An Adaptive Sensor Network Architecture for Multi-scale Communication

Santashil PalChaudhuri
Computer Science Department
Rice University
Sensor Network Applications

- Sensor nodes can sense, process, and communicate

- A few applications
  - Distributed surveillance
  - Preventive maintenance
  - Environment monitoring
  - Industrial automation
  - Smart buildings

Source: www.intel.com
Sensor Network Characteristics

- Adaptability to applications
  - Limited set of application requirements
- Energy optimization is critical
  - Limited battery capacity
- Data fusion capability
  - In-network processing (e.g., averaging)
- Data-centric routing
  - The sensed data, not the individual sensor, is important
- Resource constraints
  - Physical form factor
- Large scale
  - Traditional stateful routing not possible
In wireless ad hoc sensor networks, there is simultaneously a need and an opportunity for optimizing the protocol behavior to match sensor-based applications.

In my thesis, I create adaptive protocols for sensor networking, taking advantage of sensor network and application characteristics and requirements.
Multi-scale Algorithms

- Data is sensed, processed, and communicated at difference scales
  - Coarse monitoring at regular time intervals, and finer scale less frequently – *reduces required resource*
  - Anomalous data initiates drilling down to the problem region for finer resolution – *on-demand resolution*
Thesis Contributions

- An adaptive cross-layer sensor network architecture for multi-scale applications

- Design and evaluation of component protocols
  1. Data Service
  2. Transmission Scheduling
  3. Clock Synchronization
Transmission Scheduling
Network Hierarchy Overview
Hierarchical Communication

Intra-Cell (L1)

Inter-Cell (L2, L3, ...)

Super-Cell I

Cell I

Cell II

Cell III

Cell IV

A

B

C

D

1 2 3 4 5 6
Scheduling Goals

- Design a transmission schedule to provide energy efficiency and minimal latency for multi-scale data service
- Cross-layering for relevant information
  - Routing topology gives data flow
  - Network Programming Interface (NPI) implemented by the data service provides communication characteristics
  - Fusion or aggregation function characteristics
Related Work: Scheduling

- **Contention-based**: 802.11, PAMAS, SMAC, TMAC, BMAC
  - ✓ Flexible with increased traffic fluctuations and node failures or additions
  - ✹ Overhead from idle listening, packet collisions, control packet overhead
- **Schedule-base**: TRAMA, Sichitiu
  - ✓ Collision-free and no idle listening
  - ✹ Reduced flexibility to handle variable traffic and changing neighborhood
- **Hybrid approach**: 802.11 PCF, 802.15
  - Combination of both
Time-Frame Design

Time-frame divided into two parts

- **Synchronized Contention Free Period (CFP)**
  - Periodic deterministic application traffic

- **Event-driven Contention Access Period (CAP)**
  - Event-driven traffic—e.g., intrusion or fire detection

I design a protocol for CFP, and leverage existing random access techniques for CAP
Intra-cell Structure Schematic

- Multi-hop routing structure formed by *beacons*
Design Principle - I

- Conflict-free slot allocation in 2-hop neighborhood
  - Knowledge of 2-hop neighbors and their allocated slots
Design Principle - II

- Latency reduction
  - Parents allocated slots after children
  - Depth-first search gives a post-order traversal having the same property
Design Principle - III

- Reduced duration of awake state
  - Allocate new slot close to slots of siblings with already allocated slots
Design Principle - IV

- Nodes know about fusion function
  - Determines no. of slots to allocate for forwarding
- Fusion functions assumed to be simple
  - Does not depend on input, or has a bound
Strawman Centralized Algorithm

- DFS finish times gives post-order traversal
- Conflict-free

- Inefficient in terms of latency
- Assumes complete graph knowledge
- Inter-cell interference not accounted for
Distributed Protocol

- Token sent in DFS order
- Parent sends token to each child, child sends back
- Tokens carry information about allocated slots
- Nodes know 2-hop neighborhood

Base algorithm: choose children in random order
Mapping to Graph Coloring

- $G =$ network graph; form $G^2$ from $G$ such that
  - $V =$ vertex set from $G$; $E =$ 2-hop connected in $G$

- Minimal graph coloring of $G^2$ is similar to scheduling problem (this is NP-hard)

- Competitive Ratio of Base algorithm
  - Upper Bound: 6 (assuming unit-disc graph)
  - Lower Bound: 1.2 (with worst-case graph)
Order of Child Traversal Matters

- Evaluated various heuristics for choosing child traversal order
  - Larger sub-tree, larger # of children, larger 1-hop neighborhood
- Best was *Degree algorithm*: Sort and visit children according to larger 1-hop neighborhood (degree of the node)
Bounds on Number of Slots

- $\Delta = \text{degree}$
  - $\Delta^2 = \text{2-hop degree}$
  - $h = \text{height of tree}$

- **Theorem 1**: Upper bound of number of slots used is $\Delta^2 + h$

- **Theorem 2**: Lower bound of number of slots used is $\Delta + h$
Evaluation

- Latency and energy efficiency
- No existing solution has utilized these cross-layer parameters before for optimization
- Therefore, for fair comparison
  - Implemented optimal solution for slot allocation
  - Used simulated annealing for reducing exponential search space
Simulation Parameters

- Nodes: 10–1000
- Average degree: 6, 8, and 10
- Random uniform distribution of the nodes
- Sink is the node closest to center of the field
- Average of 50 runs per point
Levels of Hierarchy vs. No. of nodes
Average # of Slots Awake vs. # of Nodes
Latency to Gather from All Nodes vs. # of Nodes

![Graph showing latency to gather from all nodes vs. number of nodes. The x-axis represents the number of nodes, ranging from 0 to 1000, and the y-axis represents the total slots needed, ranging from 5 to 45. The graph includes lines for different average degrees: Average Degree = 6 (red), Average Degree = 8 (green), and Average Degree = 10 (blue).]
Demonstration of Optimality Proof

![Graph showing total slots needed vs. number of nodes]
Comparison of Total Latency

![Graph showing comparison of total latency for different algorithms.]
Normalized Optimality

![Graph showing normalized optimality over the number of nodes.]
Synchronization
Related Work: Synchronization

- Traditional clock synchronization
  - NTP
- Wireless clock synchronization
  - Romer, Huang
- Receiver-receiver synchronization
  - CesiumSpray, Reference Broadcast Synchronization (RBS)
- Probabilistic synchronization
  - Christian, Arvind
Need for Adaptation

- Applications have divergent needs for maximum synchronization error and confidence probability
  - Provide synchronization according to application requirements
  - Convert application service specifications (maximum error and confidence) to necessary resource requirements (number of messages and how often to resynchronize)
Receiver-Receiver Synchronization

- Idea from CesiumSpray and RBS

Sender-Receiver

Receiver-Receiver
Protocol Design

- Some sensors act as senders
  - Broadcast $n$ packets to neighbors
  - Receivers record the time according to local clock, and sends back summary to sender
  - Sender computes clock skew slopes and broadcasts to all receivers
  - Each receiver knows its clock skew slope relative to other nodes in the neighborhood
Synchronization Overhead

- Calculate $n$, given maximum synchronization error, $\epsilon_{max}$, and confidence probability, $P$

- Assuming Gaussian distribution of clock frequency at nodes, analyzing the protocol gives the following:

Theorem 1: 

$$P(|\epsilon| \leq \epsilon_{max}) = 2 \text{erf} \left( \frac{\sqrt{n} \epsilon_{max}}{\sigma} \right)$$
Probability of Achieved Error

Ratio = $\frac{\epsilon_{\text{max}}}{\sigma}$
Synchronization Interval

- How often to resynchronize, $T_{sync}$, depends on the application-specified maximum error, $\gamma_{max}$, and physical clock drift, $\rho$

- $\sigma_{max}$ is max delay in reception time between any two receivers

- Theorem 2: $\gamma_{max} = \epsilon_{max} + (T_{sync} + \sigma_{max})\rho$
Theorem 2: \[ \gamma_{max} = \epsilon_{max} + (T_{sync} + \sigma_{max}) \rho \]
Summary: Thesis Contributions

- An adaptive cross-layer sensor network architecture for multi-scale applications
- Exploiting sensor network and application characteristics and requirements
- Design and evaluation of component protocols
  1. Data Service
  2. Transmission Scheduling
  3. Clock Synchronization
List of Publications


3. Adaptive Medium Access Scheduling for a Multi-scale Sensor Network Architecture. To be submitted to *14th IEEE International Conference on Network Protocols (ICNP)*, Santa Barbara, California, USA, '06


- Related publications (Joint work)


Questions and comments?

More details (papers and code) at http://www.cs.rice.edu/~santa/research/
Backup Slides Follow
Adaptive Network Architecture

Application

Data Fusion

Data Service

Medium Access

Information Exchange Service (IES)

Synchronization Service

Localization Service

Radio

[Rice Tech Report '04]
Future Research Areas

- **Shorter term**
  - Dynamic adaptation without application-aid
  - Abstractions to support reliability and QoS

- **Longer term**
  - Heterogeneous networks and nodes
  - Enabling pervasive environments
Upper bound

- $\Delta^2 + 1$ slots enough for any neighborhood
- Minimum no. of slots needed $= \Delta^2 + h$
Lower bound

- Average degree gives a clique of size $\Delta + 1$
- Minimum no. of slots needed $= \Delta + h$
Inter-cell Interference

- To support parent->child & child->child communication
- If node $m$ was near node $g$, it cannot know about 2/3
- All nodes locally broadcast after sub-tree is done
- Node $m$ now knows about allocated slots
Communication Interfaces

- Scheduling gets information about traffic pattern through the NPI
  - Put/Get – corresponds to down/up stream traffic
  - Parent/Peer/Cell – corresponds to scope of communication
  - Each interface specifies required periodicity
No. of nodes vs. 1-hop neighbors

PhD Thesis Defense | Computer Science | Rice University
No. of nodes vs. 2-hop neighbors
Empirical Total Latency

- Finding chromatic polynomial is NP-hard
- Empirical formula can be used to interpolate points
Evaluation: Error Vs. packet sent

![Graph showing the relationship between Relative Clock Skew (sec) and Number of packets sent. The graph shows a trend where the Relative Clock Skew decreases as the Number of packets sent increases.](image)
Adaptivity using *Selectors*

- Hop proximity is relevant for some applications
- But a collaborative set of nodes might be based on other criteria
  - Vibration signatures from similar machines on fab floor
- Beacons have *selectors* associated with them
  - $Selector = (\text{attribute}, \text{value}, \text{operator})$
  - $Selectors = \text{NULL OR } Selector \ (\text{AND|OR} \ Selector$
  - Nodes associate with drum if $Selectors$ matches
- Applications use Selectors to adapt hierarchy
Hierarchy Formation Latency

- Latency increases logarithmically with network size

\[ \text{Startup} \sim \log (N) \]
The average number of drums at each level is inversely proportional to the square of the diameter at that level.

\[ \frac{n_i}{n_{i+1}} = \left( \frac{D_{i+1}}{D_i} \right)^2 = \alpha^2 \]
Fault Tolerance

- Routing protocol ensures
  - High percentage of alive nodes can be reached even with high degree of node failure
My Assumptions

1. Bi-directionality of links
   - True for limited range

2. Interference range = Transmission range
   - Techniques such as Radio Interference Detection (RID) [Infocom ’05] provides interference range

3. Clock synchronization present
   - Provided necessary clock synchronization

4. Have 2-hop node information
   - Easy for static networks
Erf function

\[
\text{erf}(x) = \frac{\int_0^x e^{-\frac{t^2}{2}} \, dt}{\sqrt{2\pi}}
\]
Example: Intel Fabrication Lab

- Deployment in ultra-clean facility for chip fabrication
- Controlled temperature and humidity
- Equipment monitored by vibration sensors
- Breakdown of any equipment has critical impact on production line and results in significant losses
- Wireless sensors monitor vibrations of machines and temperature/humidity
Example: Intel Fabrication Lab

- Need measurement at different scales
  - Coarse monitoring at regular time intervals
  - Finer scale less frequently
  - Anomalous data requires drilling down to the problem region
- Complex compression algorithms utilized at each level for data reduction
Multi-Scale Monitoring
Finer-Scale Drilling Down
Multi-Scale Monitoring
But ... practical constraints inhibit standard approaches to wavelets
- Irregular sampling of data
- Physical proximity need not imply correlation of data
- No natural multi-scale hierarchy
COMPASS Network Architecture

- Adaptive clock Synchronization
  - Subscribes to synchronization needs
  - Published at IPSN, April 2004

- Publish-Subscribe based notification service
  - Modules register with interest in a specific resource
  - E.g.: App -> MAC

- Enables multi-scale application support
  - Publishes routing hierarchy and NPI
  - To be published at DCOSS, June 2005

- Scheduled medium access for the multi-scale data service
  - Subscribes to hierarchy, NPI, fusion functions
  - To be submitted
Layering is a design principle, and is important for complex systems

- Each layer is independent of another layer
- Service provided by one layer is solely dependent on service by layer directly below

Applications and networking are usually developed separately with no interaction
Design Goals

1. Cross-layering for relevant information
   - Example: MAC knows communication schedule

2. Adaptable for specific application
   - Example: Routing has cost and latency tradeoff

3. Network Programming Interface (NPI)
   - Example: Logical naming instead of physical naming
Related Work: Routing

- Traditional sensor routing – single scale
  - Directed Diffusion, Trickle
- Multi-scale approaches
  - Dimensions, Fractional Cascading
  - Regular grid structure – needs localization
- Communication abstractions
  - Abstract Region – neighborhood communication
COMPASS: Multi-scale Architecture

- Multi-scale architecture enables
  - Local collaboration
  - Multi-resolution data
  - Extreme scalability
- Abstraction middleware enables
  - Ease of programmability
  - Efficient networking
Survey of MAC Protocols

Classification according to 3 design decisions
- Number of physical channels used
- Organization of the nodes
- Notification of the nodes
Multiple Channels

Channels

Single
(Simple radio, Most Sensor MAC Protocols)

Double
(Data + Control)

Multiple

FDMA
CDMA
Notification of the nodes

- Notification
  - Listening (CSMA)
  - Wakeup (Need more than one channel)
  - Schedule (TDMA)
Organization of the nodes

Organization

Random (Flexible)  Slotted (Middle-ground)  Frames (Efficient)
Slotted Protocols

- Low latency for Convergecast only
- Slot sequences are staggered
- Overflow policy adapts to the traffic load
  [BMAC, WMAN’04]
- Perfect match for a MAC for TreeCast [WMAN’04]
Random Access Protocols

- IEEE 802.11 access control
- Power Save (PS) Mode: Access point buffers traffic for the nodes
Random Access Protocols

- Preamble notifies receivers of upcoming transfer
- Shifts cost from receivers to transmitters
- Low Power Listening (LPL) and Preamble Sampling
  [Hill, Micro’02][WiseMAC, SenSys’03]
Random Access Protocols

- Accurately determines if channel is clear
- Clear-Channel Assessment (CCA)
- LPL, CCA and ACK can be tuned by application
  [BMAC, SenSys’04]
Slotted Protocols

- Synchronize nodes into slots
- Implement duty cycle within each slot, which determines energy saved
- 802.11 style data transfer using RTS/CTS
- Overhearing avoidance & Streaming sequence of messages [SMAC, Infocom’02]
Slotted Protocols

- Adaptive duty cycle over SMAC
- Nodes have very short *Active* periods, and go to sleep if no traffic is detected
- Automatically adjusts to fluctuations in traffic [TMAC, SenSys’03]
Sink-Based Scheduling

- A central base-station, or sink or cluster-head computes the TDMA schedule
- Sensors inform cluster-heads of traffic demands, which are addressed in the next scheduled frame

[IBM Systems Journal, 1995]

- Cluster-heads can be rotated to save energy for that node [LEECH, PACT, BMA]
Schedule-based Protocols

- Energy-efficient and fixed latency
- Challenge is to adapt TDMA protocols to work without any infrastructure
- Assumption 1: Clock synchronization present or achieved through TDMA frames
- Assumption 2: Interference range is equal to the transmission range
Weakness: Schedule-based approaches

- Compromise on latency bound
  - Much worse than contention based approaches
- Application traffic pattern not harnessed
  - Many traffic demands are strictly periodic
  - Source and destination is not any to any
- Network conditions might change
  - Should provide same service to data service layer, even with changing conditions
- Sink-based scheduling does not take 2-hop neighborhood into account
Distributed Scheduling

- Nodes know the 2-hop neighborhood
- Nodes broadcast their future traffic demand
- Sufficient information to choose one transmitter in a collision-free way
  - Priority given by hash fn of node and slot id
- Completely general communication assumed
  
  [NAMA, MobiCom’01][TRAMA, SenSys’03]
Network Programming Interface (NPI)

- Support for both *Get* and *Put*
  - Low-rate data generation, high interest ⇒ *put*
  - High-rate data generation, low interest ⇒ *get*
- 3 different kinds of addressing
  - Parent, Peer, Cell
  - *PutPeer* (Level, Selectors, Attribute, Value)
- *Reduction* interface
  - Example: maximum temperature in cell
Hierarchy Formation Overview

1. Nodes independently choose to become *drums*
   - Heterogeneous or random

2. Drums start broadcasting *beacons* with a scope

3. Other nodes associate with the drums – hop proximity

4. Repeat this process hierarchically

5. Nodes step up or down adaptively
Multi-hop Synchronization

- Introduce concept of levels: number of hops away from a sender
- Further away from sender implies lesser accuracy
- Routing neighbors $\Rightarrow$ Synchronization neighbors
- Timestamp of each message is transformed by the forwarding node locally