

$\theta_7 : Z_4 = 1, Z_5 = 1, Z_7 = 1$

$\theta_8 : Z_4 = 1, Z_7 = 1, Z_8 = 1$

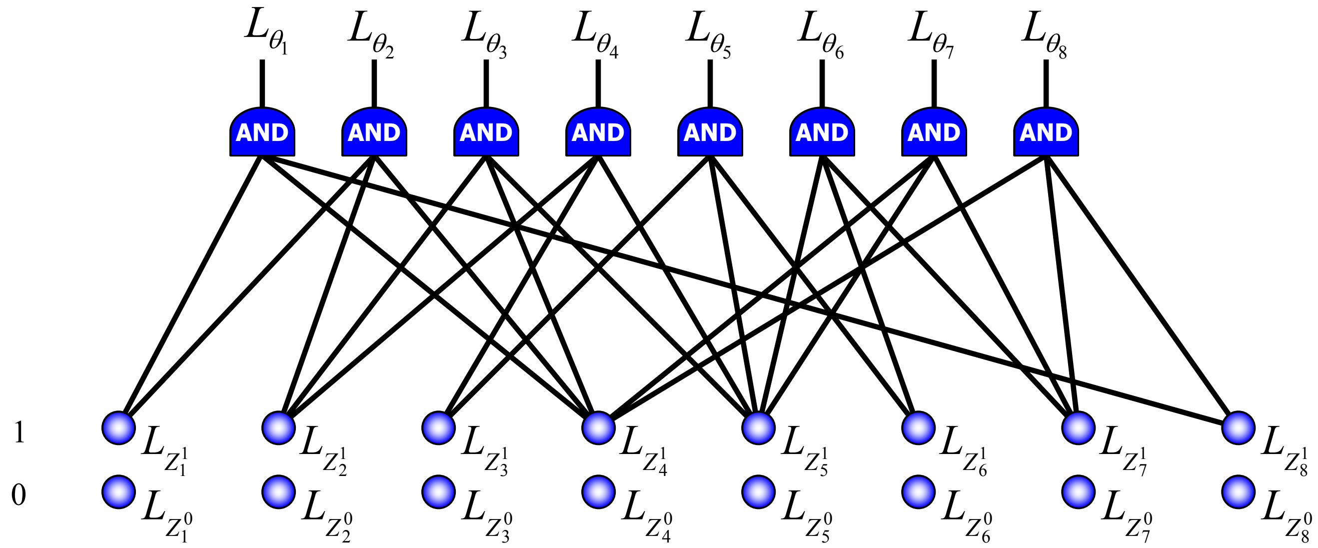
- Bipartite graph



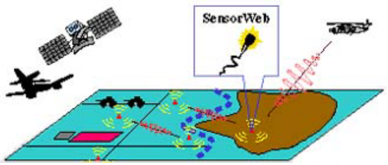


Interpretation as Feedforward Network

- Associate a boolean variable with each vertex of bipartite graph.
- Higher level representations considered to be logical functions of boolean variables from lower level representations.

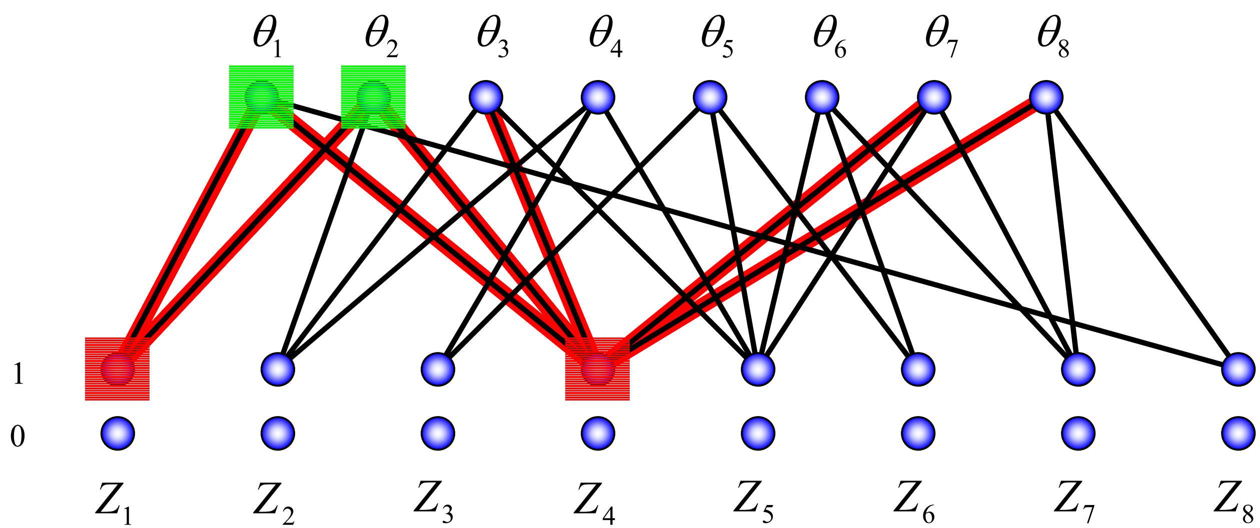


- Idea is to represent complex functions by compositions of simple primitive functions.
- Primitive functions should be associated with the primitive operations of original problem.

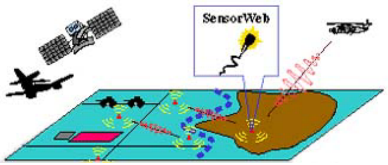


Hierarchy Construction

- Primitive operation for communication algorithms is sending a belief to another node.
- Primitive function for hierarchy is a two-input logical AND.

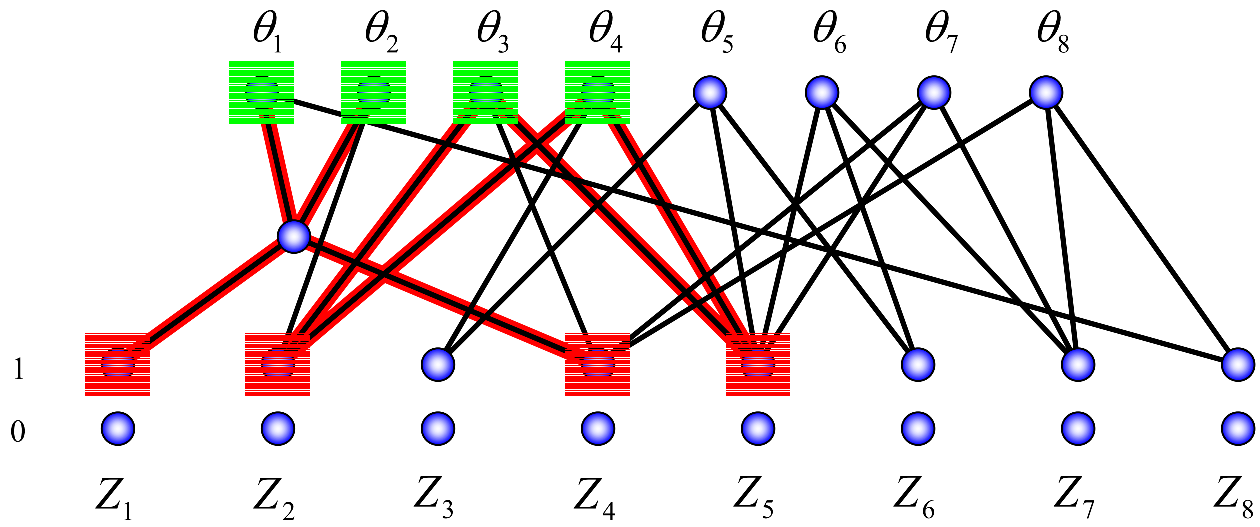


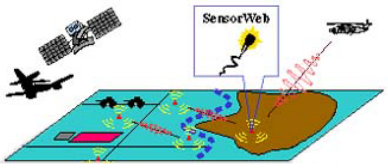
- Add intermediate nodes until all nodes are associated with primitive functions.



Hierarchy Construction

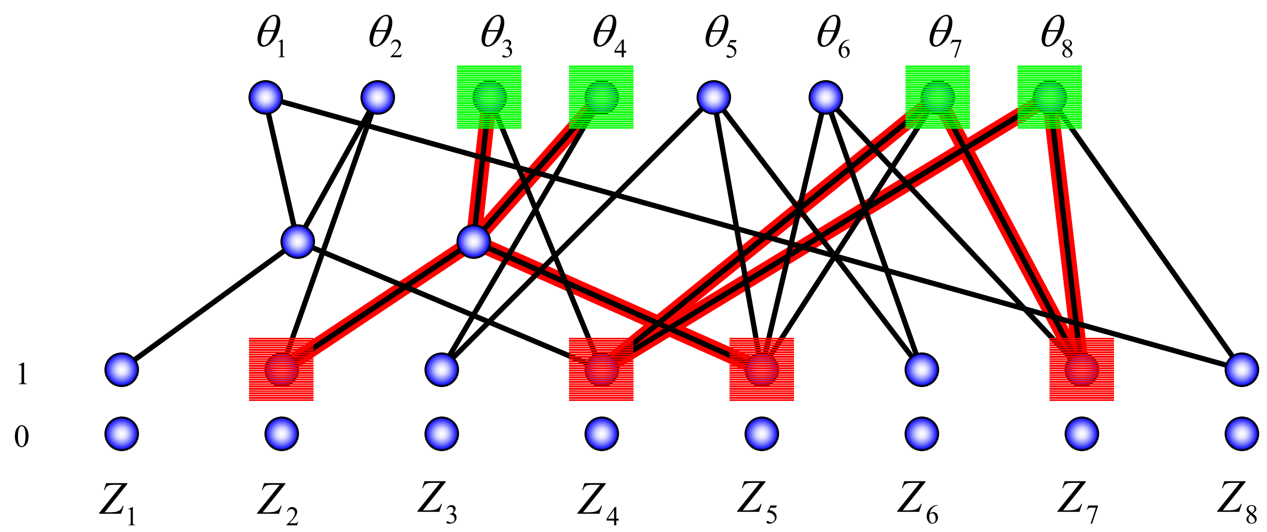
- Primitive operation for communication algorithms is sending a belief to another node.
- Primitive function for hierarchy is a two-input logical AND.

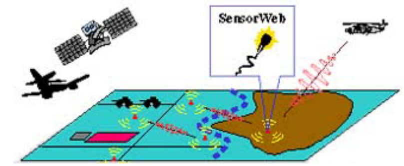




Hierarchy Construction

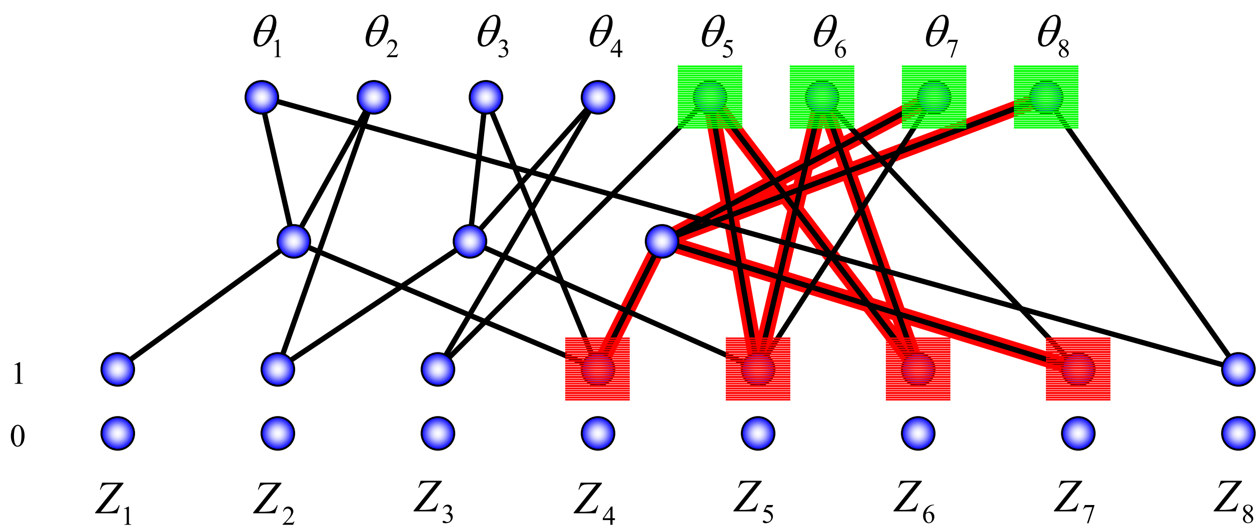
- Primitive operation for communication algorithms is sending a belief to another node.
- Primitive function for hierarchy is a two-input logical AND.

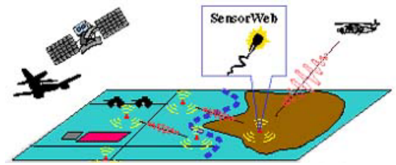




Hierarchy Construction

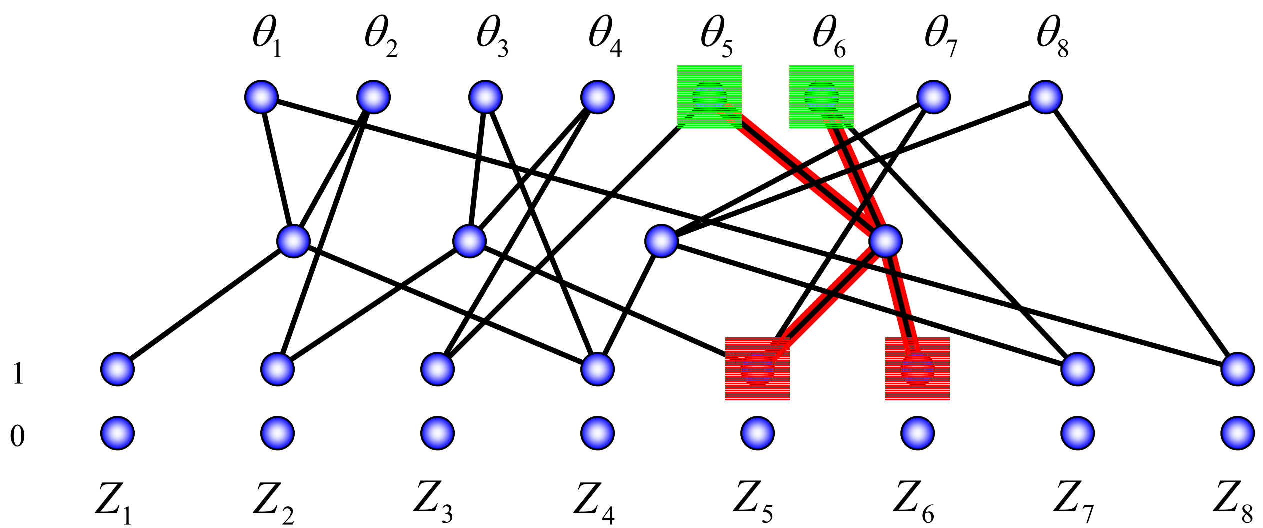
- Primitive operation for communication algorithms is sending a belief to another node.
- Primitive function for hierarchy is a two-input logical AND.



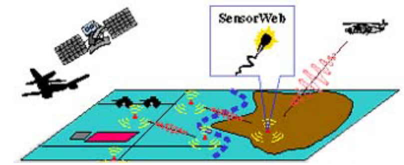


Hierarchy Construction

- Primitive operation for communication algorithms is sending a belief to another node.
- Primitive function for hierarchy is a two-input logical AND.

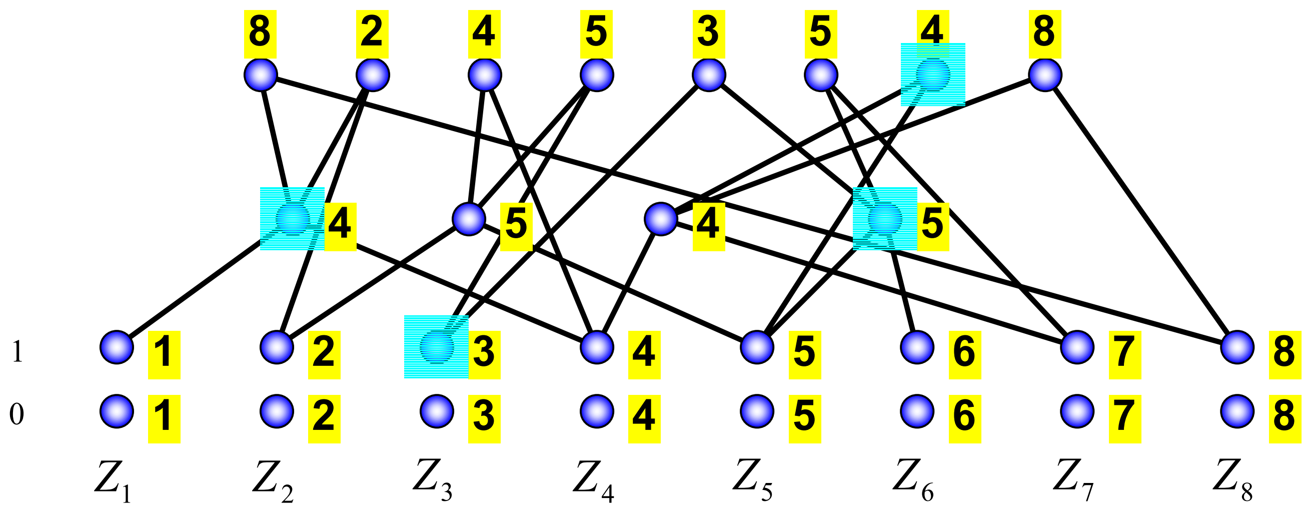


- Resulting hierarchy where every node is associated with a two-input logical AND.



Hierarchy to Communication Algorithm

- Assign hierarchy nodes to sensor nodes.
- Read off communication rules.

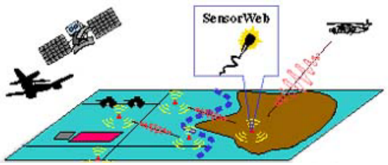


Belief : $Z_1 = 1,\ Z_4 = 1$
Send : $\{2,8\}$

Belief : $Z_3 = 1$
Send : $\{5\}$

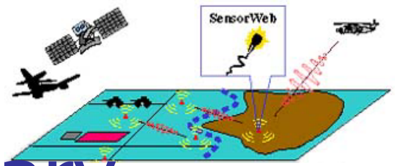
Belief :
Send :

Belief : $Z_4 = 1,\ Z_5 = 1,\ Z_7 = 1$
θ_7 **present in data**

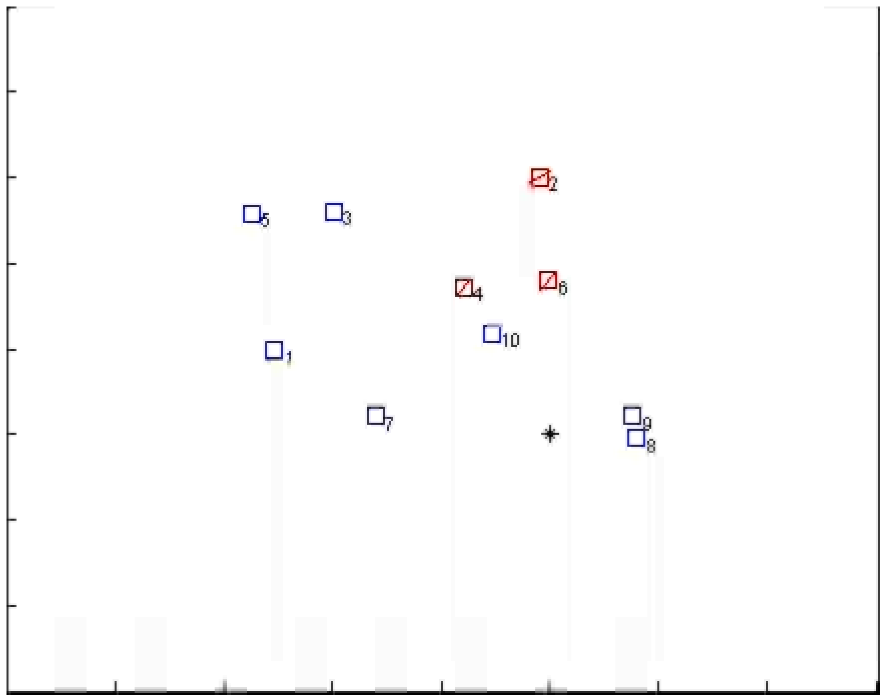


Resulting Communication Algorithms

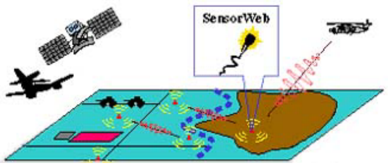
Sensor Node	Rule Number	Rule Belief	Send
1	1	Z_1 = 1	4
2	1	Z_2 = 1	5
3	1	Z_3 = 1	5
4	1	Z_1 = 1, Z_4 = 1	2, 8
	2	Z_4 = 1, Z_7 = 1	8
5	1	Z_5 = 1	4
	2	Z_2 = 1, Z_5 = 1	4
	3	Z_5 = 1, Z_6 = 1	3
6	1	Z_6 = 1	5
7	1	Z_7 = 1	4, 5
8	0	*no rules*	



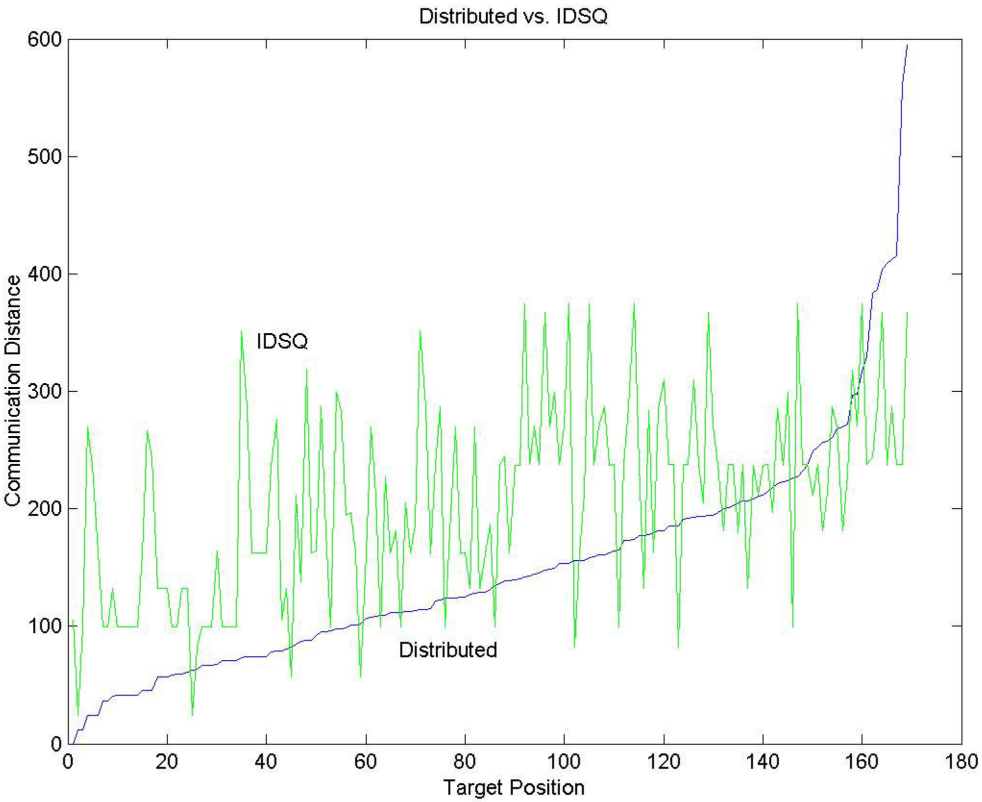
Distributed Algorithm for Stationary Target Localization



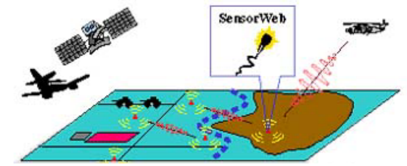
- Concurrent communications can occur for faster data collection.
- Only a single node carries the global belief after all communication have settled.



Comparison
Distributed Algorithm vs. IDSQ

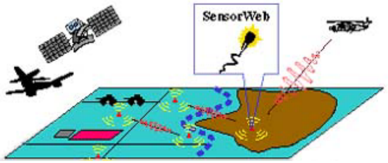


Percent better: 80.5%
Distributed Mean: 149.4
IDSQ Mean: 206.2



Discussion and Future Work

- Two modes of data collection
  - Distributed algorithm
    - Estimate quality poor
    - Generous data collection
  - IDSQ
    - Estimate quality good
    - Parsimonious data collection
- Tracking
  - Use distributed algorithm to initialize tracker and determine leader.
  - Use modified IDSQ to perform tracking.

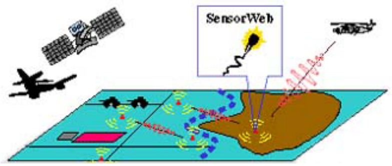


Conceptual Problem

- How are open nonlinear dynamical systems composed of an interconnected assembly of subsystems?
- How are automata composed of an interconnected assembly of (sub-)automata?

What we have done

Our problem is vastly more difficult than Krohn-Rhodes



Gluing Systems

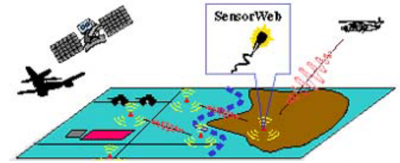
$$\textbf{Gluing} = \left\{\begin{matrix}\textbf{creating a new system by making} \\ \textbf{identification on a totality of systems}\end{matrix}\right\}$$

$$\left\{\begin{matrix}\textbf{gluing behaviors} \\ \textbf{of subsystems}\end{matrix}\right\} \textbf{ CONSTRAINS } \left\{\begin{matrix}\textbf{the behavior of} \\ \textbf{the overall system}\end{matrix}\right\}$$

Local Behavior

Local Behavior = "switching off" every other subsystem except the one under focus

$$\left\{\begin{matrix}\textbf{LOCAL BEHAVIORS} \\ + \\ \textbf{INTERCONNECTION DATA}\end{matrix}\right\} \rightarrow \textbf{OUR COHOMOLOGY GROUPS}$$

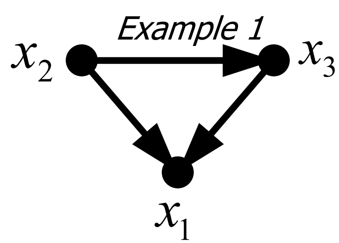


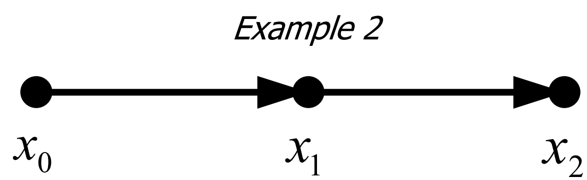
Examples

Our cohomology theory: nontrivial, discriminates between systems

- Example 1 (underlying graph is exactly a cycle)

$$\dot{x}_1(t) = x_2(t)\cdot u_1(t), \dot{x}_2(t) = x_3(t)\cdot u_2(t), \dot{x}_3(t) = x_1(t)\cdot u_3(t), y = x_1$$





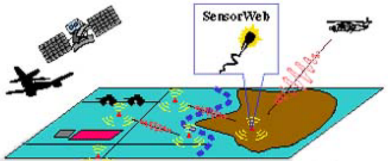
H^1= invariant differential operators on a real line

H^1= 0

Recall: representation theory of invariant differential operators —> origin of harmonic analysis. The latter underlies most of LTI systems in signal processing & control

- Example 2 (Underlying graph a linear tree - gluing of ends absent)

$$\dot{x}_0 = 0, \dot{x}_1 = x_0\cdot u_1(t), \dot{x}_2 = x_1\cdot u_2(t), y = x_2$$



Codes from Languages

- From automata, we produce a sequence of error-correcting codes. Suitable projections of the inclusions

  {degree-r cocycles in the cohomology} $\subset$ {degree-r cochains}

Why Build Codes in This Way?

- Cohomological nature of codes —> a conceptual platform to analyze the distance of codes
- Yields almost a dictionary: languages —> codes. Languages - studied extensively