Data-Centric Storage

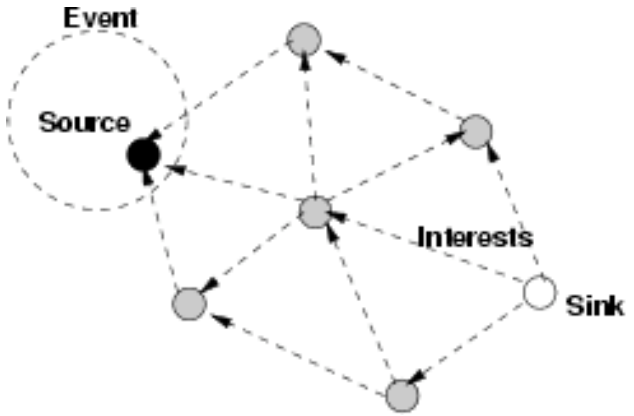
Ramesh Govindan
ramesh@usc.edu

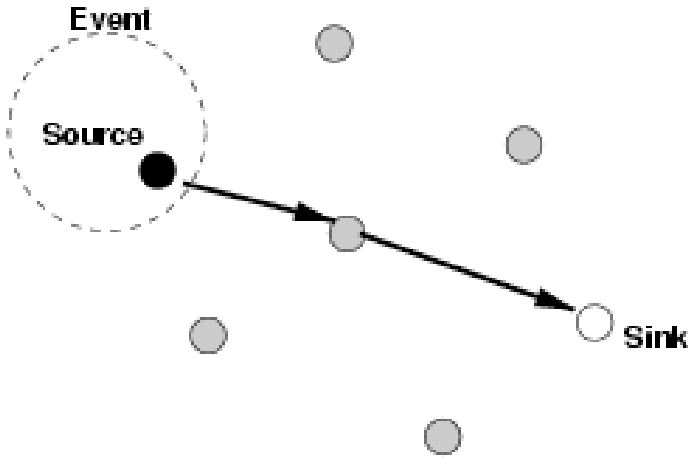
Introduction

- The Challenge
  - Sensor network data will be stored within the network for energy-efficiency
  - How can we efficiently (and in a scalable fashion) access this information?
- What do we mean by information?
  - Events: data representing physical phenomena (e.g. bird sightings)
  - Features: statistical patterns in the sensor field (e.g. thermoclines)
- What kinds of access
  - Users "outside" the network
  - Nodes "within" the network
    - *e.g.* triggering
- What kinds of scale?
  - 100s of nodes

Prior Work

- *Flood-then-respond* systems
  - Directed Diffusion
  - TAG/TinyDB
  - Two-tier data dissemination
- Good for
  - Long-lived or continuous queries
  - Issued by users outside the network

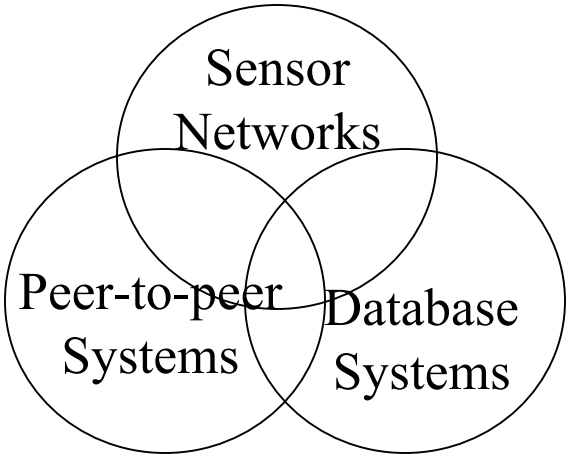




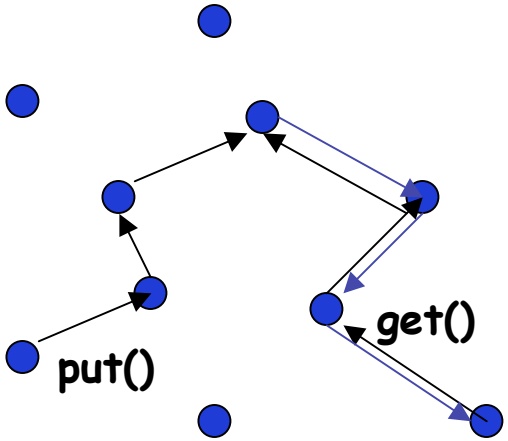
An Alternative Class of Queries

- One-shot queries
- Issued from *within the network*
  - Correlating events across nodes
  - Testing for conditions within the network
- Useful for *triggering* actions within the network
  - Can be fairly frequent
- *Flood-then-respond systems* don't scale for this class of queries

Data-Centric Storage

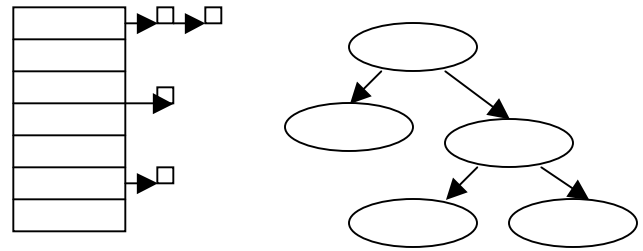


Data-Centric Storage



- Simple primitives
- Efficient information retrieval in sensor networks
- Challenges
  - Geographic routing
  - Robustness to failure

Distributed Data Structures



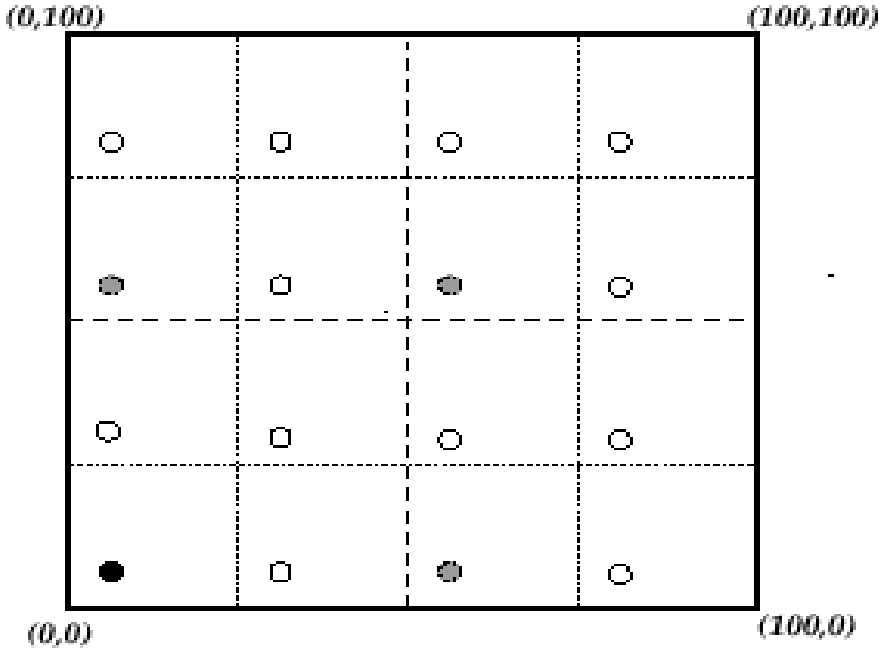
Efficient sensornet querying and triggering

An Instance of Data-Centric Storage

- Geographic Hash Tables (GHTs)
  - *Hash* the name of the data to a geographic location
  - Store data at the node closest to that locations
    - Use a geographic routing protocol (e.g., GPSR) for this
  - Can retrieve data the same way
- Extensions
  - Robustness: Perimeter Refresh
  - Scaling: Structured Replication

Joint work with Deborah Estrin, Brad Karp, Sylvia Ratnasamy, Scott Shenker, L. Yin and F. Yu

Structured Replication



- *root point: (3,3)*
- *level 1 mirror points: (53,3) (3,53) (53,53)*
- *level 2 mirror points: (28,3) (3,28) (28,28) (78,3) (53,28) (78,28) (3,78) (28,53) (28,78) (78,53) (53,78)(78,78)*

Building Blocks for Data-Centric Storage

- Key observation
  - *Hashing* and *geographic tesselation* can be used to build a variety of *distributed data structures* for efficient access to information in sensor networks
- Examples
  - Distributed Index for Features in Sensor Networks (DIFS): Greenstein *et al.*
  - DIMENSIONS: Ganesan *et al.*
- Can hope to leverage a common software base to build a variety of data structures

Distributed Index for Multi-dimensional data (DIM)

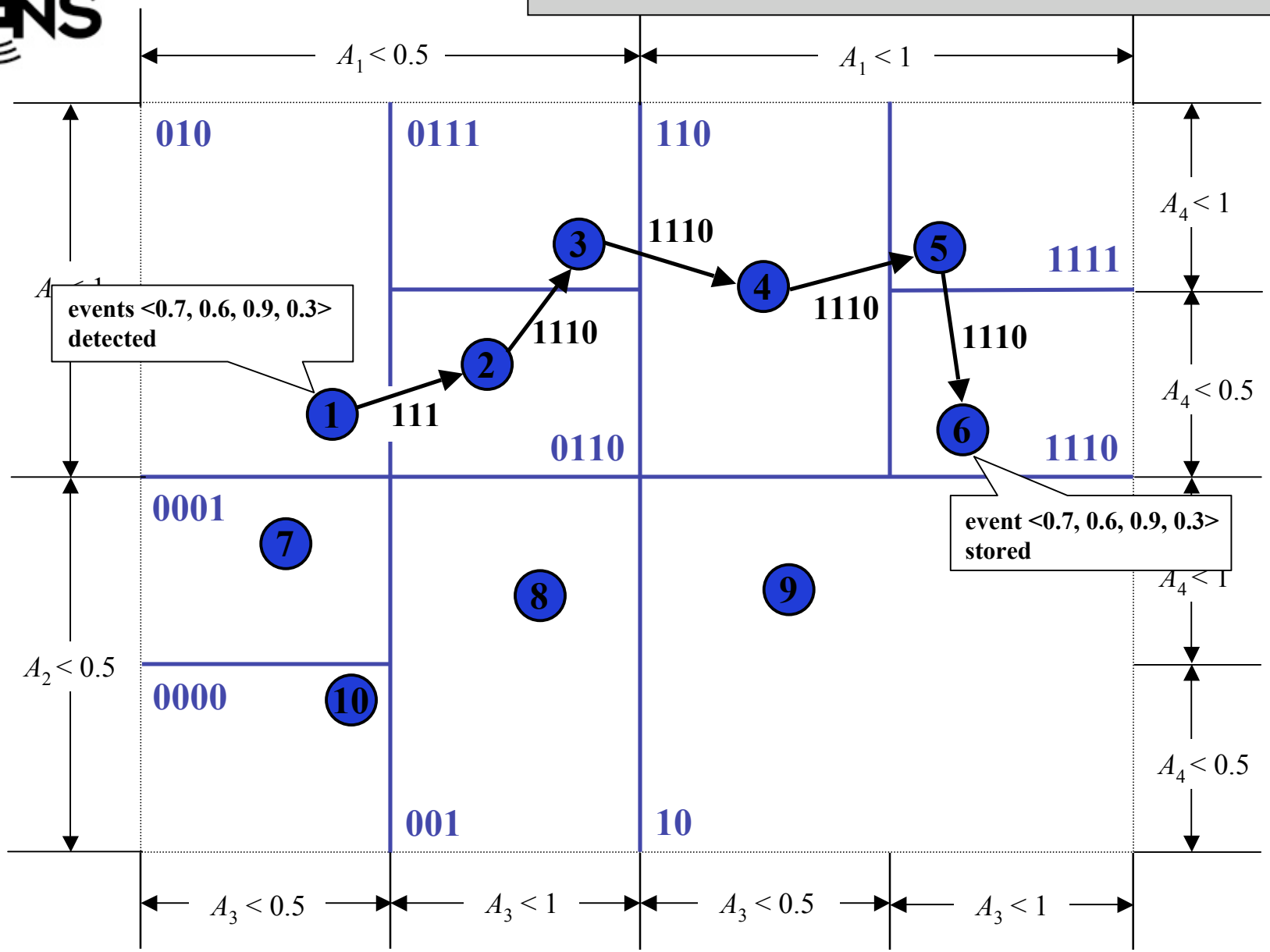
- Provides support for multi-dimensional range queries in sensor networks.
  - e.g. List all events whose temperature lies between 70 and 80 and whose light levels are between 10 and 15.
  - Can be used for searching and correlating events of interests with multiple attributes.
- Classical approaches in traditional databases --- Indices
  - For energy-reasons, we focus on a distributed index

Joint work with Xin Li, Young Kim and Wei Hong

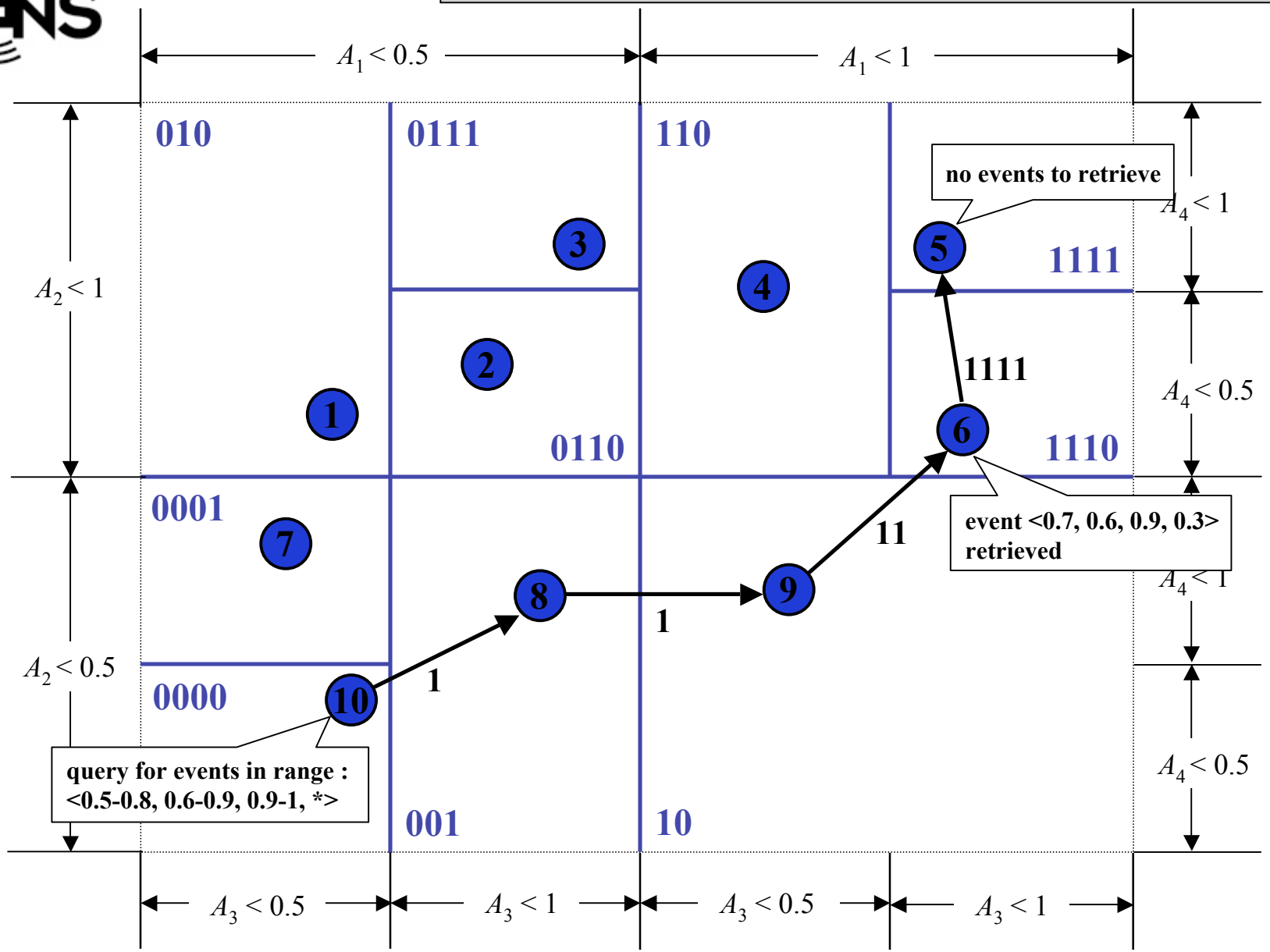
DIM Components

- A virtual index tree that covers the entire network
  - A binary tree jointly built by all nodes in a recursive and distributed way
  - DIM builds the tree such that each node knows only a part of the tree and the rest of the tree is not crucial for its work.
- An allocation scheme that assigns each node to a bin.
  - DIM calls each bin a *zone*.
- A *hash* function that maps data and queries to bins.
  - DIM maps each data tuple to a single zone.
  - DIM maps each query to one or multiple zones depending on the ranges of the query.

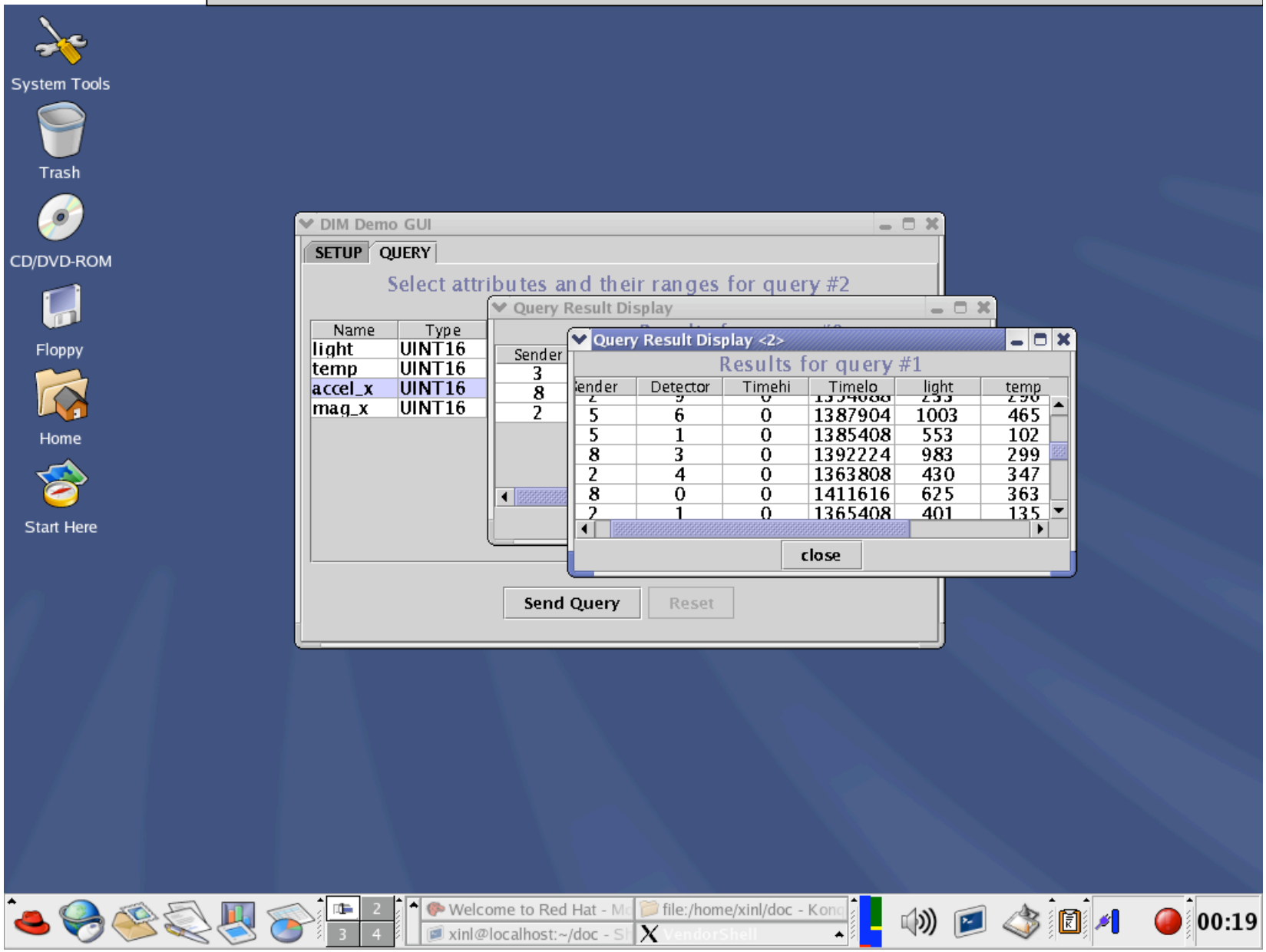
Data Insertion in DIMs



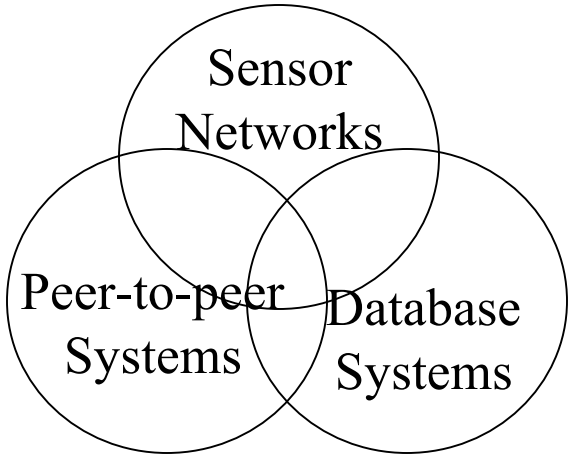
Querying in DIM



See the Demo!


System Tools
Trash
CD/DVD-ROM
Floppy
Home
Start Here
DIM Demo GUI
SETUP
QUERY
Select attributes and their ranges for query #2
Name
Type
light
UINT16
temp
UINT16
accel_x
UINT16
mag_x
UINT16
Sender
3
8
2
Query Result Display
Query Result Display <2>
Results for query #1
Sender
Detector
Timehi
Timelo
light
temp
5
6
0
1387904
1003
465
5
1
0
1385408
553
102
8
3
0
1392224
983
299
2
4
0
1363808
430
347
8
0
0
1411616
625
363
2
1
0
1365408
401
135
close
Send Query
Reset
Welcome to Red Hat - M
file:/home/xinl/doc - Kong
xinl@localhost:~/doc - Sh
VendorShell
00:19

Status and Futures



- Many data-centric storage components implemented
  - GPSR
  - DIM
- Future directions
  - Incorporating non-geographic routing protocols
  - Query optimization in distributed data structures
  - Supporting procedural and logic programming languages using data-centric storage