WIRELESS SENSOR NETWORKS
(18MCA55E)
UNIT – IV
"SENSOR NETWORK DATABASES"

FACULTY:

Dr. R. A. Roseline, M.Sc., M.Phil., Ph.D.,
Associate Professor and Head,
Post Graduate and Research Department of Computer Applications,
Government Arts College (Autonomous), Coimbatore – 641 018.

CONTENTS

WHAT IS SENSOR?

- Sensor is a device , module , machine or subsystem whose purpose is to detect events or changes in it's environments and send the information to other electronics.
- Sensor always used in other electronics.

SENSOR DATABASE

- A sensor database involves a combination of stored data and sensor data.
- Sensor data is generated by signal processing functions.
- Stored data include the set of sensors that participate in.

```
dvdrental=# select title, release_year, length, replacement_cost from film
dvdrental-#   where length > 120 and replacement_cost > 29.50
dvdrental-#   order by title desc;
          title          | release_year | length | replacement_cost
-------------------------+--------------+--------+------------------
 West Lion               |         2006 |    159 |            29.99
 Virgin Daisy            |         2006 |    179 |            29.99
 Uncut Suicides          |         2006 |    172 |            29.99
 Tracy Cider             |         2006 |    142 |            29.99
 Song Hedwig             |         2006 |    165 |            29.99
 Slacker Liaisons        |         2006 |    179 |            29.99
 Sassy Packer            |         2006 |    154 |            29.99
 River Outlaw            |         2006 |    149 |            29.99
 Right Cranes            |         2006 |    153 |            29.99
 Quest Mussolini         |         2006 |    177 |            29.99
 Poseidon Forever        |         2006 |    159 |            29.99
 Loathing Legally        |         2006 |    140 |            29.99
 Lawless Vision          |         2006 |    181 |            29.99
 Jingle Sagebrush        |         2006 |    124 |            29.99
 Jericho Mulan           |         2006 |    171 |            29.99
 Japanese Run            |         2006 |    135 |            29.99
 Gilmore Boiled          |         2006 |    163 |            29.99
 Floats Garden           |         2006 |    145 |            29.99
 Fantasia Park           |         2006 |    131 |            29.99
 Extraordinary Conquerer |         2006 |    122 |            29.99
 Everyone Craft          |         2006 |    163 |            29.99
 Dirty Ace               |         2006 |    147 |            29.99
 Clyde Theory            |         2006 |    139 |            29.99
 Clockwork Paradise      |         2006 |    143 |            29.99
 Ballroom Mockingbird    |         2006 |    173 |            29.99
(25 rows)
```

SENSOR DATABASE CHALLENGES.

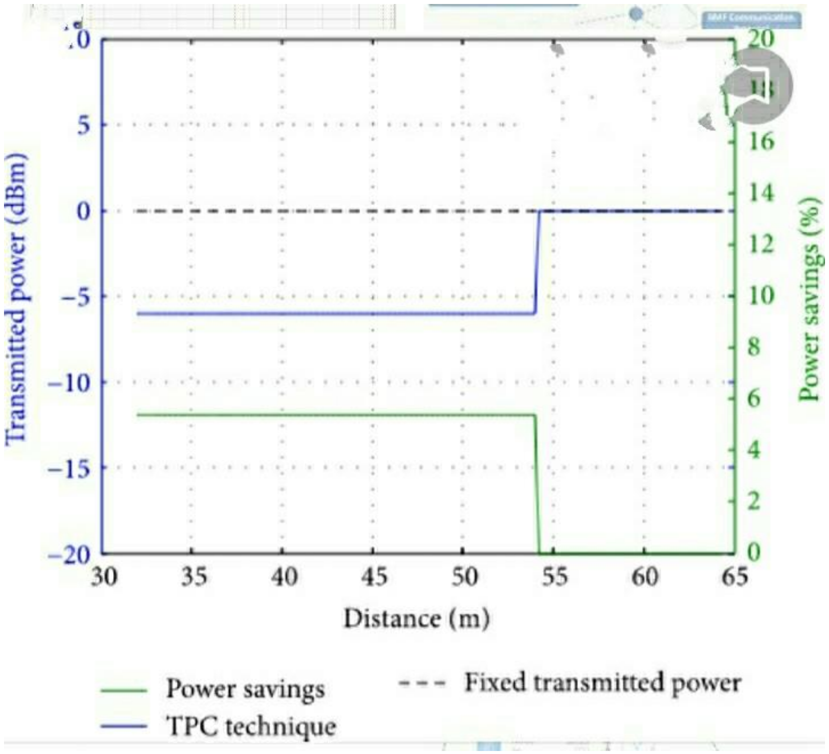
- Physical level 2 major distinguising characteristics of sensor network when it’s comes to data implementation.
  - Network replace storage and buffer manager data transfer from data held in old memory as opposed to data blocks on hold.
  - Second node memory is limited by cost and energy considerations.
  - This difference generate several new challenges for sensor network database.

SENSOR DATABASE CHALLENGES

- Energy Efficiency
- Scalability
- Delay
- Robustness
- Data transmission and transmission models
- Sensor location

1.ENERGY EFFICIENCY

- Wireless sensor networks are mostly battery shortage is major issue in this sensor networks powered. Energy specially in aggressive environments such as battle field etc.



SCALABILITY

- As sensor are becoming cheaper day by day . hundreds or thousands can be installed in wireless sensor network easily.so the routing protocol must support scalability of network.
- If further nodes are to be added in the network any time then routing protocol should not interrupt this.

DELAY

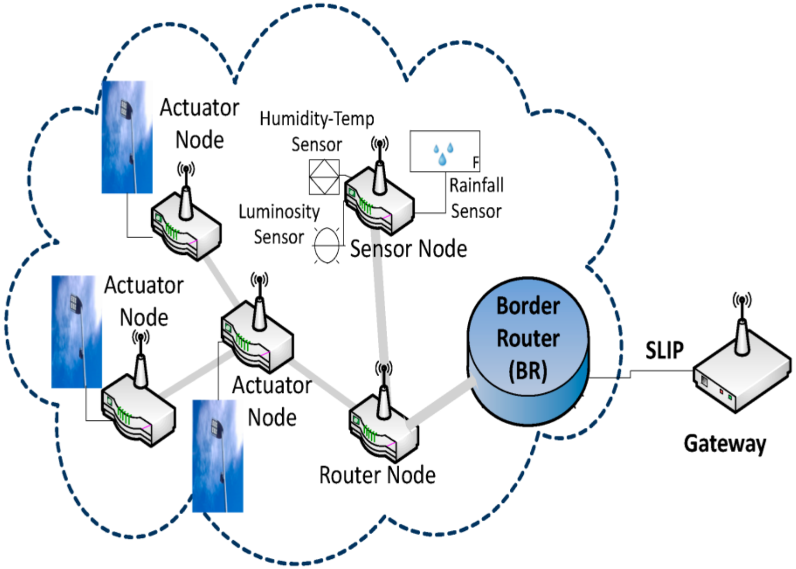
- Some application requires instant reaction or response without any substantial delay such as temperature sensor and alarm monitoring.
- So routing protocol should offer minimum delay.
- The time needed to transmitter sensed data is required to be as little as possible.

ROBUSTNESS

- Wireless sensor networks are deployed in very crucial and loss environment frequently.
- Occasionally a sensor node might be expire or leaving the wireless sensor networks.
- The routing protocol should be capable to accept allsorts of environments. the functionality of routing protocol should be fine also.

DATA TRANSMISSION

- The performance of routing protocol is a function of a network size and transmission media.
- so transmission of good quality enhances the network performance directly.



TRANSMISSION MODELS

- There are four model of transmission depending of the application in wireless sensor networks namely as
  - Query driven
  - Event driven
  - Continuous type
  - Hybrid type

SENSOR LOCATION

GPS:

- Another major challenges that is faced by WSN designer is to correctly locate of the sensor node.
- Most routing protocol use some location technique to obtain knowledge containing their locations.
- GPS receivers are used some scenario.

QUERYING THE PHYSICAL ENVIRONMENT

- What is query?
  - Query processing in wireless sensor networks provides declarative access to data from multiple sensor nodes.
  - In distributed query processing a runtime database is maintained about a specific type of query, such as how the nodes in a wsn will respond to a particular query.

QUERYING THE PHYSICAL ENVIRONMENT

- It is advantages to express queries to a sensor network database at a logical level, declarative level, using relational languages such as SQL.
- High level interfaces allow non expert to easily interact with the database.
- It would be difficult for non expert user to anticipate all the possible events and design the corresponding execution plan.

USING SQL QUERY

- Flood warming system as an example of SQL- querying of sensor networks. A user from a state emergency management agency may send a query to the flood sensor database.
- For the next three hours , retrieve every 10 minutes the maximum rainfall level in each country in southern California , if it is grater than 3.0 inches.
- This is an example of long running query.

QUERYING THE PHYSICAL ENVIRONMENT

- Example:
  - This is an example of long running, monitoring query, it can be expressed in the following SQL.

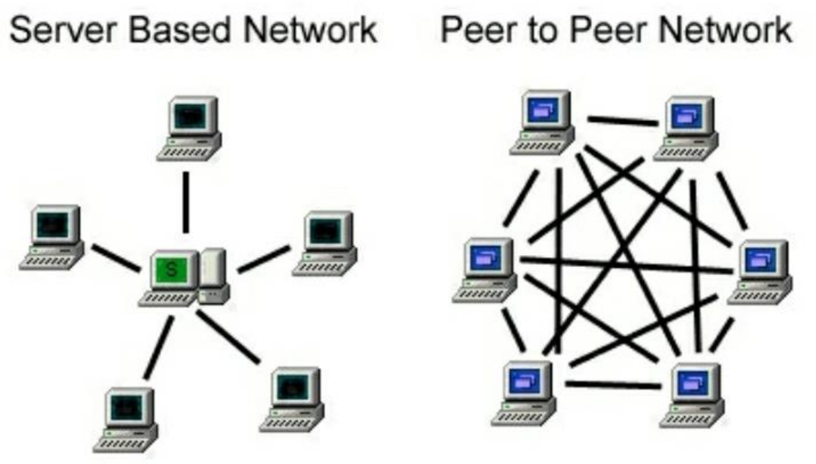
```
SELECT max(rainfall_ level),country

FROM sensors

WHERE state = Tamilnadu

GROUP BY country

HAVING max(rainfall_ level)>3.0in

DURATION [now,now+180 min]

SAMPLING PERIOD 10 min
```

QUERIES

- For peer to peer system a query can originate from any node ,the database schema will broadcast to every node.
- The query result is computed by interrelating data from a set of sensor.

P2P

- Peer to peer system(p2p).
- Files can be shared directly between systems on the network without need of central server.



A QUERY CAN ASK FOR RELATIONS OR CORRELATIONS AMONG SET OF EVENTS

- A query can ask for relations or correlations among set of events
- Example:
  - Sound an alarm whenever two sensors within 10 meters of each other simultaneously detect an abnormal temperature.

1. AGGREGATIVE QUERIES

- Query result is computed by integrating data from set of sensors.
- Delivery of data from distributed set of modes to a central node for communication.
  - AVG
  - COUNT
  - MIN
  - MAX
  - SUM

SELECT

COUNT(*)

FROM

products;

RESULT:

Count(*)

99

EXAMPLE USING QUERY FOR SUM

QUEREY

```
SELECT product_id,
SUM(quantity) stock_count
FROM production.stocks
GROUP BY product_id
ORDER BY stock_count DESC;
```

production.stocks
* store_id
* product_id
quantity

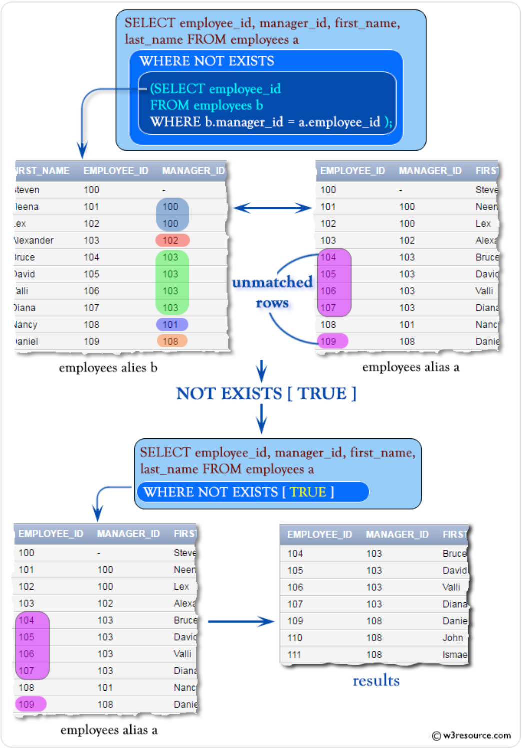
product_id	stock_count
188	86
64	82
109	79
196	79
61	78
182	77
166	77
219	75
142	75
252	75

2. CO RELATED SUBQUERIES

- When ever two sensors within 10 meters of each other simultaneously detect an abnormal temperature.

3. SNAPSHOT QUERIES

- Retrieve every 10 minutes the maximum rainfall level in each county in Southern California, if it is greater than 3.0 inches.



4.HISTORICAL QUERIES:

- Aggregate information over historical data.
- EXAMPLE:
  - SUM
  - AVG
  - MIN
  - MAX
- To find the employee who earned the salary above 70,000

```sql
-- SQL Server UNION Example
SELECT [EmpID]
      ,[FirstName] + ' ' +[LastName] AS Name
      ,[Education], [Occupation]
      ,[YearlyIncome], [Sales]
  FROM [Employees 2015]
  WHERE YearlyIncome >= 70000
UNION
SELECT [EmpID]
      ,[FirstName] + ' ' +[LastName] AS Name
      ,[Education], [Occupation]
      ,[YearlyIncome], [Sales]
  FROM [Employees 2016]
  WHERE Occupation = 'Professional' OR
        Occupation = 'Management'
  ORDER BY Occupation DESC
```

100 %

Results | Messages

tutorialgateway.org

	EmpID	Name	Education	Occupation	YearlyIncome	Sales
1	1	John Yang	Bachelors	Professional	90000	3578.27
2	4	Christy Zhu	Bachelors	Professional	80000	3078.27
3	6	John Ruiz	Bachelors	Professional	70000	539.99
4	11	Gail Erickson	Education	Professional	90000	4319.99
5	2	Rob Johnson	Bachelors	Management	80000	3399.99
6	7	John Miller	Masters Degree	Management	80000	2320.49
7	10	Christy Carlson	Graduate Degree	Management	70000	2234.99
8	12	Barry Johnson	Education	Management	80000	4968.59

Query executed successfu... | PRASAD (12.0 SP1) | PRASAD\suresh (52) | SQL Tutorial | 00:00:00 | 8 rows

5.LONG RUNNING OR CONTINUOUS QUERIES:

- REPORT RESULT OVER an extended time window .
- Example:
  - for next 3 hours retrieves every 10 minutes the rainfall level in California.

EXAMPLE

- consider the following flood warning system as an example of SQL-style querying of sensor networks
- A user from a state emergency management agency may send a query to the flood sensor database.
- "For the next three hours, retrieve every 10 minutes the maximum rainfall level in each county in Southern California, if it is greater than 3.0 inches." This is an example of a long-running, monitoring query, and can be expressed in the following SQL-like syntax:

QUEREY:

- SELECT max(Rainfall_ Level), FROM sensors
- WHERE state = California
- GROUP BY county
- HAVING max(Rainfall_ Level) > 3.0in DURATION [now, now+180 min] SAMPLING PERIOD 10 min

SUMMARIZE THE QUERIES ON SENSOR NETWORKS

- Aggregate over a group of sensor or a time window.
- Contain condition restricting the set of sensors from contributing data.
- Correlate data from different sensor.
- Trigger data collections or single processing on data network.
- Spawn subqueries as necessary.

QUERY PROCESSING

- To request the physical attributes in a region of interest, a user can pose a set of queries to the BS, which is then disseminated by the BS to the underlying sensor network.
- As in conventional database systems, a query describes a logical set of data that the user is interested in, but does not describe the actual protocol and software modules that the system uses to collect the answer set.
- A query processing technique is then applied based on the nature of the query.

- The system can choose from these techniques and operator orderings for any given logical query.
  - For example, to find the average temperature of a specific subregion of the network, the system may collect readings from every sensor node in the network and then filter the list of collected readings for the particular subregion the user is interested in.

QUERY CHARACTERISTICS

- Query Request. A query request conveys a query from the BS to the sensor nodes in a region of interest.
- Query Response. A query response carries a query answer back to the BS for further processing by the user.

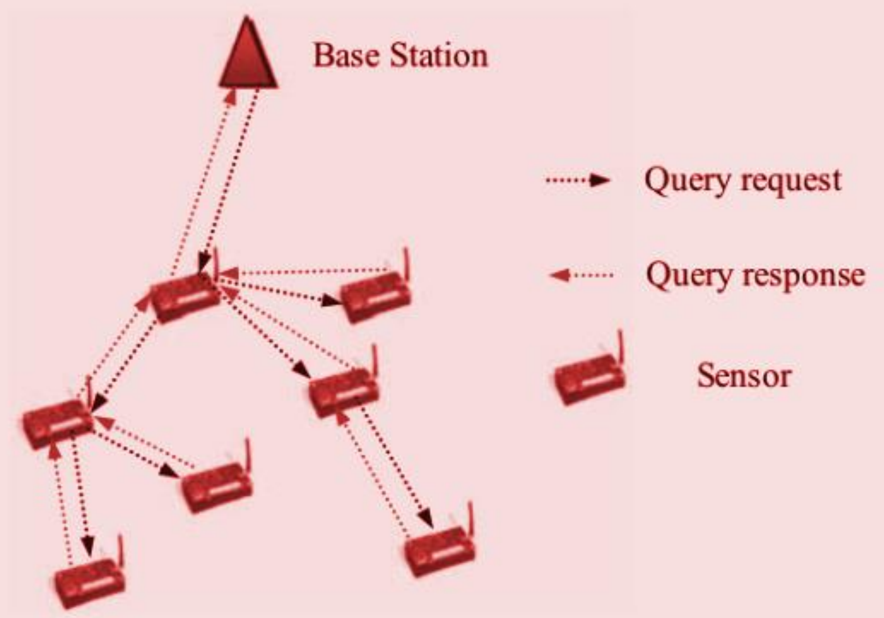


Fig. 7.1 Query processing components.

QUERY OPERATORS

- Sensor queries involve two types of data: stored data and sensor queries, which can be termed as relations (a predefined table format for storing data) and sequences, respectively, in database terminology.
- A sensor query can be defined as an acyclic graph of relational and sequence operators
- The inputs of a relational operator are base relations or the output of another relational operator, whereas the inputs of a sequence operator are base sequences or the output of another sequence operator

QUERY CLASSIfiCATION

- criterion 1: Frequency of Query Response
- Historical Query. A n historical query is mainly used for analysis of historical data stored at a remote BS or any specifi c node in the network in the absence of a BS.
- One - Time Query. A one - time query provides an instantaneous (or snap-shot) view of the network. If a user wants to know data at a specifi c instance, the query is posed just once and data is returned only for that particular instance. It has the same start and end time.

PERSISTENT QUERY

- A persistent query is used to monitor a network for a continuous period of time. It is used when a user wants to know data periodically starting from a particular moment to a logically infinite time. In this case, the start time can be " now " and the end time is " infinite " if the user wants to know the result starting at the time the query is generated.

CRITERION 2: NATURE OF SEARCH SPACE

- Spatial Query. A spatial query looks for a particular attribute value occur- ring in a given space of interest. For example, for a SQL- type query, this implies SELECT attr_val FROM sensor WHERE loc = (20, 30).
- Temporal Query. A temporal query looks for a particular attribute value occurring during a specified period of time. For example, for a SQL - type query, this implies SELECT attr_val FROM sensor WHERE time = 6:50 am – 2:30 am.
- Spatio - Temporal Query. A n example of spatio- temporal queries is SELECT attr_val FROM sensor WHERE loc = (40, 25) and time = 1:30 pm – 4:30 pm.

CRITERION 3: SEMANTIC NATURE OF QUERY RESPONSE

- Exploratory Query. An exploratory query retrieves a single record from the database. An example of exploratory queries is Return the record on 01- 18 - 07 at 5:00 pm .
- Monitoring Query. A monitoring query is defined as a more complex query that searches in the database and returns more than one record within the same RDF file (a fi le format that makes the semantic network information machine readable). An example of monitoring queries is Return all records for 01 - 18 - 07 .
- Range Query. A range query requires importing several RDF files into a particular database model and returns the data satisfying the query in each file. An example of range queries is Return all days in the month of March when only fan number 1 was on .

CRITERION 4: RANGE OF DATA IN QUERY RESPONSE

- Filtering Query. A filtering query returns sensor data only if it is within a specified range or has a condition to be satisfied. In case of a filtering query with a range specified, data that are out of range are filtered out and results only satisfying the range condition are returned back to the user.
- Non - filtering Query. A non - filtering query does not have any condition and it returns all raw sensor data.

CRITERION 5: NUMBER OF ATTRIBUTES IN QUERY REQUEST

- One - Dimensional Query. A one - dimensional (1D) query requests results with only one type of attributes or sensed data. An example of 1D queries is Return all the data sensed in the temperature range 30– 4 5 ° C .
- Multidimensional Query. A multidimensional query requests two or more types of attributes or sensed data. An example of multidimensional queries is Return all the data sensed in the temperature range of 30– 4 5 ° C and humidity of 80 – 85% .

CHALLENGES IN QUERY PROCESSING

- Query Broadcast- the query originator node receives the relevant data from all nodes, it can get a complete answer to the specific query
- Querying Interface- The interface should also allow users to collect information and process the information in an efficient way.
- Efficient Database for Querying- A language needs to be developed for querying and task assignments, as well as a database that can be readily queried.

- Uncertainty and Transience in Sensor Readings- Since most of the attributes sensed by sensor nodes are environmental parameters, they are associated with some inherent uncertainty caused by noise in the environment.
- Probabilistic Queries and Answers- Since a WSN cannot obtain all possible data, any reading from a sensor is approximate or probabilistic, that is, it only represents the true state of the world at the distinct instants and locations where sampling was performed.

SENSOR SELECTION FOR QUERY PROCESSING

- Each sensor node consists of one or more sensors attached to it and connected to the physical world.
- The types of sensors can be classified into temperature sensors, light sensors, or pressure sensors that can determine the occurrence of events (e.g., the sudden change in their readings) in their surrounding area.
- Therefore, each sensor can be considered as a separate data source that generates records with different fields

QUERY PROCESSING TECHNIQUES

- here are broadly two techniques for processing sensor queries: the warehousing technique and the distributed technique.
- In the warehousing technique, processing of sensor queries is kept completely separate from the access to the underlying sensor network, where the actual data is sensed and stored.
- In the distributed technique, the query workload determines the data that should be extracted from sensors. The distributed technique is more flexible because it enables different queries to extract different data from the sensor network.

QUERY FLOODING

- A query can be flooded among all nodes in a WSN; that is, broadcast to all the nodes reachable from the BS.

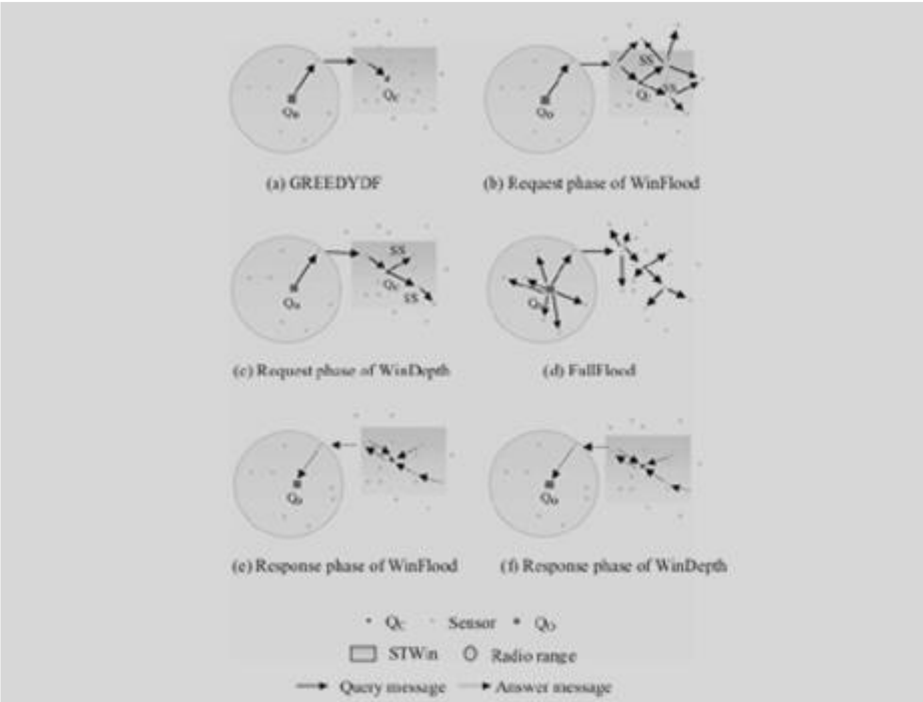


Fig 7.2 Illustration of basic methodology for query processing.

SNAPSHOT QUERYING

- WSNs are prone to node failures due to nodes' running out of their battery.
- Therefore, it is important to use a data - centric network in such scenarios, which allows access to the collected measurements in an integrated manner.
- In case of a node failure, the network should be able to self - heal by using surplus stand - by nodes to keep the network running.
- Thus, nodes should be able to generate a model of their surrounding environment so that they can represent the sensed data of the entire network.

DATA AGGREGATION

- Data aggregation is an effective technique for removing data redundancy and improving energy efficiency in WSNs. The basic idea is to combine the data received from different sources so that the redundancy in the data is minimized and the energy consumption for transmitting the data is reduced in the aggregation process.
- the locations of reporting sensor nodes, which can be geographical coordinates, should not be left out in performing data aggregation

CHALLENGES IN DATA AGGREGATION

- Energy Efficiency: Since sensor nodes have limited battery power, efficient sleep scheduling protocols should be employed for saving the energy of idle sensor nodes.
- Timing Control: In addition to saving energy, it is also important to perform data aggregation within reasonable time bounds.
- Application Orientation: A WSN is usually designed with a specific application in mind.
- QoS Support: In addition to energy efficiency and timing control, some- times it is also necessary to meet specific application requirements in terms of quality of severe (QoS) in data aggregation.

DATA AGGREGATION TECHNIQUES

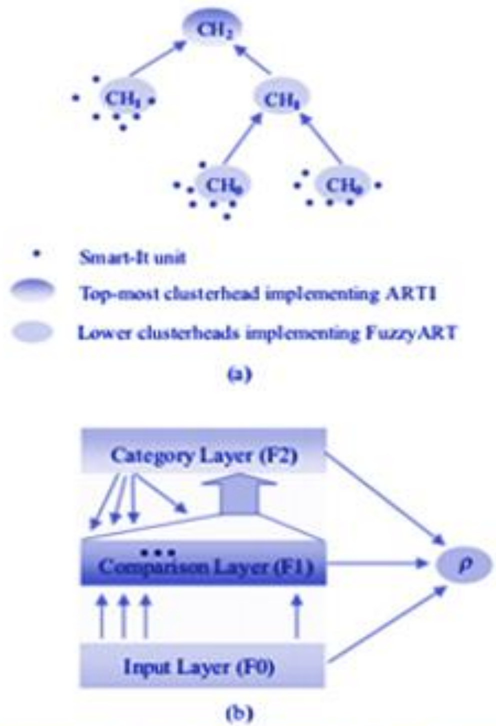
- Energy – Efficient Data Aggregation.
- Neural - Network - Based Data Aggregation
- Delay -Constrained Data Aggregation.
- Q o S Constrained Data Aggregation
- Data Aggregation for Range Query
- Structure-Free Data Aggregation

ENERGY – EFFICIENT DATA AGGREGATION.

- Use of Simple Mathematical Operators: Since energy is one of the most crucial parameters for a longer network lifetime, most of the research efforts have been directed to this issue
- Aggregating Object Reports at Source   Another approach is to aggregate the object reports right at the source node and then send the object report to the BS for processing

NEURAL - NETWORK - BASED DATA AGGREGATION

- Data aggregation can be achieved in an innovative manner by combined knowledge of artificial intelligence and networking concepts. A possible technique for compressing redundant data is to perform dimensionality reduction . This can be done in some scenarios by employing a neural - network - based algorithm, which runs based on an iterative pattern of adjusting weights to learn irregular inputs.



(a) Data collection by CHs from lower clusters. (b) Three internal layers of each CH.

DELAY -CONSTRAINED DATA AGGREGATION

- I n data aggregation, a sensor node needs to combine its own data and the data received from its 1- hop neighbors, and then send the combined data to the BS. Hence, the node needs to wait for its neighbors ' data before transmitting to the BS.
- The delay of data aggregation can be reduced by employing a finite state machine (FSM) based feedback control scheme

FSM BASED FEEDBACK CONTROL SCHEME

- he waiting time at each level of the DAT tree to respond to a query is adjusted according to the output of the FSM, while maintaining a substantial level of accuracy.
- Variants of FSM Based Scheme

$$T\,n + 1 = Tn - 1\;if\;Nrec > Nopt,$$
$$T\,n - 1 = Tn + 1\;if\;Nrec > Nopt$$

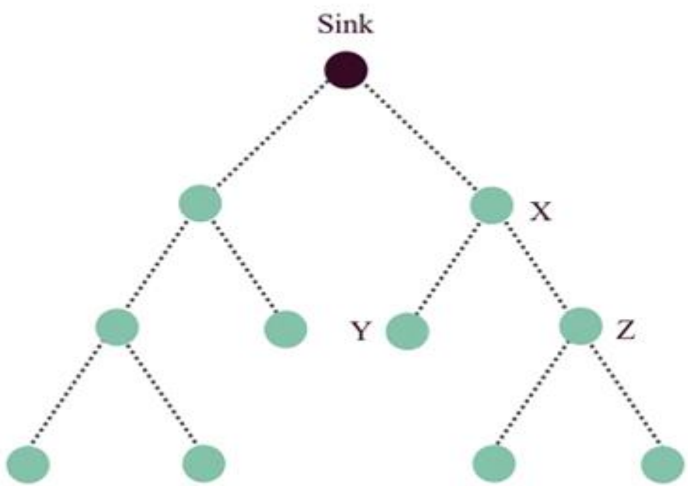


Fig. 7.6 Data aggregation tree.

QOS CONSTRAINED DATA AGGREGATION

- Balancing Trade – Offs: data aggregation scheme is also proposed to balance the trade - offs among energy – efficiency, delay requirement, accuracy, and buffer overflow probability
- Packet Delivery Probability: Energy consumption decreases as the deferred period increases. This happens as the average number of packets that can be used for data aggregation increases with an increase in the deferred aggregation period.

DATA AGGREGATION FOR RANGE QUERY.

- The network consists of mobile sinks (equipped with PDAs), which request nearby static sensors to sense event data, and a specific sensor in a grid, which has knowledge about the event, is designated as the head of the grid. For a regular- shape range query, a mobile sink designates the range (e.g., a range of rectangle) for data aggregation, and requests the source (i.e., a static sensor sensing data) to collect the data in the designated range.
- A cache mechanism is used to resolve the identical queries issued from mobile sinks. The protocol can reduce the overall energy consumption of the network.

STRUCTURE-FREE DATA AGGREGATION

- Data aggregation can be performed based on an aggregation structure or aggregation tree in a hierarchical manner
- maintaining a hierarchical structure introduces additional cost, especially in a dynamic environment with mobile sensors.
- The DAA protocol is proposed to improve spatial convergence while the RW technique is proposed to improve temporal convergence.
- DAA + R W can efficiently aggregate packets and decrease the number of data packets transmitted in the network without need for any preconstructed structure.

COUGAR SENSOR DATABASE AND ABSTRACT DATA TYPES

- Abstract data type as in most modern object-relational database
- An ADT provides controlled access to encapsulated data through a well defined set of access functions.
- Examples:
  - The public interface of a seismic sensor ADT may comprise signal processing functions, such as short-time Fourier transform and vibration signature analysis.

Probabilistic Queries

- As attribute values become uncertain (actually, imprecise), operators (e.g =, <,>) over these data need to be defined.
- These operators may no longer return Boolean results. Instead, given the probability distributions, they can return *probabilistic answers*

Sensor Errors

- In the previous examples, uncertainty was introduced in order to avoid incorrect results
- Uncertainty may be inherent due to measurement errors, e.g.
  - Most scientific instruments have well known errors
  - GPS has a Gaussian distribution
  - Micro-array data have a Lorentzian distribution
  - Statistical results also have margins of error
- Similar to previous case

Probabilistic Models

- There are two main types of probabilistic data uncertainty addressed in recent work:
  - Attribute uncertainty
    - The value of an attribute of a tuple is not known precisely
    - Modeled as a set or range of possible values with associated probabilities
  - Tuple uncertainty
    - The membership (presence) of an entire tuple within a relation is uncertain
    - Maybe modeled as an probability attached to the tuple.

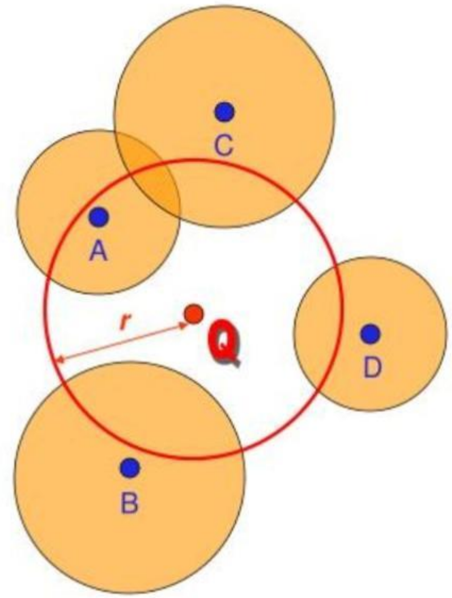
Probabilistic Data Model

- Barbara, Garcia-Molina, Porter [BGP92]
- Discrete attribute uncertainty
- Key attributes are deterministic (precise)
- Notion of attribute groups (handles dependent data)
- Captures missing probability (no assumption)
- Probabilities may be user defined, statistically determined, due to staleness, etc.

STUDENT	GPA	INTEREST	ACC_EVAL
Adam	3.8	0.7[theory]	0.6[Y A]
		0.3[*]	0.1[N A]
			0.3[* *]

Probabilistic Nearest Neighbor Query

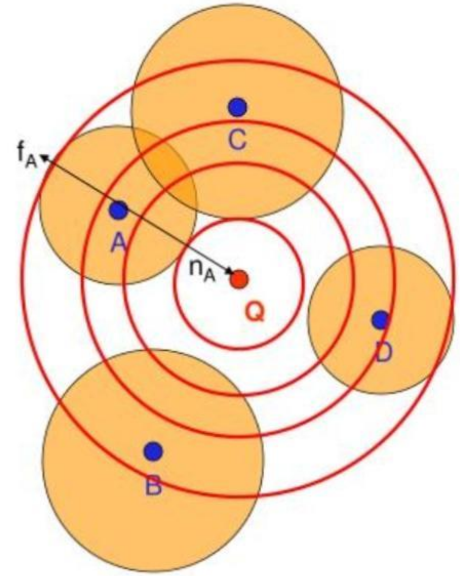
- At distance r, A is the nearest neighbor of Q if:
  - A is at distance r from Q
  - B,C,D are all located at distances $> r$ from Q.
  - The *pdf* $p_A(r)$ can be computed.



Probabilistic Nearest Neighbor Query

- Compute $p_A(r)$
  - From the shortest distance of *A* to *Q* (n_A)
  - To the longest distance of *A* to *Q* (f_A)

$$P_A = \int_{n_A}^{f_A} p_A(r)dr$$



Classification of Probabilistic Queries

	Continuous	*Discrete*
Independent	What is the temperature of sensor *x*?	Which sensor has temp between 10°F and 30°F?
Inter-dependent	What is the average temperature of the sensors?	Which sensor gives the highest temperature?

The notion of query answer quality was also introduced.

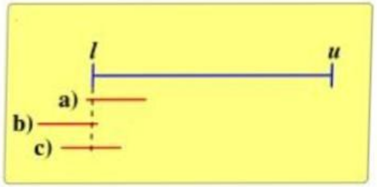
For each class of queries, a metric for query quality was specified.

Intuitively, this metric captures the degree of uncertainty in the answer (as compared to an answer derived over precise data).

Quality of Probabilistic Result

- Probabilistic queries: notion of result "quality"
- Example: range query (is Ti.z in range [l, u]?)
  - regular range query
    - "yes" or "no"
  - probabilistic range query

$$Score = \frac{|p_i - 0.5|}{0.5}$$



$$Score\_of\_an\_ERQ = \frac{1}{|R|}\sum_{i \in R}\frac{|p_i - 0.5|}{0.5}$$

Introduction

- DBMS stands for Database Management System.

- DBMS is a software system for creating, organizing and managing the database.

- It provides an environment to the user to perform operations on the database for creation, insertion, deletion, updating and retrieval of data.

What is Data ?

- A collection of raw facts and figures.
- Raw material that can be processed by any computing machine.
- A collection of facts from which conclusions may be drawn.
- Data can be represented in the form of numbers and words which can be stored in computer's language.

  i.e. Paan Singh, Anshul 007

Database

- A repository of logically related and similar data.
- An organized collection of related information so that it can easily be accessed, managed and updated.

  E.g.:

  - Dictionary
  - Airline Database
  - Student Database
  - Library
  - Railways Timetable
  - YouTube (All songs of Rahul Vaidya)

Data Models, Schema and Instances

- **Data Models:**

  -Describes structure of the database.

  -Aim is to support the development of information systems by providing the definition and format of data.

  -If the same data structures are used to store and access data then different applications can share data.

  -Classification:

  1. High-Level Model
  2. Representation Model
  3. Low-Level Model

1. High-Level Model

- Ensures data requirement of the users.
- Not concerned with representation, but it's a conceptual form.
- Three Imp terms:

  a)Entity: - Any object, exists physically or conceptually.

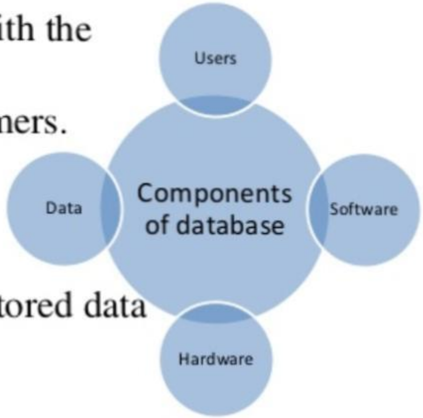
  b)Attribute:- Property or characteristic of entity.

  c)Relationship:- Association or link b/w two entities.

- These 3 terms make Entity-Relationship Model.

Components of Database System

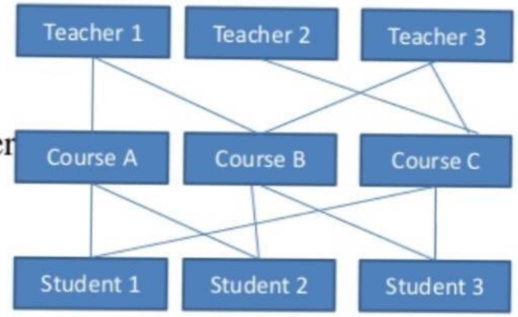
- **Users**- People who interact with the database:
  - Application Programmers.
  - End Users.
  - Data Administrators.

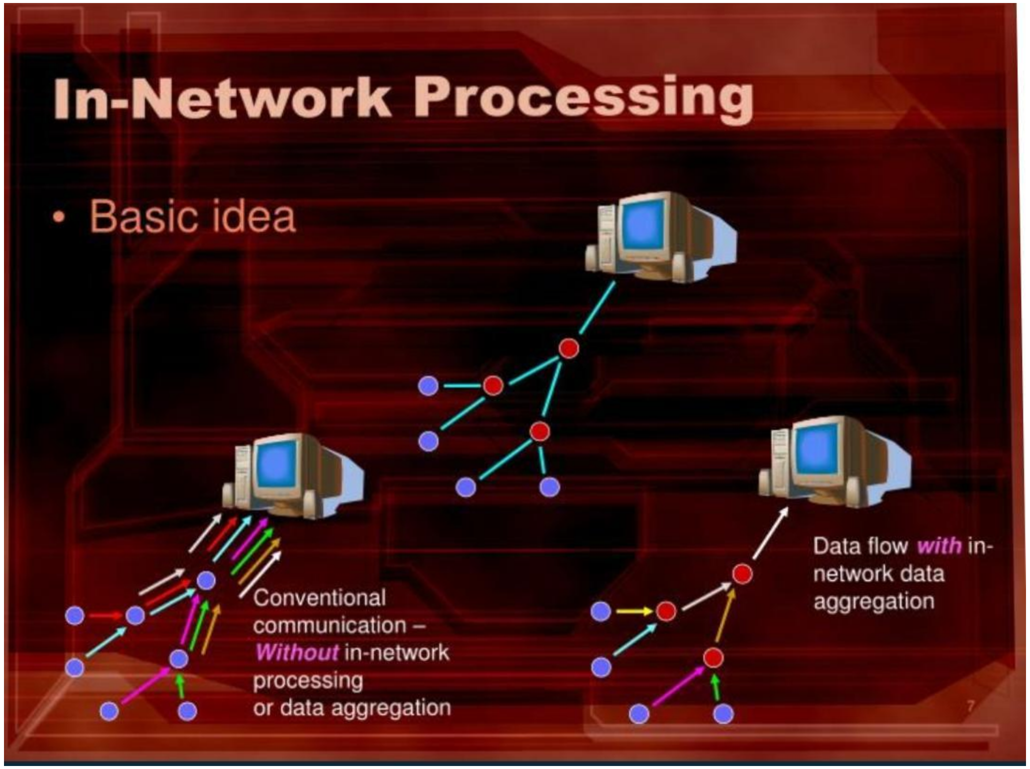


- **Software**- Lies between the stored data and the users:
  - DBMS.
  - Application Software.
  - User Interface.

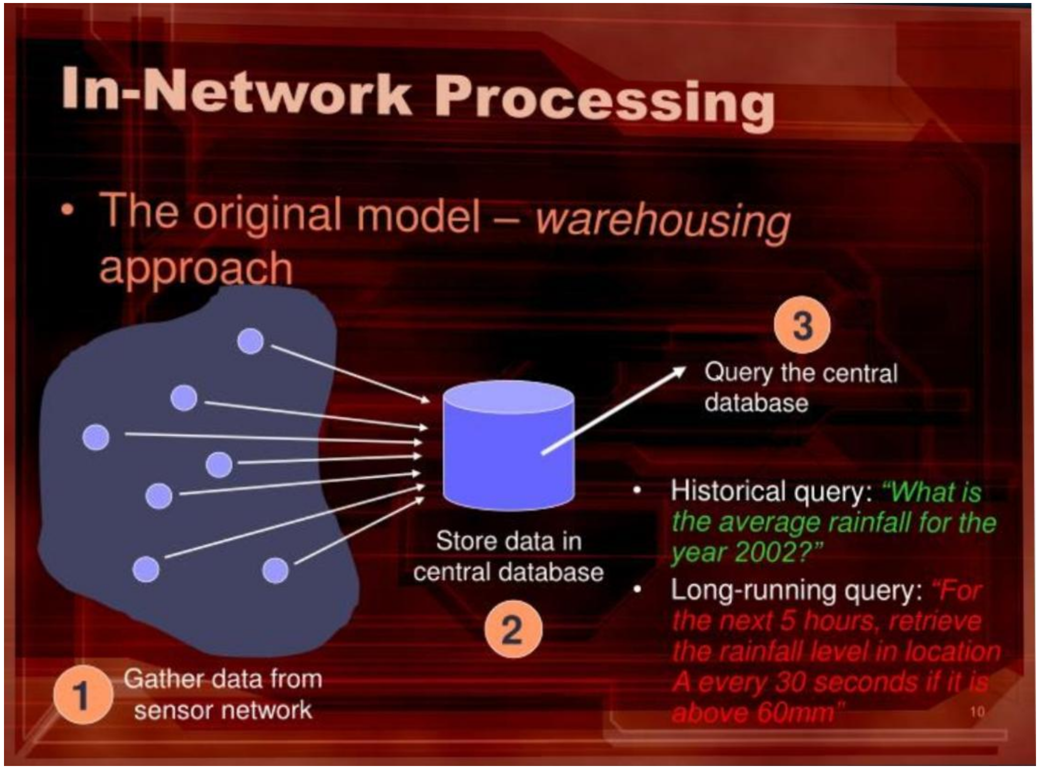
c. Network Database Model

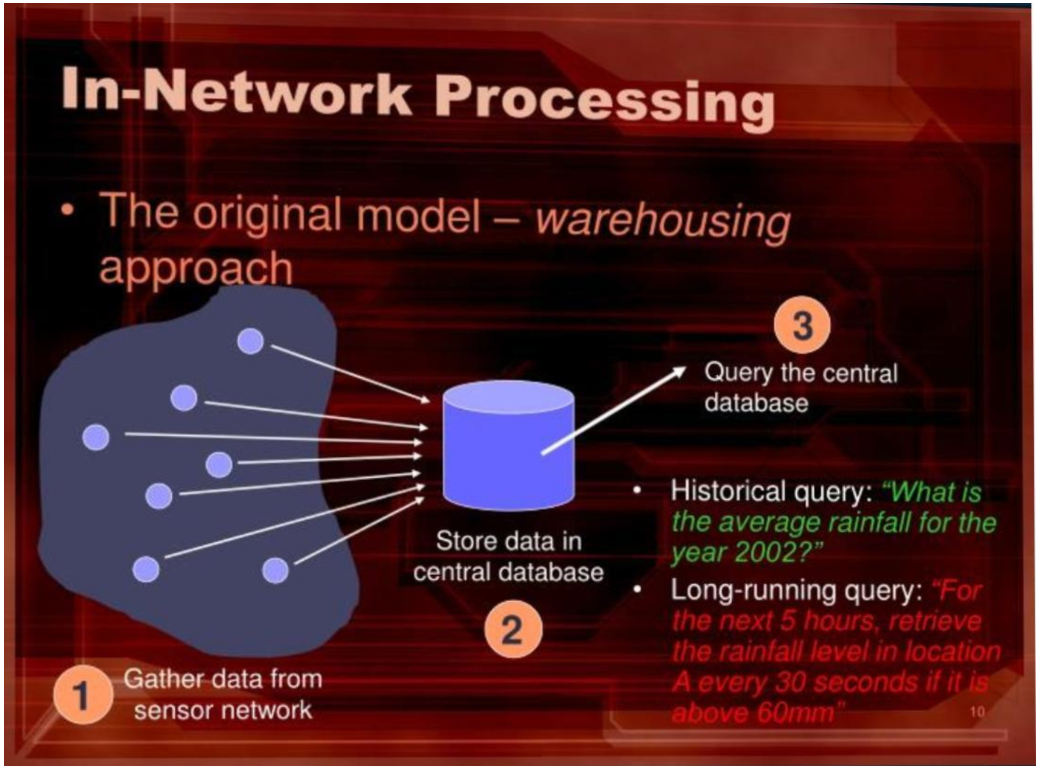
- Represented using a Data-Structure Diagram.
- Boxes represents the records & lines the links.
- Based on 'owner-member relationship.'
- Members of an owner may be many but for many member owner is one.
- Can represent one-to-one and many-to-many as well.

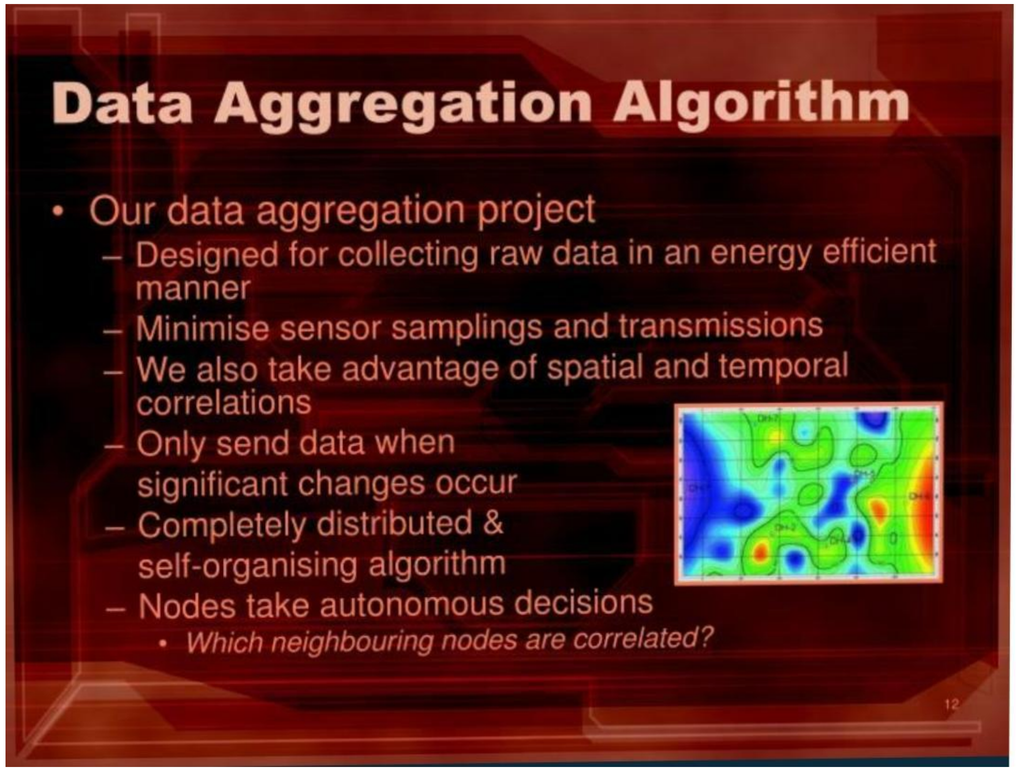


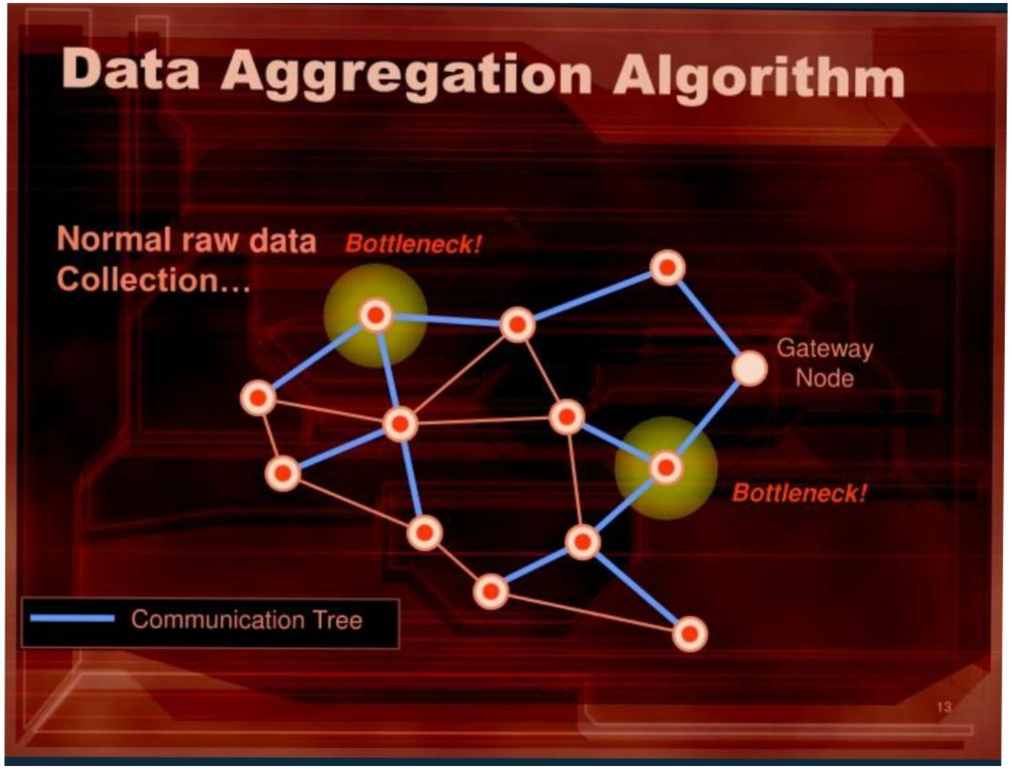


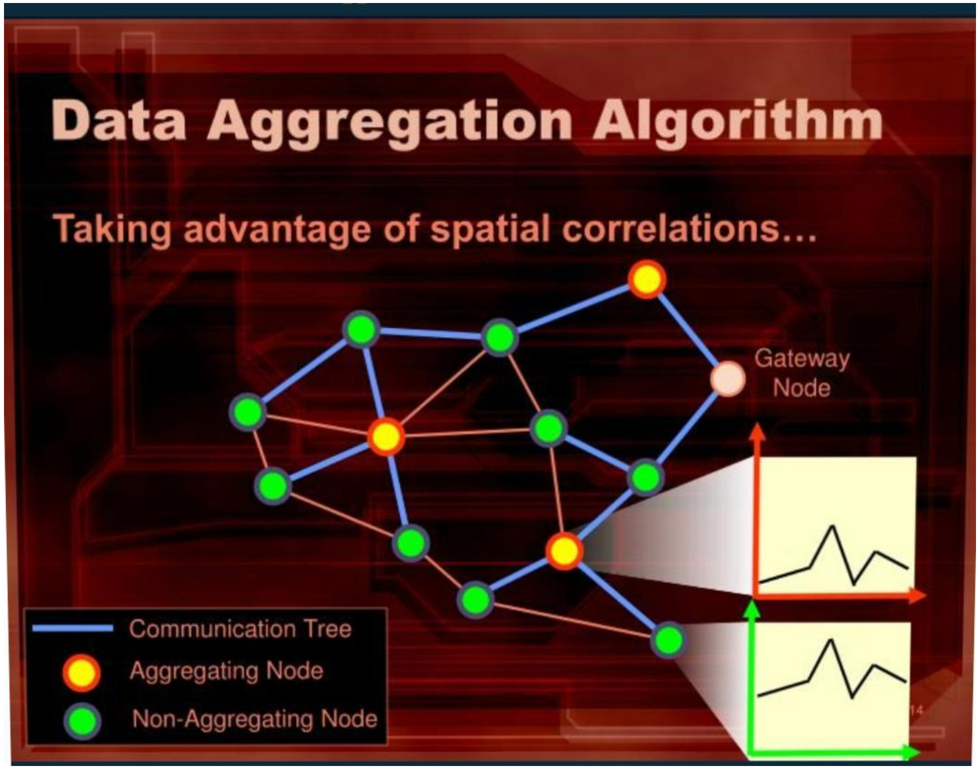
In-Network Processing
• Basic idea
Conventional communication – Without in-network processing or data aggregation
Data flow with in-network data aggregation
7

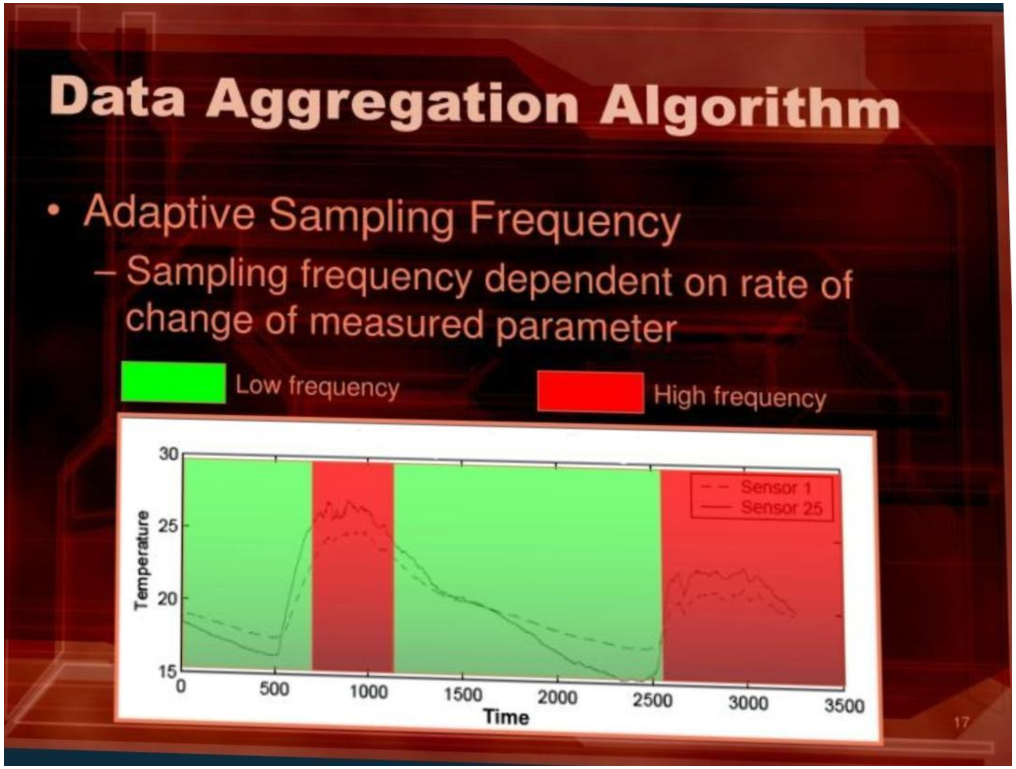

In-Network Processing
• The original model – warehousing approach
Query the central database
Store data in central database
Gather data from sensor network
1
2
3
• Historical query: "What is the average rainfall for the year 2002?"
• Long-running query: "For the next 5 hours, retrieve the rainfall level in location A every 30 seconds if it is above 60mm"
10


In-Network Processing
• The original model – warehousing approach
3
Query the central database
Store data in central database
2
• Historical query: "What is the average rainfall for the year 2002?"
• Long-running query: "For the next 5 hours, retrieve the rainfall level in location A every 30 seconds if it is above 60mm"
1
Gather data from sensor network
10


Data Aggregation Algorithm
• Our data aggregation project
– Designed for collecting raw data in an energy efficient manner
– Minimise sensor samplings and transmissions
– We also take advantage of spatial and temporal correlations
– Only send data when significant changes occur
– Completely distributed & self-organising algorithm
– Nodes take autonomous decisions
• Which neighbouring nodes are correlated?
12


Data Aggregation Algorithm
Normal raw data
Collection…
Bottleneck!
Gateway
Node
Bottleneck!
Communication Tree
13


Data Aggregation Algorithm
Taking advantage of spatial correlations…
Gateway
Node
Communication Tree
Aggregating Node
Non-Aggregating Node


Data Aggregation Algorithm
• Adaptive Sampling Frequency
– Sampling frequency dependent on rate of change of measured parameter
Low frequency
High frequency
Temperature
Time
Sensor 1
Sensor 25
30
25
20
15
0
500
1000
1500
2000
2500
3000
3500
17

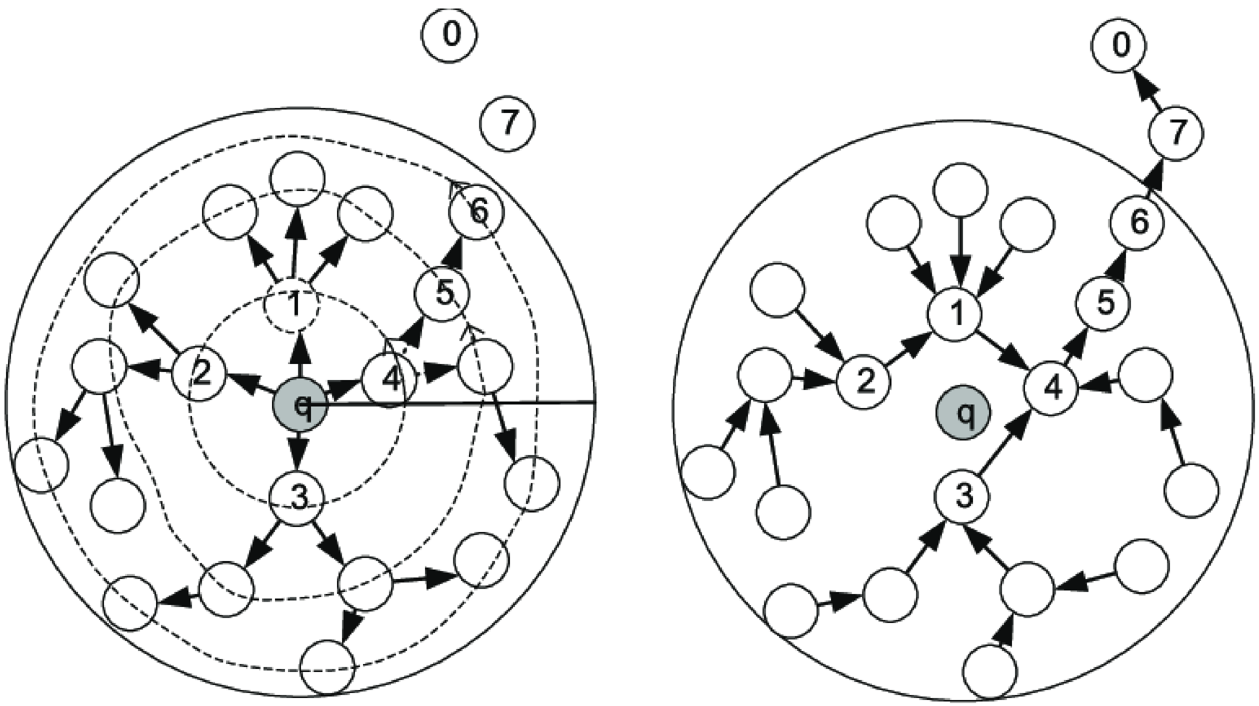
QUERY PROPAGATION AND AGGREGATION

- Query propagation(Distribution).
- Query Aggregation(collection).
- In server based approach, where the the aggregation occurs at an external server, each sensor sends its data directly to the server.
- It Requires a total of 16 transmissions.
- Alternatively, every sensor may compute a partial state Record.

- (SUM,COUNT) based on its data and that of its children if there are any .This requires a total of only the message transmissions.
- To push a query node in a network an efficient routing structure has to be established
- For example: A routing tree rooted at a base station.
- A query may be propagated through the rooting using a Broadcast Mechanism.

- It may use multicast to reach only those nodes that may contribute to the query.
- Once a query has been distributed all the nodes that satisfy the query conditions data is then collected and aggregate within the network.
- Snapshot query such as "Report the current maximum temperature" in a region –short span aggregation.
- For application query the nodes must Synchronize more accurately in order to correctly return the result.
- A key Challenge In network Aggregation is the Design of an optimal data aggregation schedule that is energy and time efficient.

QUERY PROPAGATION AND AGGREGATION



(a) (b)

TINYDB

- Tiny DB is a simple protocol which provides (SQL) Interface
- Allows user to query the network declarative (Retrieves data from one or more sensor nodes).
- It collects data from sensor nodes, filters and aggregate it together, and send it to the base station via an energy efficient in network processing algorithm.

SQL OPERATORS

- It supports five SQL operators.
- Sum,Count,Max,Min,Average.
- Two extensions (Median, Histogram).
- TinyDB :
  - Declarative query interface.
  - Efficient and extensible framework.
  - Open-source implementation on real nodes.

QUERY PROCESSING SYSTEMS FOR WSN

- Provide Abstractions for SQL Query Interfaces.
- Represents Sensors or sensor network as a table.
- User inserts query at base station and it converts those queries into sensor node understandable format.
- Based on Relational Database Management system.
- Some popular Query Processing system:
- Tiny DB (for tiny OS).
- Tikin DB (for Contiki).

GOAL OF SCHEDULING

- Communication efficiency and reliability
  - Coordinate nodes in communication
    - Wireless collisions among neighbors
    - No receiving on sleeping nodes

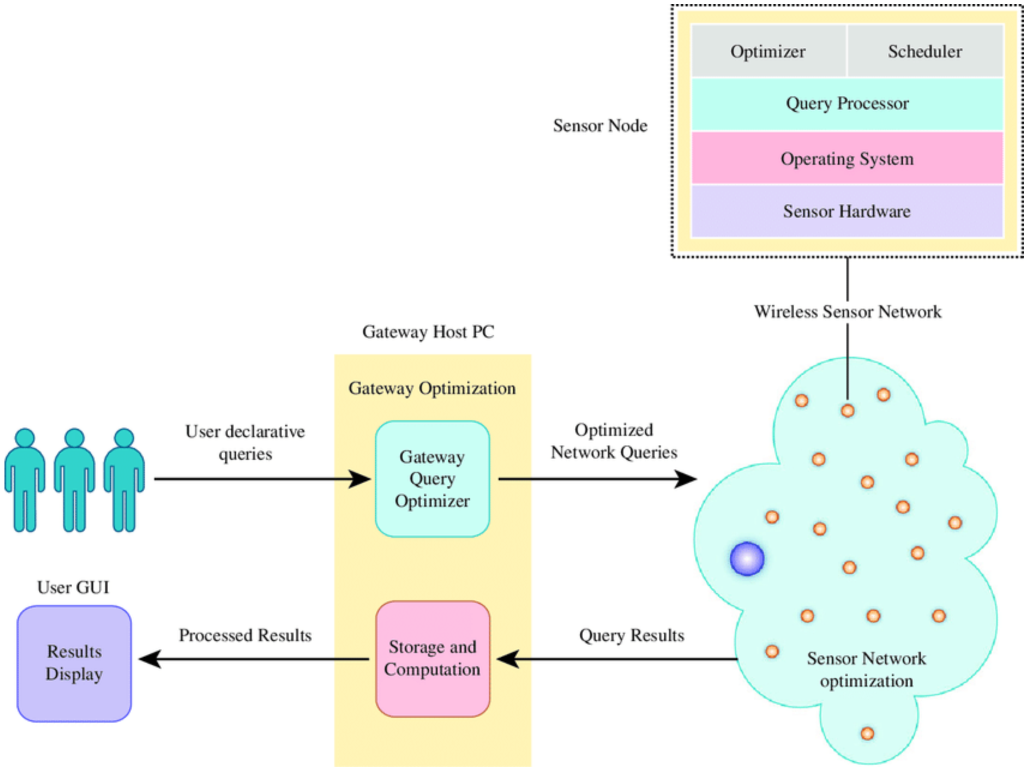
SCHEDULING AND SENSOR DATABASES

- SensorDB's are complex to schedule.
- Opportunities in scheduling for sensorDBs
- On-node multi-query scheduling
- Limited resources
- Changing sensor environments
- Query-aware transmission scheduling
- Interaction between Scheduling and Query Execution.

SUMMARY OF SCHEDULING

- Scheduling is done on one or more layers.
- Scheduling is crucial for performance.
  - Communication reliability
  - Coordination of nodes
  - Energy consumption
  - Sleep scheduling
  - Response time
  - Different transmission timings of neighboring nodes result in different delays.

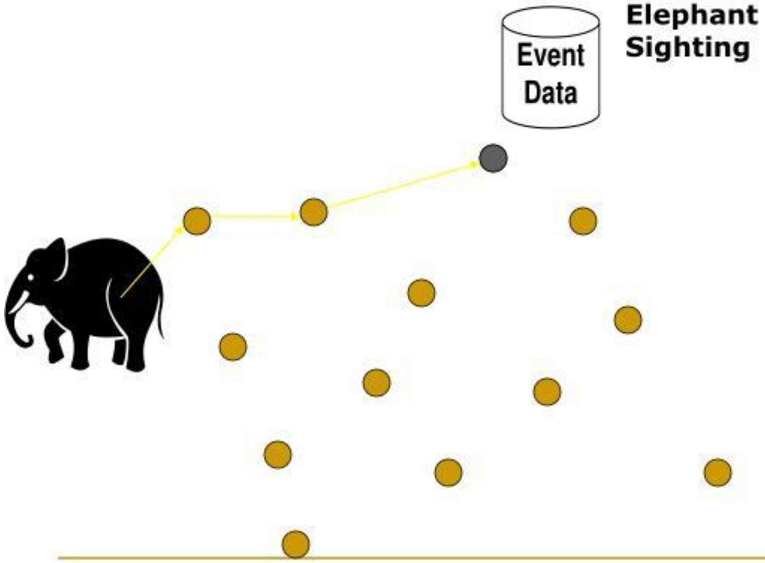
OPTIMIZATION QUERY PROCESSING



DATA-CENTRIC STORAGE

- Flexible storage needed when queries can originate from within network
  - Server-based approaches flood queries to all nodes
  - Data-Centric Storage (DCS) make use of rendezvous points to aggregate queries and data

Data centric Storage



Introduction

- Sensornet
  - ♦ A distributed network comprised of a large number of small sensing devices equipped with
    - • Computation • Communication • Storage
  - ♦ Great volume of data
- Data Dissemination Algorithm
  - ♦ Energy efficient
  - ♦ Scalable
  - ♦ Self-organizing

Keywords and Terminology

- Observation
  - low-level readings from sensors
  - e.g. Detailed temperature readings
- Events
  - Predefined constellations of low-level observations
  - e.g. temperature greater than 75 F
- Queries
  - Used to elicit information from sensor network

Total Usage /Hotspot Usage

- Total Usage

  Total number of packets sent in the Sensor network

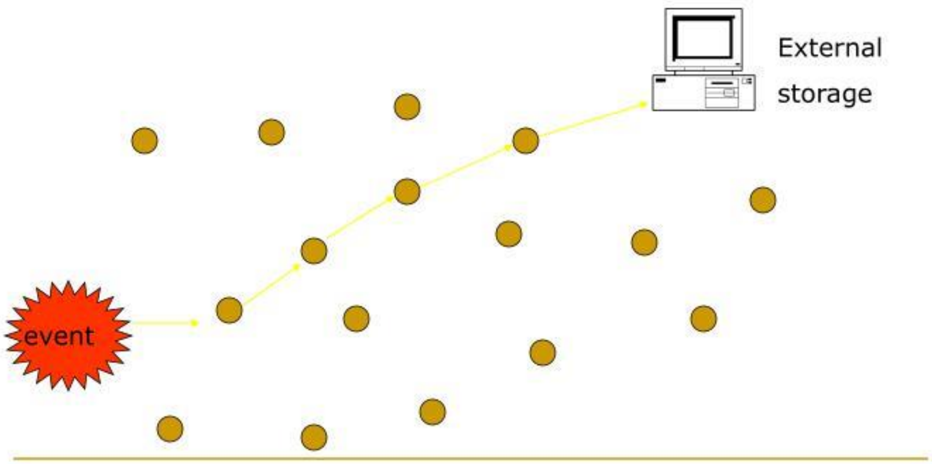
- Hotspot Usage

  The maximal number of packets send by a particular sensor node

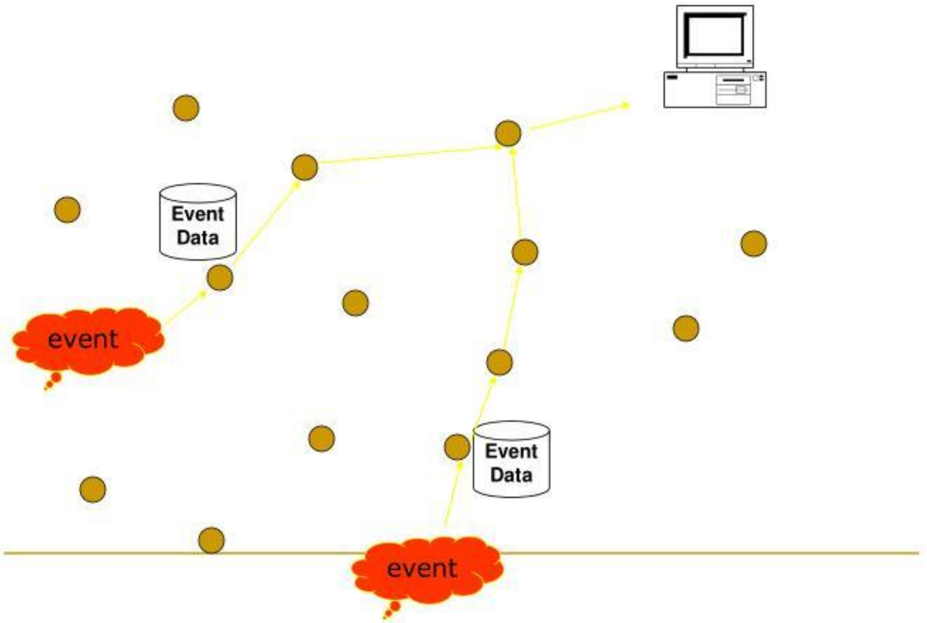
Existing schemes for Storage

- External Storage (ES)
- Local Storage (LS)
- Data Centric Storage (DCS)

External Storage (ES)



Local Storage (LS)



Why do we need DCS?

- Scalability
- Robustness against Node failures and Node mobility
- To achieve Energy-efficiency

Data Centric Storage

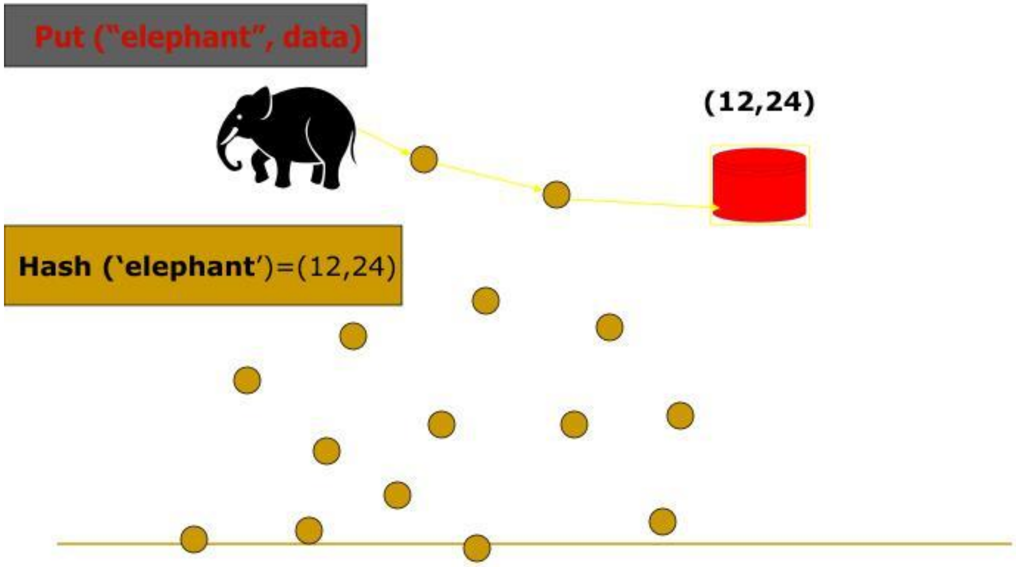
- Relevant Data are stored by “name” at nodes within the Sensor network
- All data with the same general name will be stored at the same sensor-net node.

  e.g. (“elephant sightings”)
- Queries for data with a particular name are then sent directly to the node storing those named data

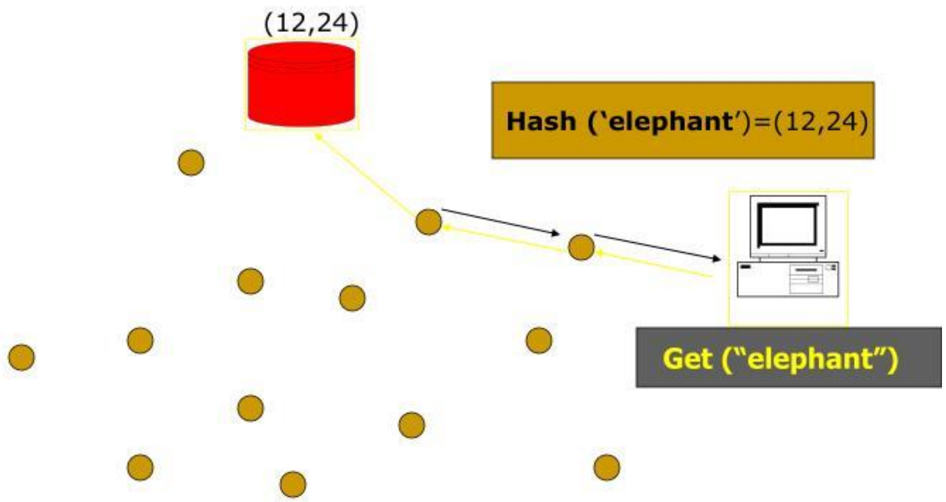
GEOGRAPHICAL HASH TABLE (GHT)

- Mapping from node attribute to storage location by hash function
  - Distributes data evenly across network
  - Nodes know their location
  - Objects have keys
  - Nodes responsible for storing a range of keys
- Geographical routing algorithm
  - Any node can locate storage node for any attribute
- Nodes put and get data to/from storage
  - Hash table interface
- Data replicated at nodes near the storage point for a key

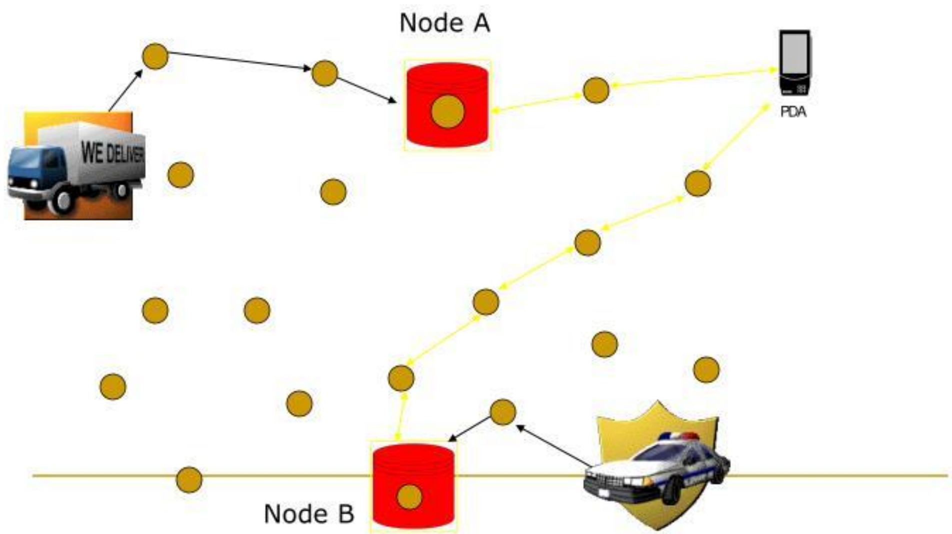
Storing Data in GHT



Retrieving data in GHT



Geographic Hash Table



Algorithms Used By GHT

- Geographic hash Table uses GPSR for Routing

  (Greedy Perimeter stateless routing)

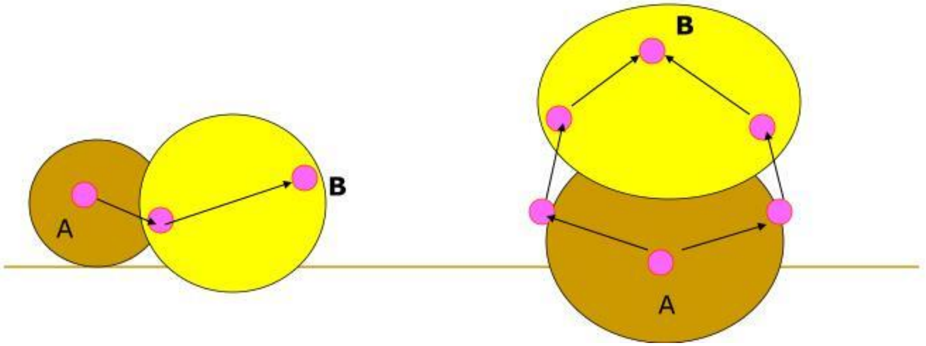
- PEER-TO-PEER look up system

  (data object is associated with key and each node in the system is responsible for storing a certain range of keys)

Algorithm (Contd)

GPSR- Packets are marked with position of destinations and each node is aware of its position

- Greedy forwarding algorithm
- Perimeter forwarding algorithm



DCS TYPES

- Normal DCS – Query returns a separate message for each detected event
- Summarized DCS – Query returns a single message regardless of the number of detected events

(usually summary is preferred)

DATA INDICES & RANGE QUERIES

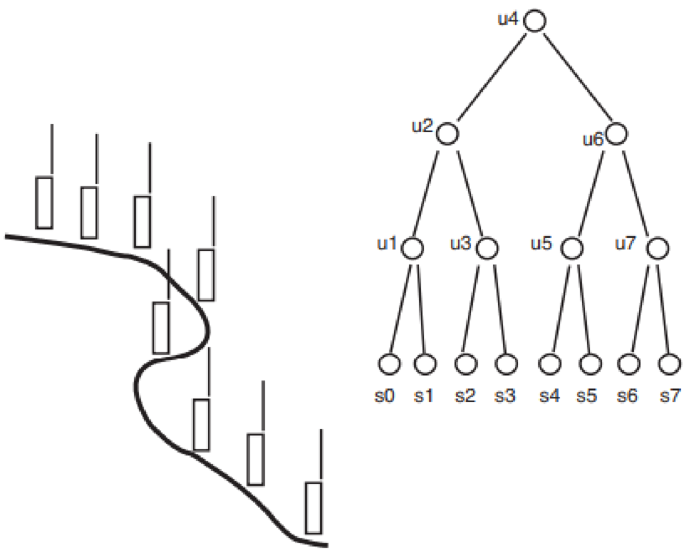
- Sensor network DBs need to support range queries
  - Query specifies value range for each attribute
- GHTs and base TinyDB model not adequate as is
  - Need to index data to support complex queries
- Metrics for measuring index
  - Speed gains for query processing
  - Size of index
  - Costs of building and maintaining index

- A type of query that is especially appropriate for sensor network databases is a range query.
- 3 types:
  - One-Dimensional Indices
  - Multidimensional Indices for Orthogonal Range Searching
  - Non orthogonal Range Searching

ONE-DIMENSIONAL INDICES

- The bulk of indexing research in the database community has dealt with one-dimensional indices
- It is based on B-trees, hash tables, and other structures.
- Indices for range queries have been developed by the computational geometry community.
- In the context of range searching, a key idea is that of pre-string the answer to certain special queries and then delivering the answer.
- The particular subsets of data forming these pre-stored answers are referred to as the canonical subsets.

- Again there is a trade-off between the number of pre-stored answer and the speed of query execution.
- At the end, if nothing is stored, every record has to be queried.
- Over the course of a day, these sensors measure the traffic past them and accumulate traffic counts every minute.
- The road department wishes to decide where the road has to be widened to accommodate increased traffic.



(a) (b)

u_1	$s_0 \oplus s_1$
u_2	$s_0 \oplus s_1 \oplus s_2 \oplus s_3$
u_3	$s_2 \oplus s_3$
u_4	$s_0 \oplus s_1 \oplus s_2 \oplus s_3 \oplus s_4 \oplus s_5 \oplus s_6 \oplus s_7$
u_5	$s_4 \oplus s_5$
u_6	$s_4 \oplus s_5 \oplus s_6 \oplus s_7$
u_7	$s_6 \oplus s_7$

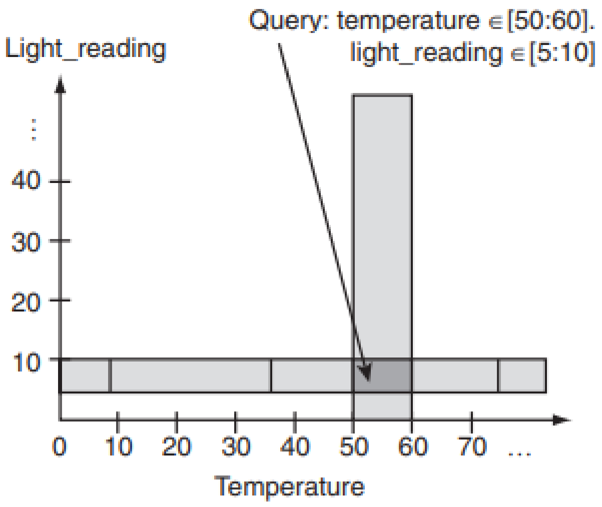
MULTIDIMENSIONAL INDICES FOR ORTHOGONAL RANGE SEARCHING

- Sensor data is rarely indexed by a single attribute
- sensors are typically deployed over a two-dimensional domain.
- Set of attributes is parameterized by scalar value and query with the range along a subset of the parameters.
- A direct use of one-dimensional techniques can very inefficient.
- For ex: we would build two separate one dimensional indices for the petrel-nesting.

- One on temperature and one on light.
- We can then use one-dimensional range searching to retrieve all nesting events where the temperature is in desired range and separately, and where the light is in the desired interval.
- Finally these two set of records can be intersected to produce the final answer.

- For Ex.:

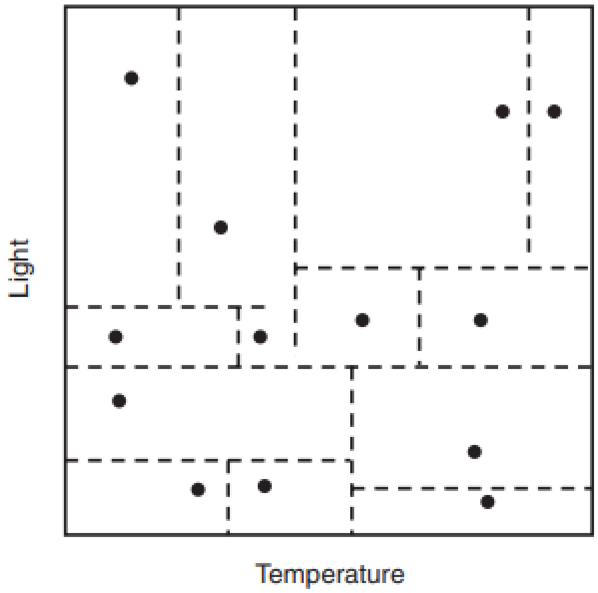
```
SELECT * FROM Nesting_ Events
WHERE Temperature >=50
AND Temperature <= 60
AND Light >=5 AND Light <= 10
```



- Multidimensional indexing techniques have been developed based on both hashing/partitioning schemes and tree structures.
- Grid files and partitioned hashing are examples of the former type.
- With one exception to be mentioned later hash functions aim to spread the data around and not to preserve locality.
- Tree-based index structure include multilevel indices, k-d trees, quad-trees, and R trees.
- All of these can be adapted for multidimensional range searching in the Orthogonal case.

- Let us illustrate this in a two dimensional setting.
- We use k-d trees as an example.
- Consider a small sample of nesting events in our petrel scenario, represented by points in a two dimensional plane parameterized by their temperature and light values.
- Now that each node in this k-d tree represents a rectangular region in the temperature-light plane and can record the count of all the events stored inside it.
- We can drill down the k-d tree with rectangle Q.

- Whenever we reach a node whose corresponding rectangle is disjoint from Q, we can just stop propagating.
- When we encounter a node whose corresponding rectangle is fully contained in Q, we can incorporate its count into a running total for the events of interest.

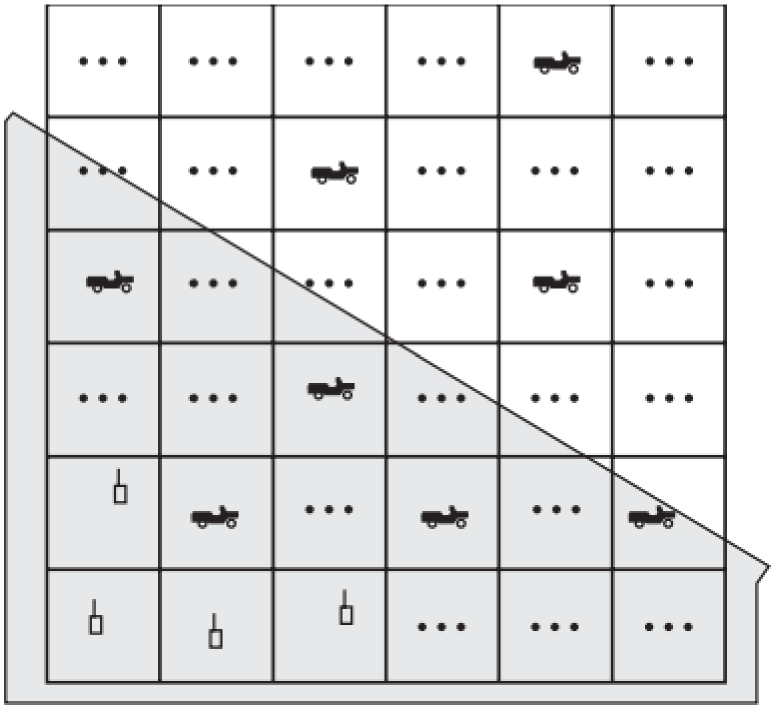


NONORTHOGONAL RANGE SEARCHING

- Forcing all attributes to be one-dimensional is sometimes overly constraining.
- Viewing sensor location as two independent attributes(x, y) coordinates.
- we have n sensors fairly evenly spread over a two-dimensional area.
- Suppose further that these sensors are able to detect and count the number of targets in a small region around themselves.
- Either in isolation or in a lightweight collaboration with their neighbours.

- To count the total number of targets present in certain type of simple nonorthogobal geometric areas.
- For the example we assume that these areas must be halfspaces.
- This target count can be obtained easily.
- We put a $\sqrt{n} * \sqrt{n}$ grid over the sensor network so that in each of its cells there is at least one sensor whose sensing region contains the cell.
- Now activate the sensor and count the number of targets within each of the grid cell, and we propagate these counts along rows of the grid

- Consider now a halfspace query.
- Except for at most $2\sqrt{n}$ grid cells intersected by the halfspace edge.
- We can easily compute the total number of targets in the halfspace.
- We are able to answer the halfspace target counting query in $(2\sqrt{n})$ time.



THANK YOU

THE CONTENTS IN THIS E-MATERIAL IS TAKEN FROM THE TEXTBOOKS AND REFERENCE BOOKS GIVEN IN THE SYLLABUS