Design of Distributed Algorithm in WSNs for Engineering Applications

Jiannong Cao
Department of Computing
Hong Kong Polytechnic University
Outline

• Wireless sensor networks

• In-network processing:
  - existing solutions and drawbacks

• A structured approach to designing distributed algorithms in engineering applications of WSNs.

• Our work on WSN-based SHM
Wireless Sensor Networks (WSN)

- WSN consists of a large number of smart sensors nodes.
  - nodes are attached sensors and communicate through wireless connections
  - nodes can be mobile or stationary
  - no fixed infrastructure
  - a sink node connects to a server

- Sense the physical phenomenon
- Collect the information to sink
- Process it to get relevant results

- MCU: 12MIPS
- Power: 0dBm or 1dBm
- Frequency: 2.4G/830M/433MHz
- Data rate: 250k/80k/20kbps
Applications of WSN

- Medical monitoring
- Smart transportation
- Smart home
- Environmental monitoring
- Military field
- Wildlife Monitoring
- Disaster defense & response
- Fine agriculture
- Industrial monitoring
- Medical monitoring

SHM
In-network Processing

- For applications where the amount of sensory data or the scale of the network is large, collecting massive raw data to the sink node for (centralized) processing is infeasible:
  - Sensor nodes are much more limited in power, memory, and computational capacities
  - Communication consumes a lot of battery power
  - Centralized processing takes longer time

- In-network processing: data sensed is processed by sensor nodes themselves, while being transmitted to the sink.
  - facilitating a distributed, collaborative approach to computation
In-network Processing in WSN
the data aggregation approach

• Data aggregation: the process of aggregating the data from multiple sensors to eliminate redundant transmission

• Two steps to design data aggregation in WSN
  • Design an appropriate aggregation function
  • Design a routing/scheduling protocol
Aggregation Function

• Aggregation function: on receiving multiple data, how a sensor node processes the data so as to achieve the same final results as in the centralized approach.

• Types of aggregation functions:
  
  • *Constant output size*: an intermediate sensor aggregates multiple incoming packets into a single outgoing packet (Examples: Max, Min, Average⋯)
  
  • *Dynamic output size*: the size of the output changes with the inputs (e.g., two types of correlation model: self-coding, foreign coding⋯)
Example of Aggregation Function

- Design the aggregation function for calculating the maximum value of all the sensor nodes in the network.
Aggregation Routing Protocols

- Routing protocol: the aggregation paths from source nodes to the destination nodes

- Problem formulation
  - **Given**: a network model, and the aggregation function
  - **Objective**: to design a routing protocol to achieve some objectives such as maximizing system lifetime, minimizing energy consumption/latency,…
Aggregation Routing Protocols

Different Types of WSN Applications

• Environmental monitoring, forest fire monitoring, etc.

• Characteristics:
  • Low sampling frequency (per min/hour) → small amount of data
  • Simple computation (max, min, average) → algorithms are lightweight.
  • in-network processing is relatively easy
Limitations of Existing Data Aggregation in WSNs

- Targeting at conventional applications of WSNs (e.g. environmental monitoring), and design of in-network processing is relatively straightforward.

- Accordingly, the aggregation function is generally assumed to be handily available, and the focus is put on designing aggregation routing protocols.

- Assumptions for routing protocols
  
  - using the data aggregation can always obtain the same result as the centralized approach
  - each sensor node has unlimited computation capability
However, many engineering applications of WSNs have unique properties that make the design of data aggregation difficult.

Engineering applications of WSNs:

Monitoring important status, properties or parameters of objects/systems in applications of civil, mechanical, and electrical engineering, etc.
Engineering Applications of WSNs

- Structural Health Monitoring (SHM): using WSNs to identify healthy condition of infrastructures
- Seismic tomography: using WSNs to identify inner structure condition of seismic region
- Smart grid: using WSNs to estimate state in smart grid
Engineering Applications  
Structural Health Monitoring

- Data intensive: acceleration data sampled at 100Hz~1000Hz

- “Heavy” computation: based on complex matrix computation

![Diagram](image-url)
Engineering Applications
Seismic Tomography

• Data intensive:
  acceleration data
  sampled at
  \(x00Hz \sim x000Hz\)

• “Heavy” computation: least square estimation

\[
\begin{bmatrix}
A_1 \\
A_2 \\
\vdots \\
A_N
\end{bmatrix}
\begin{bmatrix}
S_1 \\
S_2 \\
\vdots \\
S_M
\end{bmatrix} =
\begin{bmatrix}
t_1 \\
t_2 \\
\vdots \\
t_N
\end{bmatrix}
\rightarrow
\begin{bmatrix}
A_1 \\
A_2 \\
\vdots \\
A_N
\end{bmatrix}
\begin{bmatrix}
t_1 \\
t_2 \\
\vdots \\
t_N
\end{bmatrix}
\]

\(A_1, A_2, \ldots, A_N\) (known)
\(S_1, S_2, \ldots, S_M\) (tbd)
\(t_1, t_2, \ldots, t_N\) (known)

the distance of p-wave in each grid
speed in each grid
travel time
Engineering Applications

Summary

• Data-intensive: not feasible to transmit raw data, so in-network processing is necessary

• Algorithms are complex:
  • **Computation intensive**: implementing in-network processing may be infeasible or cost for a sensor node
  • **Data-level collaboration of sensors**: Low level data fusion from different sensors: designing an aggregation function is not trivial
Low and High Level Data Fusion

- High level fusion: each sensor node can make a decision based on its own or neighbors’ data, local decisions can be combined afterwards.
  - Decision level fusion
  - Feature level fusion
- Low level fusion: final decision can only be made after collecting data from all the sensor nodes
High-Level Data Fusion

- Obtaining the maximum value within a WSN
Low-level Data Fusion

- The Eigensystem Realization Algorithm (ERA) in SHM
- Implements singular value decomposition of a matrix $H$: $H = \text{SVD}$

$$H = \begin{bmatrix}
y(1) & y(2) & \cdots & y(p) \\
y(2) & y(3) & \cdots & y(p+1) \\
\vdots & \vdots & \ddots & \vdots \\
y(q) & y(q+1) & \cdots & y(p+q+1)
\end{bmatrix}$$

Hankel data matrix $H$

$$y(k) = \begin{bmatrix} y^1(k) \\
y^2(k) \\
\vdots \\
y^m(k) \end{bmatrix}$$

Each element involves data from multiple nodes
data from sensor 1
data from sensor 2
data from sensor $m$

- When calculating $S, V,$ and $D$, data from all the sensor nodes are involved simultaneously
Algorithms with Low-level Data Fusion

• Large matrix with data from multiple sensor nodes

• Naive distributed algorithms would require data from different sensor nodes being exchanged frequently

• Algorithm complexity would be very high and generally cannot be implemented with sensor nodes.

• Designing distributed version for these algorithms is challenging!
Requirements of Algorithm Design

- **Communication efficiency**: without collecting raw data to a center, reducing wireless transmissions.

- **Computation efficiency**: able to be implemented in resource-limited wireless nodes.

- **High accuracy**: able to achieve the similar quality as the centralized algorithms.
Algorithm Design Steps

- **Step 1**: design appropriate aggregation functions for these algorithms
- **Step 2**: select appropriate type of routing structure
- **Step 3**: design optimal routing protocols under various constraints
Basic Aggregation Functions

- Type 1: Combining multiple raw data
- Type 2: Combining a single raw data with an intermediate result
- Type 3: Combining multiple intermediate results
Composite Aggregation Functions

\[ F(\{x\}) \quad F(x, si) \quad F(\{si\}) \]

\[ F(si, \{x\}) \quad F(\{si\}, x) \quad F(\{si\}, \{x\}) \]
How to identify basic aggregation functions for complicated algorithms?
• Generally algorithm-specific

• But two approaches may help
  • Divide and conquer
  • Incremental algorithm
Approach 1: Divide and Conquer

- Example: the ERA algorithm for SHM, whose objective is to obtain mode shape of a structure.

ERA:

\[
\begin{bmatrix}
\text{Raw data Hankel matrix}
\end{bmatrix}
\]

\rightarrow

SVD

- Mode shapes: vibration pattern of a structure – used for damage detection, structural modeling, etc.
Designing Distributed ERA

- Adopt the ‘divide and conquer’ approach to identify mode shape in a distributed manner.

Centralized ERA

- Partitioning the network into clusters.
- CH in each cluster identifies local mode shapes.
- Local mode shapes are assembled afterwards based on the least square estimation (LSE).
Aggregation Function for the Distributed ERA

- Divide and conquer approach: we can identify two basic aggregation functions of Type 1 and Type 3

How multiple intermediate results are combined? Via LSE

How multiple raw data are combined? Via ERA
Generalizing the Approach

• Key idea: Transform low-level data fusion to high-level fusion, the feature-level fusion

- Divide sensor nodes into clusters
- Low-level data fusion within each cluster
- Fusing results from all clusters

- This approach generates two aggregation functions of type 1 and type 3
Generalizing the Approach

Examples:

- Subspace identification (SI) and
- Frequency domain decomposition (FDD)

  - Classic algorithms for identifying modal parameters (including mode shapes and natural frequencies in SHM) based on SVD.
Approach 2: Incremental Algorithms

• Given a sequence of input, find a sequence of solutions that build incrementally while adapting to the changes in the input
Transform Centralized Algorithms to Incremental Algorithms

- Example: the ERA algorithm for SHM, used to obtain mode shape of a structure

ERA:

\[
\begin{bmatrix}
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\end{bmatrix}
\]

Raw data Hankel matrix

\[ \text{SVD} \]

(b) Mode 1 (Freq. = 0.9 Hz)

(c) Mode 2 (Freq. = 5.7 Hz)

- Can we implement distributed ERA by designing an incremental version of the SVD?
Incremental SVD

- Calculate the $U$, $S$, $V$ of the initial input data matrix $A$

- Then update $U$, $S$, $V$ using the new data $T$

- Result is the same as calculating the $USV^T$ using data matrix

Designing Distributed ERA

- Adopt incremental SVD to identify mode shape in a distributed manner

Centralized ERA

Distributed ERA

- Implement the SVD based on initial sets of data from a few sensor nodes

- The results of SVD \((U, S, V)\) travel in the network, and will be incrementally updated when meet any node on the route.
Aggregation Function for the Distributed ERA

• Incremental updating approach: we can identify two basic aggregation functions of Type 1 and Type 2

How multiple raw data are combined? Via SVD

How a single raw data can be combined with an intermediate result
Generalizing the Approach

- Key idea: result is updated incrementally along a path in the network

- Part of the raw data are utilized to obtain a preliminary result

- The result is updated when passing every node in the network

- This approach generates two aggregation functions of type 1 and type 2
Generalizing the Approach

Examples:

- Subspace identification (SI) and

- Frequency domain decomposition (FDD)
  - Classic algorithms for identifying modal parameters (including mode shapes and natural frequencies in SHM) based on SVD.

- Least square estimation
  - Smart grid and volcanic tomography)
Designing Aggregation Functions

Summary

• Three basic aggregation functions and the variants

• Design aggregation functions using (1) divide and conquer and (2) incremental algorithms.

• Comparison of the two approaches
  • Divide and conquer: relatively easy but cannot guarantee accuracy
  • Incremental algorithm: guarantee accuracy but may not be handy available
Algorithm Design Steps

• **Step 1**: design appropriate aggregation functions for these algorithms

• **Step 2**: select appropriate type of routing structure

• **Step 3**: design optimal routing protocols under various constraints
Types of Routing Protocols

Cluster-based

Features:

- Nodes are divided into clusters, and CHs obtain local results
- Local results from different clusters are combined only once
- Example: distributed ERA using divide and conquer

Chain-based

Features:

- Information is updated incrementally along a Hamiltonian path in the network
- Example: distributed ERA using incremental SVD
Types of Routing Protocols

Tree-based

Features:
- Leaf nodes send raw data to parents
- Non-leaf nodes collect raw data/intermediate results from their children, process them and send the results to their parents

Diffusion-based

Features:
- Each node keeps communicating with neighbors and updating the results.
- Example: LSE
Selecting Routing Structure

The routing structure can be chosen according to the available aggregation functions.
Algorithm Design Steps

- **Step 1**: design appropriate aggregation functions for these algorithms
- **Step 2**: select appropriate type of routing structure
- **Step 3**: design optimal routing protocols under various constraints
Design Optimal Routing Protocols

- Given aggregation functions and type of routing protocols, designing the optimal routing protocol is generally an optimization problem and, in most cases, is NP hard:
  - Optimal clustering
  - Hamiltonian path
  - Spanning trees

- Existing routing protocols did not consider constraints from real applications:
  - Constraint on accuracy of results
  - Constraint on node’s capability
  - ...

Constraints for Routing Protocols
- example of distributed ERA using clusters

- Nodes are divided into clusters and mode shapes from different clusters are assembled together.

- However, to combine local mode shapes, clusters must overlap.

- In addition, nodes in each cluster must be large enough to obtain accurate local result.
Constraints for Routing Protocols
- example of distributed ERA using clusters

- Given: the network model and energy consumption model

- Objective: divide the deployed sensor nodes into a number of clusters such that the overall energy consumption is minimized

- Subject to (constraints):
  - Clusters overlap and connected
  - Cluster size >p for damage detection quality
Our Work on WSN-based SHM
Energy Efficient Clustering for WSN-based SHM

- Cluster-based ERA

- Using “divide and conquer” to generate aggregation functions.

- Clustering routing structure

- Optimal clustering by considering network model, system model, clustering constraints

High Quality SHM using WSN

- High quality distributed ERA
  - Using “incremental SVD’ as the first step to generate aggregation functions.
- Different routing architectures including chain-based and tree-based.
- Optimal routing by considering network model, system model, computation capability and cost.


Energy-efficient Least Squares Estimation in WSNs

- Updating along a Hamiltonian path will have less energy consumption, but largest delay
- Updating along a SPT will have higher energy consumption, but with minimum delay
- A tradeoff between Hamiltonian path and SPT

Related Publications


Summary

• Engineering applications of WSN are data-intensive and require data-level collaboration among the sensor nodes.

• Designing distributed algorithms is challenging.

• In-network processing via data aggregation is a widely used approach.

• We developed a systematic and structured method to designing distributed WSN engineering algorithms.
thank you!