

---

# Efficient Gathering of Correlated Data in Sensor Networks

---

Presented by,

Wiwek Deshmukh

---

- Definition:

A sensor  $s$  is said to be correlated to a set of sensors  $S$  if the data measured by  $s$  can be inferred/computed from the data measured by the sensors of  $S$  within an acceptable error bound as defined by the application.

---

- 
- All sensor nodes transmit their measured data of interest to the data gathering node upon being queried.
  - The focus of this paper is to exploit inherent data correlations and reduce the number of sensors that need to transmit data.
  - If a sensor 's' is correlated to a set S, and each sensor in S is transmitting, then 's' need not transmit.
-

- 
- What are we interested in ?
  
  - Given a sensor network, select a minimum set of sensors  $M$ , called connected correlation dominating set, such that:
    - (a) each sensor that is not in  $M$  is correlated to a subset of sensors in  $M$
  
    - (b) the selected set of sensors  $M$  forms a connected communication graph.
- (OR)
- the communication sub-graph induced by  $M$  is connected.
-

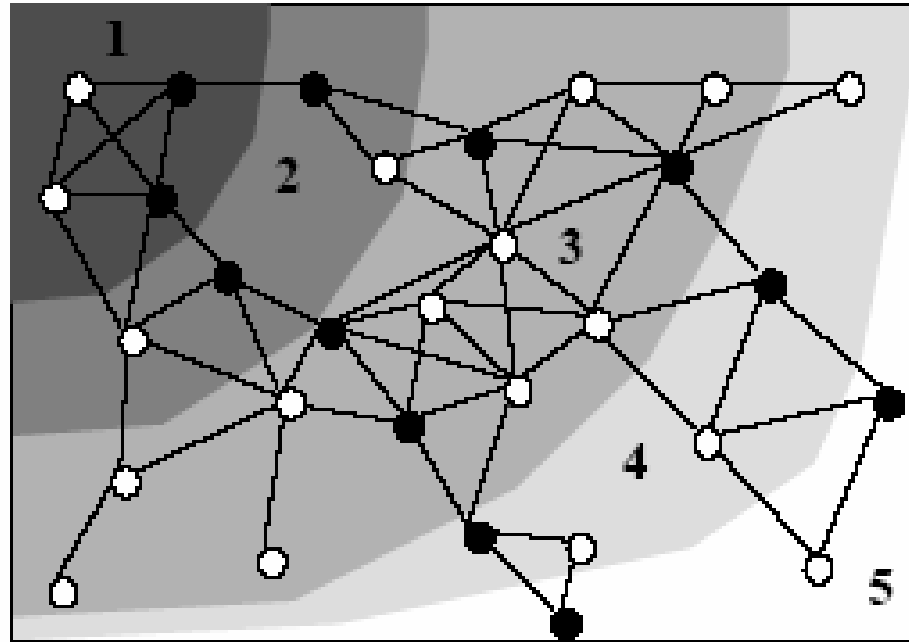
- 
- If sensor data is rich in correlations, then  $|M| \ll n$ , where  $n$  is the set of all sensors.
  - The data is relayed to the data-gathering node over a communication tree spanning over  $M$ .
-

---

# Example:

- Let us assume:  
The data correlations between sensor nodes can be represented by a simple rule:  
  
For a given region, any 2 sensor data values are sufficient to infer the data values of all other sensors in the region.
  - The correlation structure is then represented by a correlation **hypergraph** where every sensor node  $s$  has a hyperedge  $((s_i, s_j), s)$  incident on it for every pair of sensor nodes  $s_i$  and  $s_j$  that belong to the same region as  $s$ .
-

■ Example:



The set of black nodes form a connected correlation dominating set. Here,  $|M| = 12$  and  $n = 30$ .

---

# Formal Definitions:

- **Definition 1:** Communication Graph.
- **Definition 2:** Correlation Graph.

Given a sensor network consisting of a set of sensors  $I$ , the correlation graph over the sensor nodes is a directed hypergraph with  $I$  as the set of vertices, and a subset of  $(P(I) \times I)$  as the set of directed hyperedges, where  $P(I) =$  Power set of  $I$ .

Therefore a correlation graph is a hypergraph:

$$G(V = I, E \subseteq (P(I) \times I)).$$

---

---

# Formal Definitions (Contd...):

- **Definition 3: Correlation Neighbor**
  - A hyperedge is also referred to as a correlation edge. In a correlation edge  $(S,s)$ , for any  $x \in S$ ,  $s$  and  $x$  are called correlation neighbors.
  - We assume that in a hyperedge  $(S,s)$ ,  $s \notin S$ .
-

# Distributed Computation of Correlation Graph

- The authors use the least squares method to compute the correlation graph of the sensor network.

$$S = \{s_1, s_2, \dots, s_L\}.$$

$$s'[k] = \sum_{l=1}^L \alpha_l s_l[k],$$

$$E(\alpha) = \sum_{k=1}^K (s[k] - s'[k])^2,$$

- The weighted coefficients are chosen to minimise the least square error as shown above, where K is the number of samples.

---

# Distributed Computation of Correlation Graph (Contd ...)

- Let  $N(s,d)$  denote the set of  $d$ -hop neighbors of  $s$ .
  - To compute all correlation edges a node  $s$  is involved in, each node  $s$  in the sensor network should collect sufficient samples from nodes in  $N(s,d)$ , where  $d$  is sufficiently large to capture all data correlations.
  - We can collect  $d$ -hop neighbors' data at each node by piggybacking over data gathering messages for  $d$  snapshots.
-

---

# Energy Efficient Distributed Algorithm

- Initially each node assigns itself a priority.  
(eg: its sensor ID)  
Priorities are used to avoid cyclic dependency of conditions.
  - Each node collects k-hop neighborhood information.  
(we may choose a small value of k, eg: k=3)
  - Neighborhood information may be gathered during the data-gathering process using piggyback strategy.
  - Each node periodically tests for a set of conditions to be satisfied.
-

---

# Energy Efficient Distributed Algorithm

## (Contd... )

- If all the conditions are satisfied, the node marks itself as deleted and instructs some of its correlation neighbors to mark themselves selected.
  - Once a node is selected., it cannot be marked deleted in the future.
  - The selected marking of a node signifies that it is being used to infer another node and hence should not be marked deleted in the future.
-

---

# Energy Efficient Distributed Algorithm

## (Contd... )

- At any stage, the set of nodes that have not been marked form a connected correlation-dominating set.
  - The data gathering node is always selected by default. (since it is part of the connected correlation-dominating set).
-

---

# Energy Efficient Distributed Algorithm (Contd... )

## Conditions for marking as deleted

A node can be deleted if its value can be inferred  
& connectivity is preserved.

**C1:** The node  $s$  has not been marked selected already. (Trivial)

**C2:** In the communication subgraph induced over the set of non-deleted nodes using only the  $k$ -hop neighborhood information, every pair of neighbors  $(u,v)$  of  $s$  are connected by a path with all intermediate nodes having priority  $< p(s)$ .

C2 ensures that deletion of  $s$  preserves the connectivity of the sub-graph induced by non-deleted nodes.

---

# Energy Efficient Distributed Algorithm (Contd... )

## Conditions for marking as deleted

**C3:** There is a correlation edge  $(S,s)$  in the correlation graph such that every node in the set  $S$  is either marked selected or has a priority  $< p(s)$ .

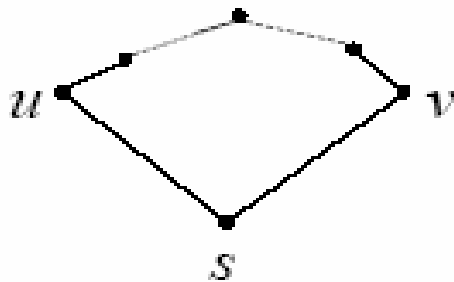
This condition selects a set of nodes  $(S)$  that can be used to infer 's' through a correlation edge.

**C4:** For every correlation edge  $(R,r)$  where  $s \in R$ , either  $r$  is marked selected, or is marked deleted or has a priority  $< p(s)$ .

C4 ensures that the set of nodes in  $R$  are not being chosen as selected by node  $r$  at the same time.

# Energy Efficient Distributed Algorithm (Contd...)

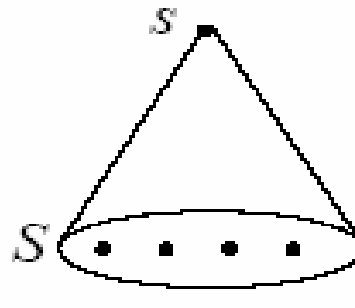
lower priority  
intermediate nodes



For all  $(u,v)$

C2

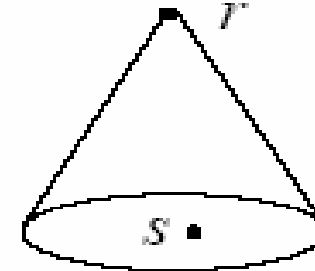
lower priority or  
“deleted/selected”



lower priority or  
“selected” nodes

For some  $S$

C3



For all  $r$

C4

---

## 2-Rounds Algorithm

(leads to increased no. of deleted nodes)

### Initial Round:

- First test C2 on every sensor  $s$ .
- Test C33 & C44 for only those 's' for which C2 is satisfied.

### Subsequent Rounds:

- Use the basic distributed algorithm.
-

---

## 2-Rounds Algorithm (Contd...)

**C33:** There is a correlation edge  $(S,s)$  in the correlation graph such that no node in the set  $S$  is marked deleted and each node in  $S$  is selected or has a priority  $< p(s)$  or does not satisfy C2.

**C44:** For every correlation edge  $(R,r)$  where  $s \in R$ , either  $r$  is marked selected, or is marked deleted or has a priority  $< p(s)$  or does not satisfy C2.

The authors find that their 2 rounds algorithm performs better.

---

---

# References

- Efficient Gathering of Correlated Data in Sensor Networks, Himanshu Gupta, Vishnu Navda, Samir R. Das, Vishal Chowdhary, Proceedings of the 6<sup>th</sup> ACM International Symposium on Mobile Ad-Hoc Networking (MobiHoc) 2005, Urbana-Champaign, IL, USA, May 25-27, 2005.
-