Localization in Wireless Sensor Network Using MDS

Rashmi Agrawal
Computer Science & Engineering Shri Ram Institute of Technology (SRIT) Jabalpur, INDIA, arash28@gmail.com

Brajesh Patel
Computer Science & Engineering Shri Ram Institute of Technology (SRIT) Jabalpur, INDIA, pat.brajesh@gmail.com

Follow this and additional works at: https://www.interscience.in/ijssan

Part of the Digital Communications and Networking Commons, and the Electrical and Computer Engineering Commons

Recommended Citation

This Article is brought to you for free and open access by Interscience Research Network. It has been accepted for inclusion in International Journal of Smart Sensor and Adhoc Network by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.
Localization in Wireless Sensor Network Using MDS

1Rashmi Agrawal
Computer Science & Engineering
Shri Ram Institute of Technology (SRIT)
Jabalpur, INDIA
E-mail: arash28@gmail.com

2Brajesh Patel
Computer Science & Engineering
Shri Ram Institute of Technology (SRIT)
Jabalpur, INDIA
E-mail: pat.brajesh@gmail.com

Abstract— In this paper, determining the localization of nodes in a Wireless Sensor Network is a very important task, which involves collaboration between sensor nodes. Localization is a fundamental service since it is relevant to many applications and to the network main functions, such as: routing, communication, cluster creation, network coverage, etc. Collaboration is essential to self-localization, so that localization can be accomplished by the nodes themselves, without any human intervention. In this paper, we first analyze the key aspects that have to be considered when designing or choosing a solution for the localization problem. Then, we present MDS localization algorithm. With this analysis of results simulated. We identified the results in topologies by taking different cases and we have addresses shortcomings, which are caused by anisotropic network topology and complex terrain, of existing sensor positioning methods. Then, we explore the idea of using multidimensional scaling technique to compute relative positions of sensors in a wireless sensor network. A distributed sensor positioning method based on multidimensional scaling is proposed to get the accurate position estimation and reduce error cumulation. Comparing with other positioning methods, with very few anchors, our approach can accurately estimate the sensors’ positions in network with anisotropic topology and complex terrain as well as eliminate measurement error cumulation. We also propose an on demand position estimation method based on multidimensional scaling for one or several adjacent sensors positioning. Experimental results indicate that our distributed method for sensor position estimation is very effective and efficient.

Keywords- Localization; Wireless; sensor; Network; Topology; Simulation.

I. INTRODUCTION

A Wireless Sensor Network (WSN) consists of a large number of tiny wireless sensor nodes (often referred to as sensor nodes or, simply, nodes) that are, typically, densely deployed. Nodes measure the ambient conditions in the environment surrounding them [5]. These measurements are, then, transformed into signals that can be processed to reveal some characteristics about the phenomenon. The data collected is routed to special nodes, called sink nodes (also called Base Station, BS), in a multi-hop basis [9]. Then, typically, the sink node sends data to the user via Internet or satellite, through a gateway. Though, depending on the distance between the user and the network, a gateway might not be needed (local monitoring).

Combining the advantages of wireless communication with some computational capabilities, WSNs allow for a wider variety of applications than traditional networks: environmental monitoring, health, surveillance, catastrophe monitoring, structural monitoring, security, military, industry, agriculture, home, traffic monitoring, etc. Nevertheless, opposing to traditional networks, WSNs are useful only if sensor nodes are aware of the environment surrounding them [2, 11]. For instance, each sensor could only monitor its region and send the collected data to the sink node. However, the great potential of WSNs lies in its ability to correlate collected data in time and in space. This is the reason why synchronization and localization are fundamental tools to WSNs [1].

In this paper, we will focus only on the localization problem. Localization refers to the ability of determining the position (relative or absolute) of a sensor node, with an acceptable accuracy. In a WSN, localization is a very important task; however, localization is not the goal of the network. In fact, localization is a fundamental service since it is relevant to many applications (target tracking, intruder detection, environmental monitoring, etc.), which depend on knowing the location of nodes. Localization is also relevant to the network main functions: communication, geographical routing, cluster creation, network coverage, etc. Even collaboration typically depends on localization of nodes.

WSNs originated a new collaboration concept. Traditionally, collaboration exists within the same group of nodes, even though they move (node-centric computing) [10]. In WSNs, collaboration occurs among nodes located in a certain region, which means that the group of nodes may not be the same (location-centric computing). For instance, if a node leaves a predefined region, it stops collaborating with other nodes. However, besides localization-based collaboration, it is possible to identify other ways to collaborate, based whether in monitoring a certain phenomenon or in the hardware characteristics of the nodes themselves [2].

In this paper, we first analyze the key aspects that have to be considered when designing or choosing a solution for the localization problem. Then, we present MDS localization...
algorithm. With this analysis of results simulated. We identified the results in topologies by taking different cases and we have addresses shortcomings, which are caused by anisotropic network topology and complex terrain, or existing sensor positioning methods. Then, we explore the idea of using multidimensional scaling technique to compute relative positions of sensors in a wireless sensor network. A distributed sensor positioning method based on multidimensional scaling is proposed to get the accurate position estimation and reduce error cumulation. Comparing with other positioning methods, with very few anchors, our approach can accurately estimate the sensors’ positions in network with anisotropic topology and complex terrain as well as eliminate measurement error cumulation. We also propose an on demand position estimation method based on multidimensional scaling for one or several adjacent sensors positioning [3]. Experimental results indicate that our distributed method for sensor position estimation is very effective and efficient.

This paper is organized as follows. In section 2, we present the key aspects network localization problem. In section 3, we have presented the description of Wireless sensor network. In section 4, the MDS localization algorithm and analysis of this algorithm is described. In section 5, the simulator is described. In section 6, we have shown the simulated results and, according to the aspects described in sections 2, 3, 4, 5 and 6, Section 7 provides some conclusions and future work perspectives.

II. NETWORK LOCALIZATION PROBLEM

Location service is a fundamental building block of many emerging computing/network in paradigms. For example, in pervasive computing knowing the locations of the computers and the printers in a building will allow a computer to send a printing job to the nearest printer. In sensor networks, the sensor nodes need to know their locations in order to detect and record events, and to route packets using geometric routing [16].

Manual configuration is one method to determine the location of a node. However, this is unlikely to be feasible for large-scale deployments and scenarios in which nodes move often. GPS is another possibility, however it is costly in terms of both hardware and power requirements.

Furthermore, since GPS requires line-of-sight between the receiver and satellites, it may not work well in buildings or in the presence of obstructions such as dense vegetation, buildings, or mountains blocking the direct view to the GPS satellites. Recently, novel schemes have been proposed to determine the locations of the nodes in a network where only some special nodes (called beacons) know their locations [7]. In these schemes, network nodes measure the distances to their neighbors and then try to determine their locations. The process of computing the locations of the nodes is called network localization.

III. WIRELESS SENSOR NETWORK

A wireless sensor network (WSN) [5] consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, enabling also to control the activity of the sensors [6, 15]. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer application, such as industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, and traffic control.

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting [12, 14]. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from hundreds of dollars to a few pennies, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth [7]. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.

Smart environments represent the next evolutionary development step in building, utilities, industrial, home, shipboard, and transportation systems automation. Like any sentient organism, the smart environment relies first and foremost on sensory data from the real world. Sensory data comes from multiple sensors of different modalities in distributed locations [17]. The smart environment needs information about its surroundings as well as about its internal workings; this is captured in biological systems by the distinction between exteroceptors and proprioceptors. PDABSC(Base Station Controller, Preprocessing) BST Wireless Sensor Machine Monitoring Medical Monitoring Wireless Sensor Wireless Data Collection Networks Wireless(Wi-Fi 802.11 2.4GHzBlueToothCellular Network, -CDMA, GSM)Printer Wire land(Ethernet WLAN, Optical)Animal Monitoring Vehicle Monitoring Online monitoring Server transmitter Any where, any time to access Notebook Cellular Phone PC Ship Monitoring Wireless Sensor Networks Roving Human monitor Data
Localization in Wireless Sensor Network Using MDS

IV. MULTIDIMENSIONAL SCALING

Some algorithms [3], either range-based or range-free, are based in one of several MDS (Multidimensional Scaling) classes. MDS is a set of data analysis techniques that display the structure of distance data as a geometrical picture [4]. Comparing to other solutions, MDS can achieve very accurate localization results (except for sparse networks), but at the expense of higher computational costs.

In general, each node applies MDS [8] to compute a local map, including only neighbors’ relative positions. Local maps are then merged together to form a global map. If enough anchor nodes are available (minimum of 3 for 2-D networks; minimum of 4 for 3-D networks), absolute positions can be computed, transforming the global map in an absolute map.

A. Algorithm used for localization

Let \( T = [t_{ij}]_{2\times n} \) be the true locations of a set of \( n \) sensor nodes in 2-D space. \( d_{ij}(T) \) represents the distance between sensor \( i \) and \( j \) based on their location in \( T \).

\[
X = [x_{ij}]_{2\times n} \text{ is the estimated locations of the set of sensor in 2D space. The pair-wise distances of sensors in } T \text{ are collected in distance matrix } D; \text{ we can use the classical multi-dimensional scaling algorithm to estimate the sensor location. If all pair wise distances of sensors are collected, we can use the classical multidimensional scaling algorithm to estimate the positions of sensors:}
\]

- Compute the matrix of squared distance \( D^2 \), where \( D = [d_{ij}]_{mn} \).
- Compute the matrix \( J \) with \( J = I - e \cdot e^T/n \), where \( e = (1, 1, \ldots, 1) \).
- Apply double centering to this matrix with \( H = -JDJ \).
- Compute the Eigen-decomposition \( H = UVU^T \).

Suppose we want to get the \( i \) dimensions of the solution (\( i = 2 \) in 2-D case), We denote the matrix of largest \( I \) eigenvalues by \( V_i \) and \( U_i \) the first \( I \) columns of \( U \). The coordinate matrix of classical scaling is \( X = U_iV_i^T \).

Before alignment we only know \( R \) and three or more anchor sensor’s physical positions \( T_1, T_2, T_3 \). Based on the local area are not available. When this happens, the iterative MDS is employed to compute the relative coordinates of adjacent sensors. We summarize the iteration steps as:

- Initialize \( X^{[0]} \) as random start configuration, set \( T = X^{[0]} \) and \( k = 0 \), and compute \( \sigma(X^{[0]}) \).
- Increase the \( k \) by 1.
- Compute \( X^{[k]} \) with above update formula and \( \sigma(X^{[k]}) \).
- If \( \sigma(X^{[k-1]}) - \sigma(X^{[k]}) < \epsilon \), which is a small positive constant, then stop; \( \epsilon \), which is a small positive constant, then stop;

Otherwise set \( T = X^{[k]} \) and go to step 2.

The \( \epsilon \) is an empirical threshold based on accuracy requirement. This algorithm generates the relative positions of sensor nodes in \( X^{[0]} \).

The above methods can estimate the relative locations of sensor nodes based on their pair wise distances. We also need position alignment techniques to map the relative coordinates to physical coordinates based on three or more anchor sensors.

B. Aligning relative positions

Since we hope to compute the physical positions of sensors eventually, it is necessary to align the relative positions to physical positions with the aid of sensors with positions known. For an adjacent group of sensors, at least three sensors physical positions are needed in order to identify the physical positions of remaining nodes in the group in 2-D case. Thus, each group of adjacent sensors must contain at least three nodes with physical positions known, which can be anchors or nodes with physical positions calculated previously.

The alignment usually includes shift, rotation, and reflection of coordinates. \( R = [rij]_{2\times n} = (R_1, R_2, \ldots, R_n) \) denotes the relative positions of the set of \( n \) sensor nodes in 2-dimensional space. \( T = [t_{ij}]_{2\times n} = (T_1, T_2, \ldots, T_n) \) denotes the true positions of the set of \( n \) sensor nodes in 2-dimensional space. Let node 1,2,3 be anchor nodes. A vector \( R_i \) may be shifted to \( R_i(1) \) by \( R_i(1) = R_i + X \), where \( X = R(1) - R_i \). It may be rotated counterclockwise through an angle to \( R_i(2) = Q_1R_i \), where

\[
Q_1 = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix}
\]

It may also be reflected across a line

\[
S = \begin{bmatrix} \cos(\beta/2) \\ \sin(\beta/2) \end{bmatrix}
\]

To \( R_i(3) = Q_2R_i \), where

\[
Q_2 = \begin{bmatrix} \cos \beta & \sin \beta \\ -\sin \beta & \cos \beta \end{bmatrix}
\]

International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN) ISSN No. 2248-9738 Volume-1, Issue-4, 2012 230
them, we compute $T_4, T_5, \ldots, T_n$. Based on the above rules, we have $(T_1-T_2, T_2-T_3, T_3-T_1) = Q_1Q_2 (R_1-R_2, R_2-R_3, R_3-R_1)$ with $R_1, R_2, R_3, T_1, T_2, \text{and } T_3$ known, we can compute $Q = Q_1Q_2 = (R_1-R_1, R_2-R_1, R_3-R_1) / (T_1-T_2, T_2-T_3, T_3-T_1)$ Then $(T_4, T_5, \ldots, T_n)$ can be calculated with $(T_4-T_1, T_5-T_1, T_n-T_1) = Q (R_4-R_1, R_5-R_1, \ldots R_n-R_1) + (T_1, T_1, \ldots T_1)$

C. Steps of Implementation
- Topology of a wireless sensor network is selected. It can be a fixed topology or anisotropic topology.
- Flooding starts from the starter node and it collects the distances from all the nodes in its range.
- Classical MDS algorithm is executed by the starter node.
- Relative positions of the nodes in the topology are calculated by using classical MDS.
- The relative positions of the anchor nodes are aligned to their original positions and this results to the aligning parameters (angle of rotation and scaling parameters)
- By using the aligning parameters, the remaining sensors relative positions are aligned to their physical positions.

V. PROBABLISTIC WIRELESS NETWORK SIMULATOR

Prowler is a probabilistic wireless sensor capable of simulating wireless distributed systems, from the application to the physical communication layer. Prowler, running under MATLAB, provides an easy way of application prototyping with nice visualization capabilities.

Prowler is an event-driven simulator that can be set to operate in either deterministic mode (to produce replicable results while testing the application) or in probabilistic mode that simulates the nondeterministic nature of the communication channel and the low-level communication protocol of the motes shown in figure 1. It can incorporate arbitrary number of motes, on arbitrary (possibly dynamic) topology, and it was designed so that it can easily be embedded in to optimization algorithms. Simulator simulates the radio transmission/propagation/reception, including collisions in ad-hoc radio networks, and the operation of the MAC-layer. Any application can be implemented on any number of motes. The radio definitions (propagation and MAC-layer) and the applications are plug-ins. The simulator can be set to deterministic operation mode (it's useful to test algorithms) as well as to probabilistic operation mode (which is even more useful to test algorithms. The parameters can be set from command line by sim_params. The parameter setting GUI can conveniently be accessed by pressing the Simulation Parameters button in the prowler GUI.

A. Applications
Each application consists of three files:
- Name_topology.m: topology information (coordinates of the motes)
- Name_application.m: the actual application code
- Name_params: defines application specific parameters. These parameters can be viewed and modified by pressing the Applications Parameters button.

The applications are event based.
Events:
- Init_Application
- Packet_Sent
- Packet_Received
- Collided_Packet_Received
- Clock_Tick
- Application_Finished
- Application_Stopped

Actions can be activated when events occur. These actions cause further events.
Actions:
- Set_Clock
- Send_Packet

VI. RESULTS AND DISCUSSION

In this paper, we have analyzed the localization of sensors in random topology. Figure 2, 3 and 4 shows the simulated results of real location position of sensors and relative location positions of sensors respectively and then, we have analyzed the. Figure 5, 6 and 7 shows the simulated results of estimated location positions of sensors and figure 8 shows the graph of error rates.

VII. CONCLUSION AND FUTURE WORK

In this paper we address shortcomings, which are caused by anisotropic network topology and complex terrain, of existing sensor positioning methods. Then, we explore the idea of using multidimensional scaling technique to compute relative positions of sensors in a wireless sensor network. A distributed sensor positioning method based on multidimensional scaling is proposed to get the accurate position estimation and reduce error cumulation. Comparing with other positioning methods, with very few anchors, our approach can accurately estimate the sensors’ positions in network with anisotropic topology and complex terrain as well as eliminate measurement error cumulation. We also propose an on demand position estimation method based on multidimensional scaling for one or several adjacent sensors positioning. Experimental results indicate that our distributed method for sensor position estimation is very effective and efficient.
For future work, we plan to carry out analysis of the communication costs during the operation of our methods. We will do experiments related to message complexity and power consumption. We also plan to investigate localization problems for mobile sensors in wireless ad-hoc sensor networks.

REFERENCES

Localization in Wireless Sensor Network Using MDS

Figure 4: Estimated locations of the real nodes

Figure 5: Real location position of sensors

Figure 6: Relative location positions of sensors

Figure 7: Estimated location positions of sensors

Figure 8: Graph of range-measure error Vs error rate