Outline

Contents

1 Research Statement

Research Goals

- Identify *common sensing* and *networking characteristics* among competing application-specific protocol solutions.
- Quantify desired *application-oriented context* and *contingencies* in terms of QoS and service requirements.
- Integrate *aggregation*, *hierarchy*, and *approximation* to promote scalability and portability.
- Design a *flexible* and *generic communication abstraction* that allow the same set of protocols to be used across applications and platforms with little or no modification.

2 Wireless Ad hoc Sensor Networks (WSNs)

2.1 Applications: Civil, Medical, Industrial, Military

Application Objectives

- Reliable *monitoring* of a variety of environments.
- Enables information *gathering* and *processing*.
- Integrates physical *sensing* and *controlling* capability with a *communication* oriented infrastructure, say Internet.
- Useful for various *applications* including civil and military.

Sensor Network Applications

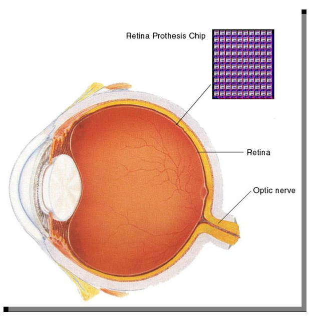
Smart Homes and Offices

Typical Uses

- Sensors *controlling appliances* and electrical devices in the house.
- Better *lighting* and *heating* in office buildings.
- The *Pentagon* building has used sensors extensively.

Sensor Network Applications

Biomedical/Medical



Typical Uses

- *Health Monitors*: Glucose, Heart rate, Cancer detection.
- *Chronic Diseases*: Artificial retina, Cochlear implants.
- *Hospital Sensors*: Monitor vital signs, Record anomalies.

Sensor Network Applications

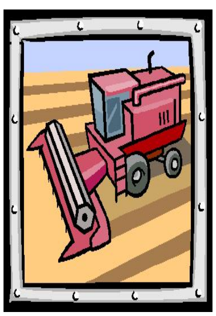
Military/Tactical

Typical Uses

- Remote deployment of sensors for *tactical monitoring* of enemy troop movements.
- Provides *situational awareness*.
- Supports *troop collaboration*, status, and coordination.

Sensor Network Applications

Commercial



Typical Uses

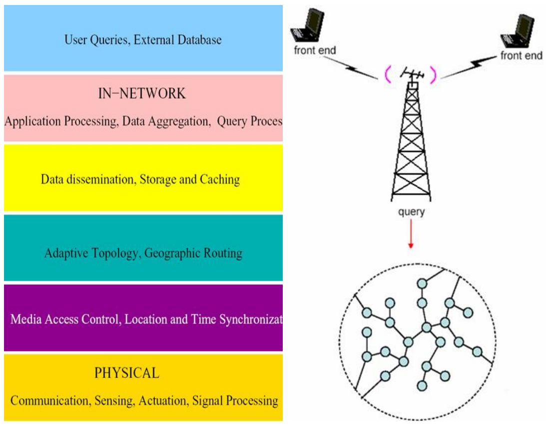
- *Agricultural* Crop Conditions.
- *Inventory* and in-Process Parts Tracking.
- Automated Problem *Reporting*.
- *RFID*: Theft Deterrent and Customer Tracing
- Plant Equipment *Maintenance* Monitoring

2.2 Network Model

Ad hoc Communication Paradigm

Proposed Layers for WSN

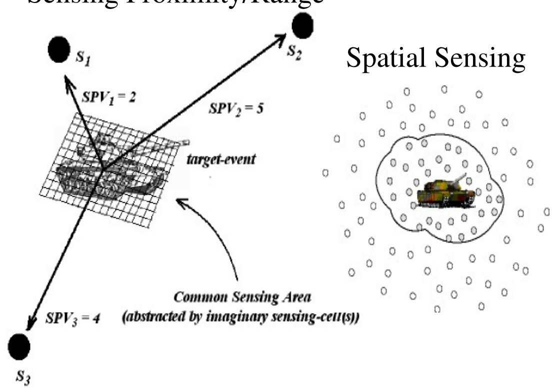
Typical WSN Infrastructure



Collaborative Sensing Paradigm

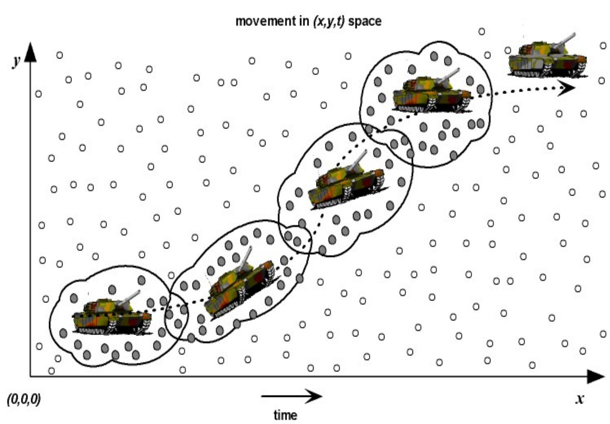
Sensing Proximity/Range

Spatial Sensing



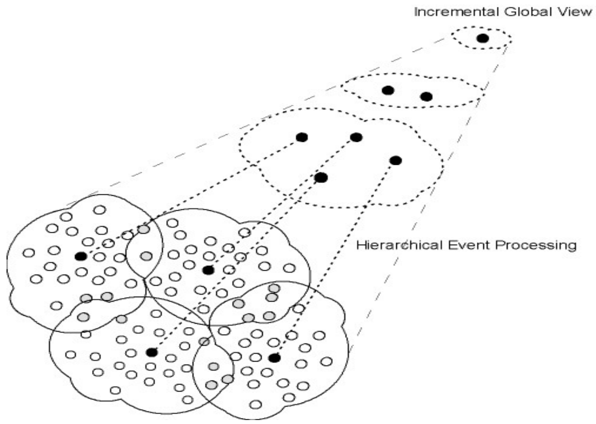
Collaborative Sensing Paradigm

Spatio-Temporal Sensing



Collaborative Sensing Paradigm

Comprehensive Sensing: Incremental and Hierarchical



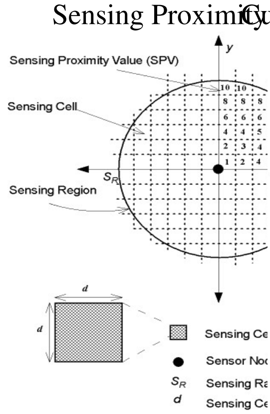
Collaborative Sensing Paradigm

Proposed Sensing Attributes (Metrics)

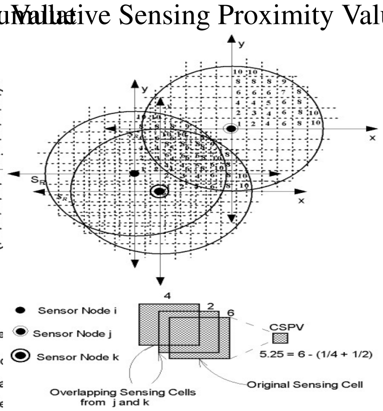
- Sensing Proximity Value (*SPV*).
- Cumulative Sensing Proximity Value (*CSPV*).
- Cumulative Sensing Degree (*CSD*).

Collaborative Sensing Paradigm

Sensing Proximity Value

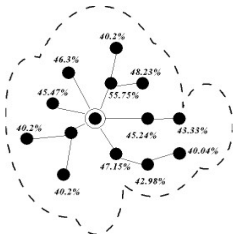


Cumulative Sensing Proximity Value



Collaborative Sensing Paradigm

Group Sensing



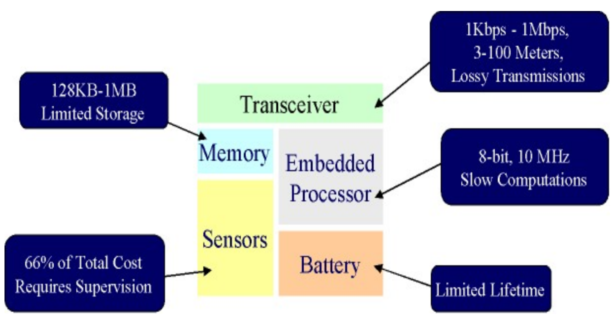
Cumulative Sensing Degree (*sQoS*)

- Degree of *fault tolerant sensing* by neighbors.
- Average of *CSPVs* of all sensing cells.
- CSD: *Percentage* coverage of ideal and solitary coverages.

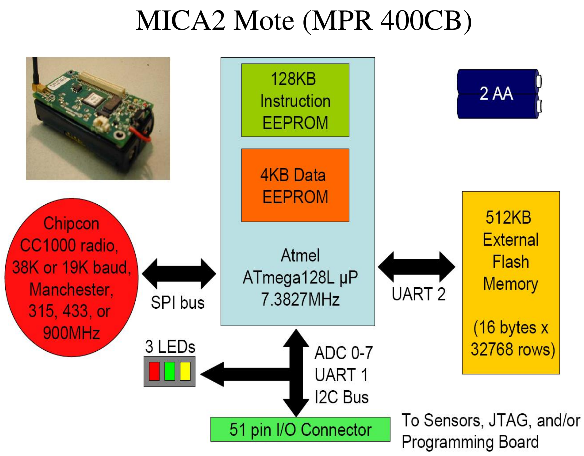
2.3 System Overview: Hardware/Software

Hardware: System on Chip

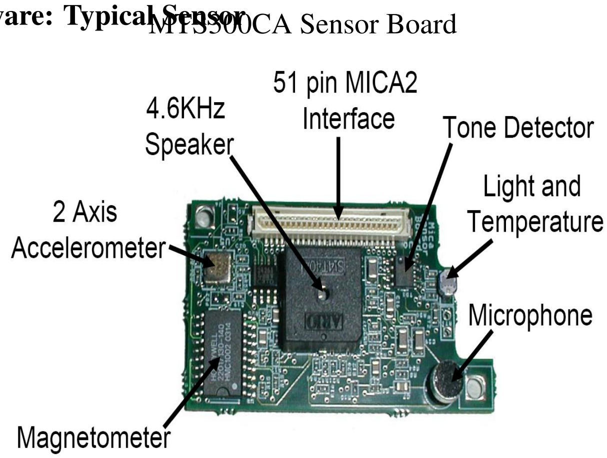
Node Hardware Limitations



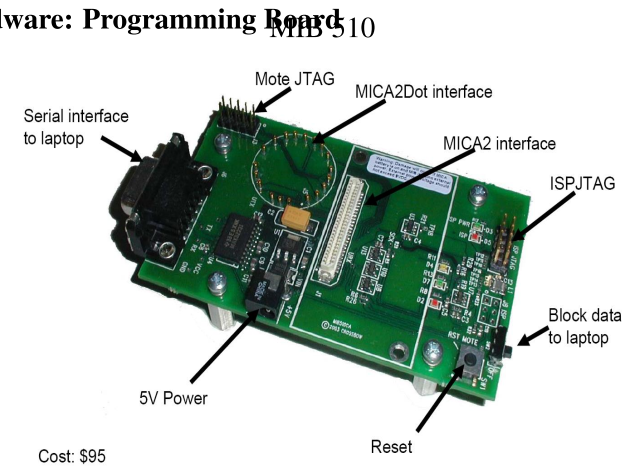
Hardware: Typical Mote



Hardware: Typical Sensor

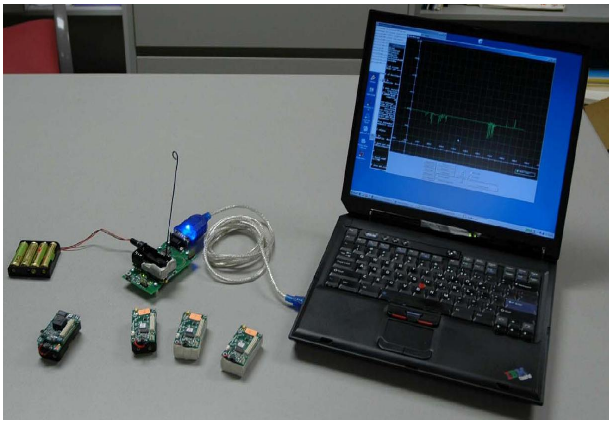


Hardware: Programming Board



Typical Hardware Setup

Ad hoc Network With Base Station



TinyOS: Micro Operating System

- *Single threaded*.
- An *open source* development environment.
- *Component-oriented* programming language (NesC).
- Main Ideology: *Sleep* as *often* as possible to save power.
- *High Concurrency*, interrupt driven (no polling).
- *Static memory allocation*. No heap (malloc) and no function pointers.

2.4 Constraints and Challenges

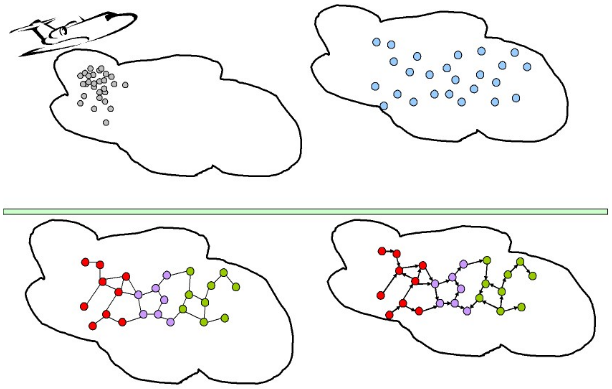
Characteristics

- Large scale deployment.
- High unpredictability.
- Redundancy.
- Constrained resources.
- Real time constraints.
- In-network processing.
- Data-centric processing.
- Security.

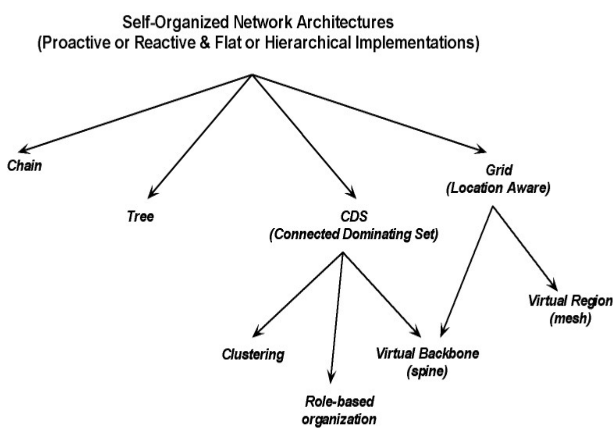
3 Research Contributions

3.1 Role-based Hierarchical Self Organization Protocol

Visualizing Sensor Self Organization

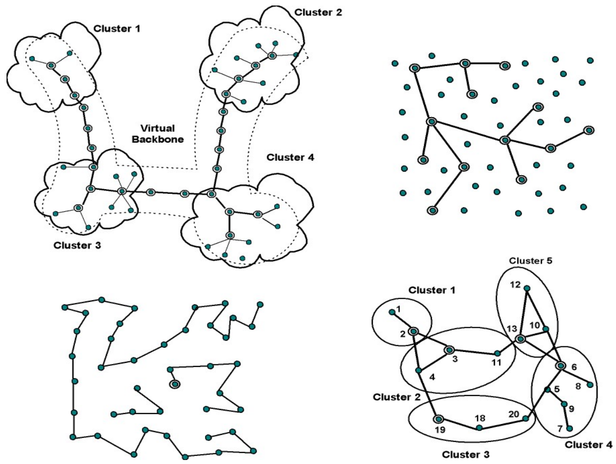


Related Work: Proposed Architectures



Clockwise: Spine, Tree, Chain, and Cluster (CDS)

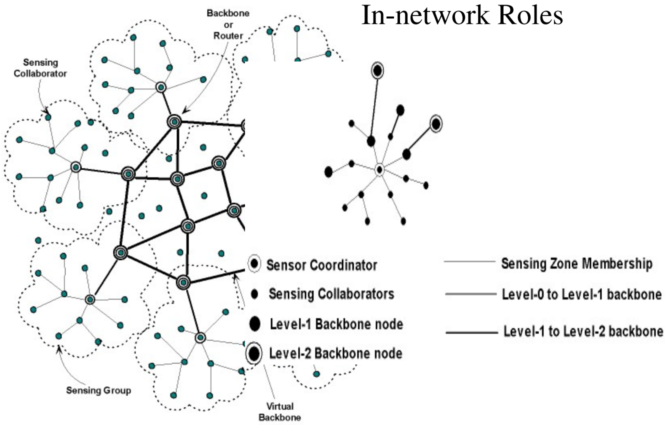
Ad hoc Network Architectures



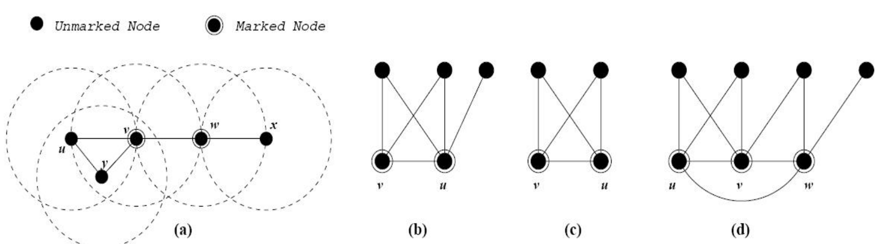
High Level RBHSO Architecture

Hierarchical Role Organization

In-network Roles



Recursive Role Domination



Initial marking process

3 examples of dominating set reduction

RBHSO: Algorithm Details

Construct CDS Hierarchy

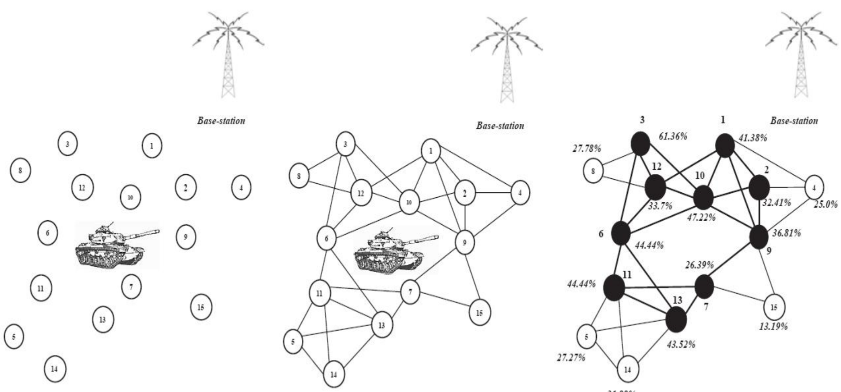
- Neighbor Discovery.
- Initial Marking Process (Level 0 marked nodes).
- Dominating Set Reduction (Level 1+ marked nodes).

Metrics used in order

- Energy Level (EL),
- CSD (or Cumulative Sensing Degree),
- Connectivity based metric or node degree (ND), and
- ID of the sensor node.

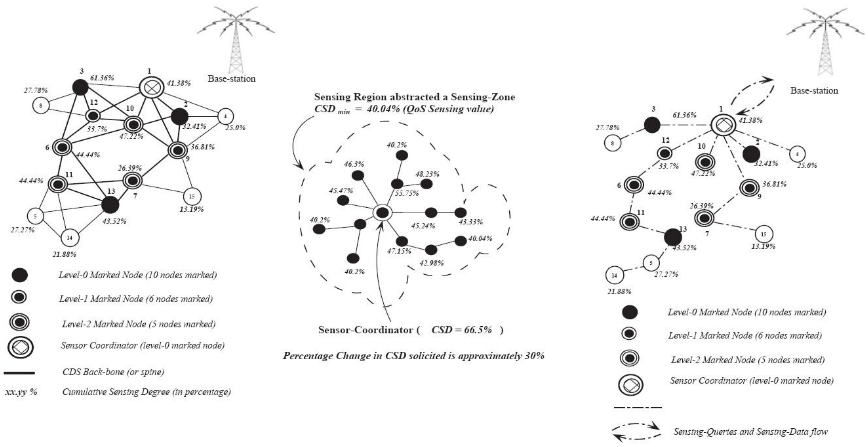
- Sensor Coordinator Selection.
- Sensing Zone Formation Algorithm.

Example Self Organization Scenario

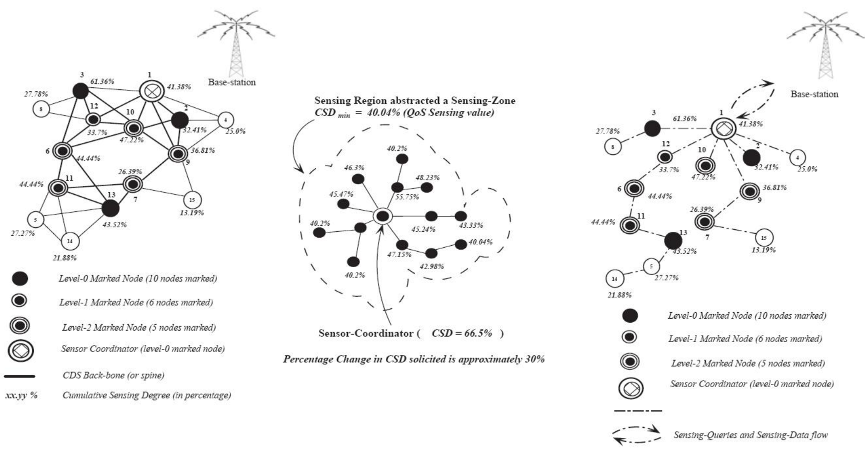


Example 15 sensor nodes deployment $NEIGHBOR_{DISCOVERY}$ stage marking stage

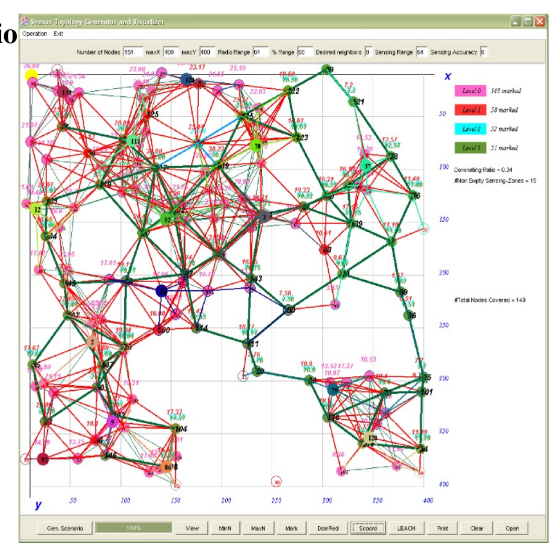
Example Self Organization Scenario



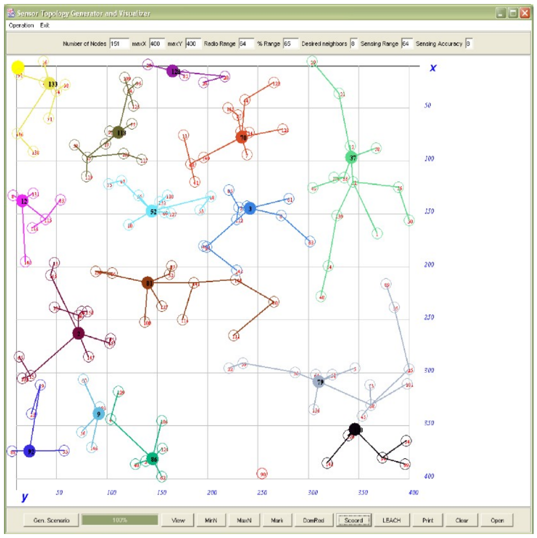
Example Self Organization Scenario



Simulatio



Simulation: Sensing Zones



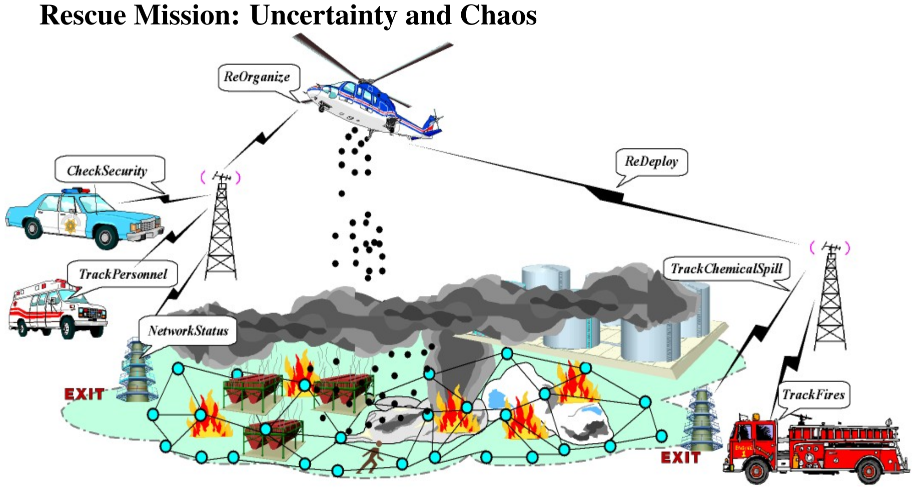
Conclusions

Abstraction: Hierarchy, Approximation, and Aggregation

- Experimented with *roles*.
- Used several networking and sensing metrics as *rules*.
- Recursive *dominating set reduction* results in *role-hierarchies*.
- Mapped application-specified *sQoS* such as CSD.

3.2 Unified Role Abstraction Framework

Rescue Mission: Uncertainty and Chaos



Rescue Mission: Uncertainty and Chaos

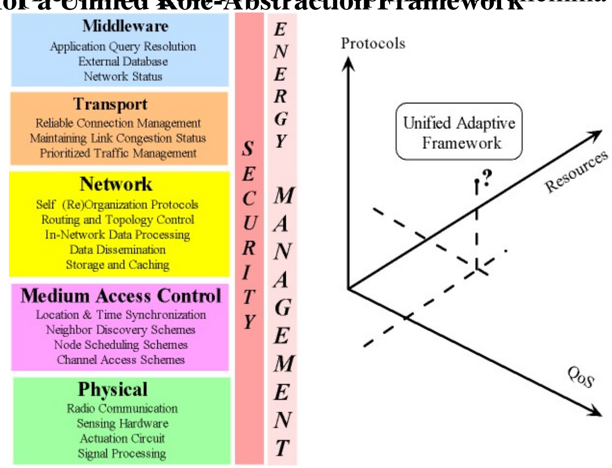
Application, Protocols, and Layering Issues

- Protocol *optimization*(s) are at *odds* with each other.

- Warrants *tradeoff decisions* among competing goals.
- Appropriate *real-time response* to application demands and environmental situations.
- *Context-awareness* requires k-hop sharing of *cross-layer* information.
- Efficient *coordination* for fair *resource-allocation* becomes necessary.

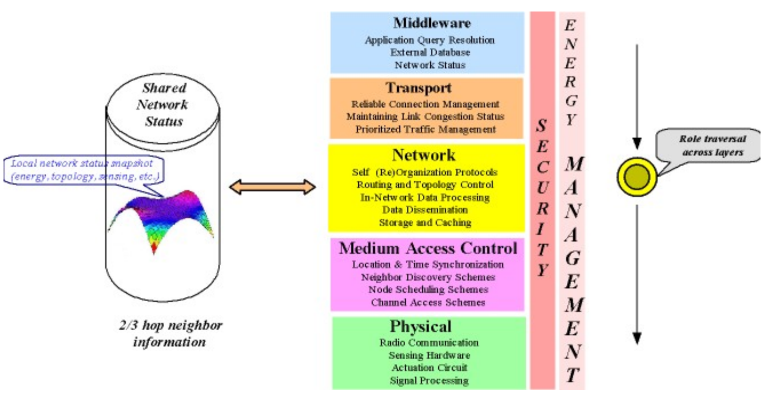
Case for a Unified Role-Abstraction Framework

{Resource, QoS, Protocols} optimization dilemma



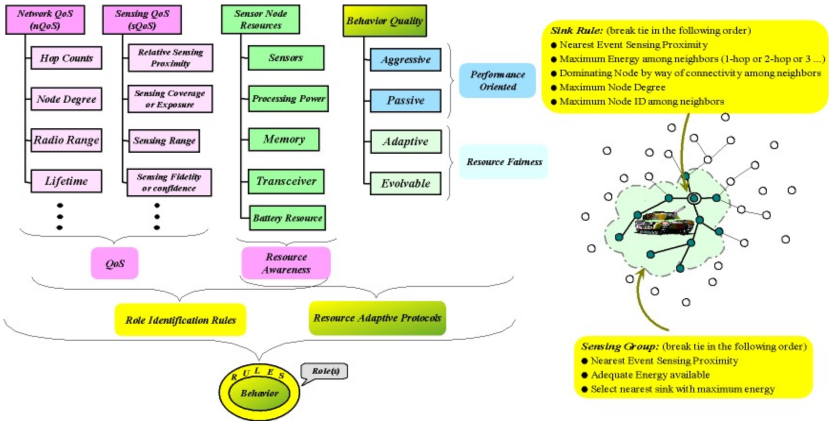
Case for a Unified Role-Abstraction Framework

Role Context-Awareness



Generic Role and Rule based Abstraction

Abstraction: QoS, Resources, Tasks, Roles, and Rules



Elementary Sensor Network Tasks

Fundamental Actions Executed by a Node

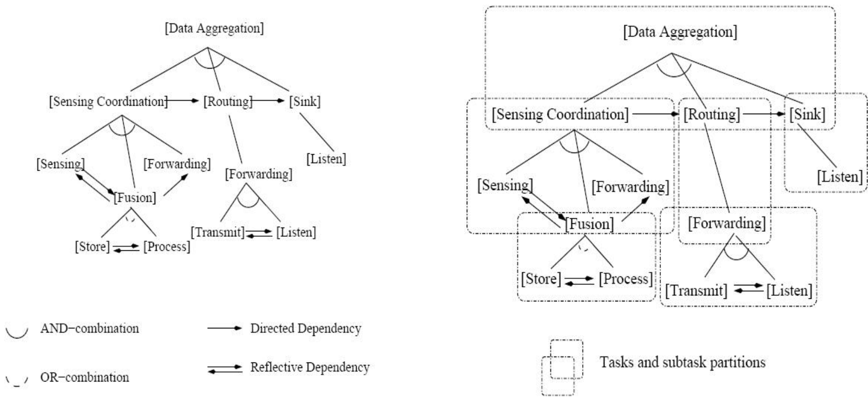
- *ON*: Node turned *ON* and is idle
- *OFF*: Node completely turned *OFF* to save energy
- *Sense*: Sensing task, *S*
- *Process*: Processing role, *P*
- *Store*: Node storing data in its memory, *M*
- *Transmit*: Transmitter, *T*
- *Listen*: Role for listening to packets, *L*

Complex Role Formulations

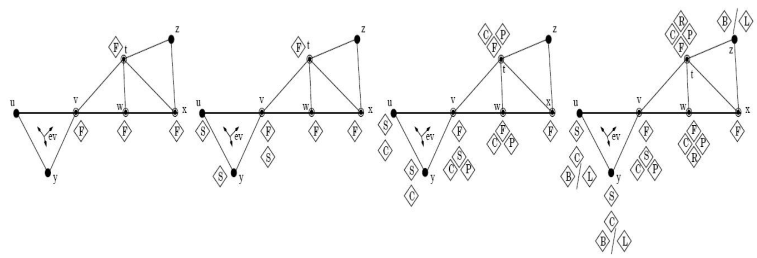
Role Compositions

- *Forwarder*: $F \Leftrightarrow T \bigwedge L$
- *Router*: Series of forwarders, $R \Leftrightarrow \bigvee_{i=1}^{n} F^i$
- *Aggregator*: Storage and processing tasks, $A \Leftrightarrow M \bigwedge P$
- *Sensing Collaborator*: Transmits sensor readings and listens for coordination, S_m i.e. $S_m \Leftrightarrow (S \bigwedge P) \bigwedge (T \to L)$.
- *Sensor Coordinator*: Coordinates sensing zone and forwards data to sink, $S_h \Leftrightarrow (\bigvee_{i=1}^{n} S_m^i) \bigvee_{j=1}^{m} F^j$
- *Sensing Zone*: $S_r \Leftrightarrow ((\bigvee_{i=1}^{n} S_m^i) \bigwedge S_h) \bigvee_{j=1}^{m} F^j$.
- *Target Tracking*: Track manager, sector manager, and sector.

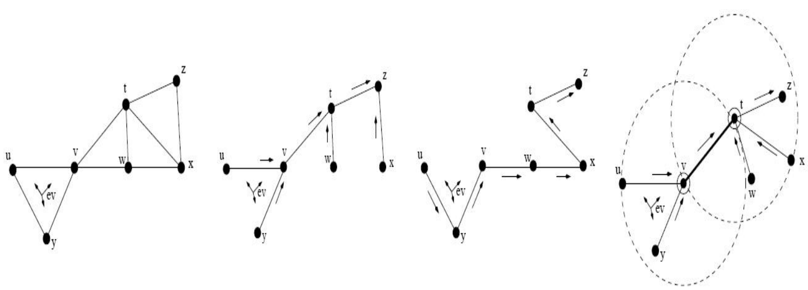
Hierarchical Task Decomposition and Grouping



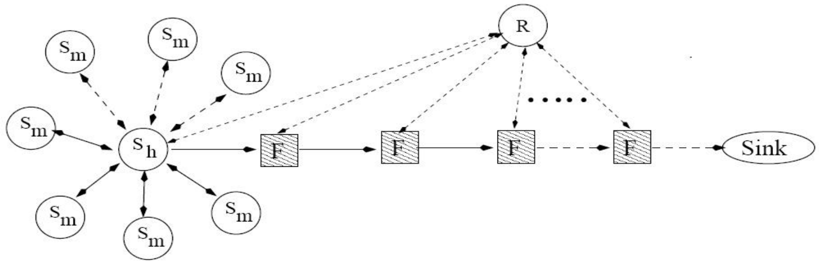
Redundant Role Assignment (RA) Technique



Role Assignment Leads to Topology Differentiation

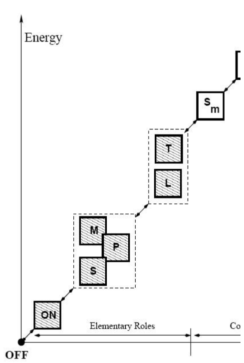


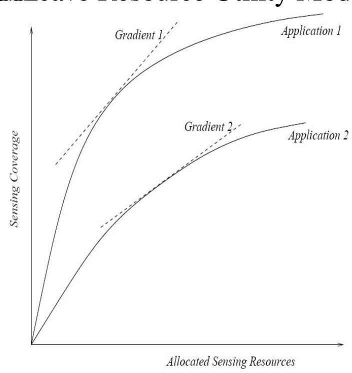
Role Coordination Graph



Domain Specific Models

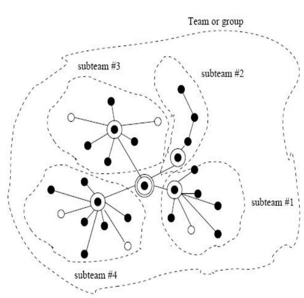
Role Energy Model Concave Resource Utility Model





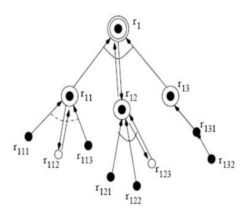
Role Failures

Proactive and Reactive Monitoring



Hierarchical Role-network organization

Level 0 role (per node)
Redundant Role
Level 1 role (per subteam or subgroup)
Level 2 role (per team or a group)



Dominating monitors for roles at lower level

AND–combination
OR–combination
Directed Dependency
Reflective Dependency

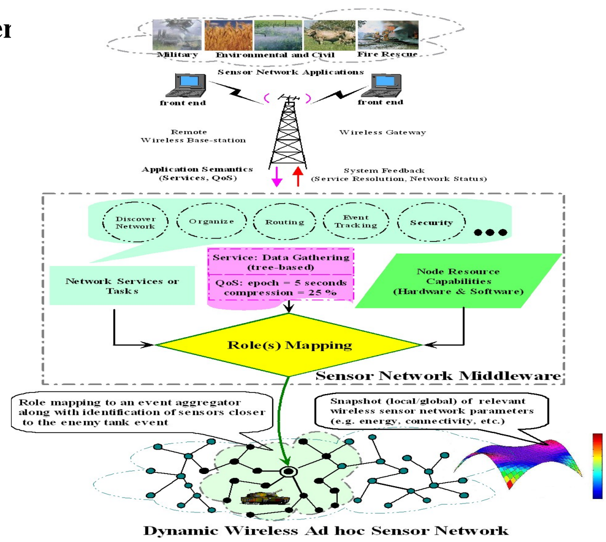
Other Role Properties

Execution Scheduling, State Machine, and Load Balancing

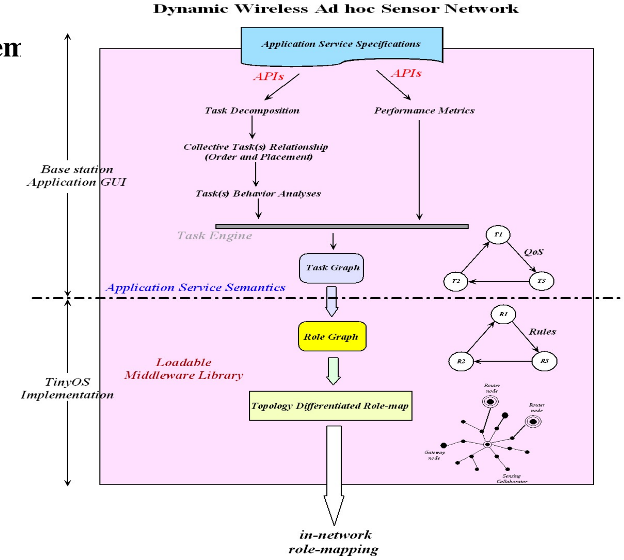
- *Execution Scheduling*: TDMA time slots coordinated by dominator for role control and execution.
- *Role State Machine*: Predefined for roles for message arrival, sensing events, and context changes.
- *Load Balancing*: Pairwise neighborhood role-exchange, role-mergers, and role-redirection protocols.

3.3 Role-based Middleware

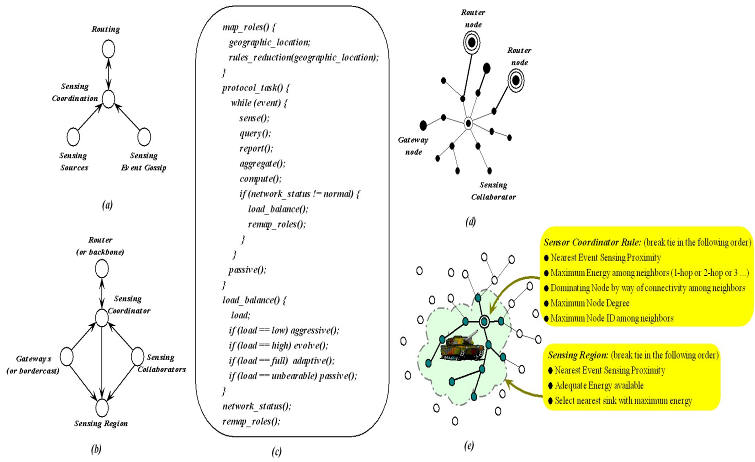
Higher



System



Generic Code for specifying Executable Roles



Generic Specification Language for Application

- *Simple* requirement expressed as points in *QoS* space.
- *Weighted* sum of *points* in QoS space.
- *Utility* based QoS specification.
- *Weighted utility* based QoS specification.

Multi-Service Minimum Energy Role Assignment (*MSMERA*)

NP-Complete Problem

- Minimizing the *number of roles* for a service.
- Minimizing *network flows among roles* necessary to reduce communication overhead.
- *Shortest path* communication among roles.
- Solving the above *distributedly* with *partial* and *local* network information is NP-Complete.

Minimum Total Energy RA (*MTERA*)

Depends on following factors

- Number of messages exchanged during every RA round.
- Number of such RA rounds per service mapping.
- Number of roles per service mapping.
- Number of nodes/role.
- State dependency among distributed roles.
- Hop-count or path-distance among roles.

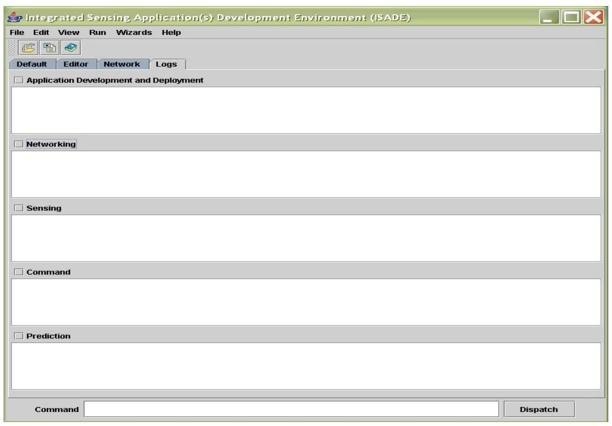
Multi-Service Minimum Energy Role Assignment (*MSMERA*) Techniques

Our Proposals

- *Redundant* role *assignment* technique (naive).
- *Greedy recursive* domination set based reduction technique.
- *Utility* based role-assignment technique by way of ranking.
- *Hybrid*: Cooperative redundant coalitional role-assignment with iterative pruning.

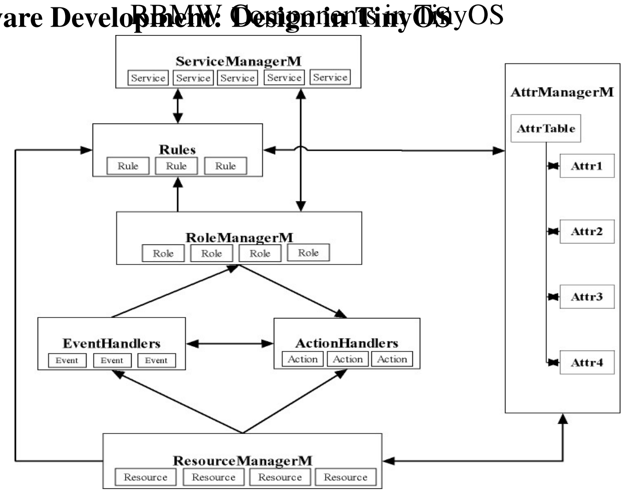
Software Development: User Front End GUI

Integrated Sensing Applications Development Environment



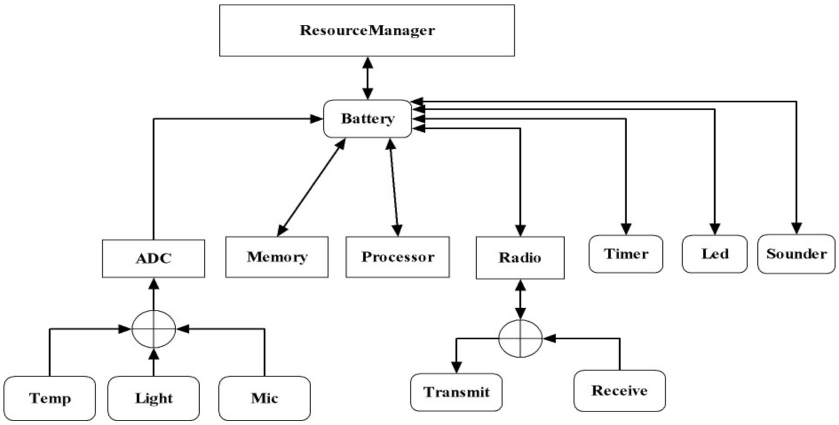
Software Development: Design in TinyOS

RBMW Components in TinyOS



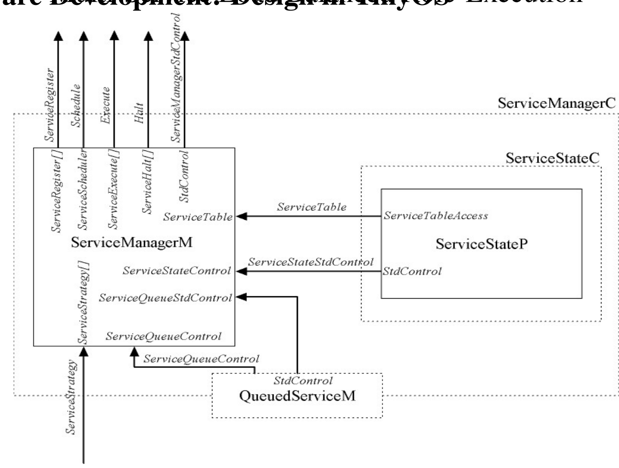
Software Development: Design in TinyOS

Resource Usage Accounting Model



Software Development: Design in TinyOS

Multi-Service Load-Balanced Role-Execution



Performance Analysis: Simulation and Implementation

Performance Parameters

- *RA algorithm efficiency*: Load per node, number of nodes per role, and RA frequency.
- *QoS mapping efficiency*: Accuracy of mapping and its degradation with time.
- Effects of *complex role formulations* or task/subtask partitioing and/or grouping.

4 Summary

Conclusions

- Proposed a *role-based service paradigm* for sensor networks.
- *Generalized and unified* role-abstraction mechanism across layers, services, and applications.

- Supports *rules* formulation in terms of *cross-layer network attributes* for selecting roles.
- Supports *service specification* in terms of single or weighted set of QoS metrics and utilities.
- Developed *emphirical models* to quantify service composition quality in terms of energy and resource allocation utility.

Future Work

- Role formulation for other services, e.g. *security*.
- Export the programming language for roles and application specification over a generic *Virtual Machine*.
- How about interaction among *generic role-societies* that employ any role-composition.
- *Economical Issues*: Provider of a service should benefit monetarily.
- *Utility* based decision making, *Game Theory*, and *Mechanism Design* can provide better RA solutions.
- *Standardization efforts* needed in the WSN arena for universal adoption of roles.

Funding: Research and Graduate Studies

- National Science Foundation (*NSF*) under grant ANI-0086020.
- Graduate Dissertation Fellowship.
- Department of Computer Science.

Collaborators

- Dr. Loren Schwiebert (Advisor), Wayne State University.
- Dr. Sandeep K. S. Gupta (Co-advisor), Arizona State University.
- Discussion with colleagues: Dr. Daniel Grosu and Fernando Martincic.