

Architecture for Distributed Source Localization in Wireless Sensor Networks

Soma sekhar Dhavala

ECpE Dept., ISU

Nov. 2004

Table of Contents

Abstract	: 3
Introduction	: 3
Algorithm for source location estimation	: 4
Architecture	: 6
The Protocol	: 9
Assumptions	: 14
Conclusions	: 15
References	: 16
Web Resources	: 16

Abstract:

In this project report, we propose a new architecture for distributed source localization in wireless sensor networks. The sensors, which are deployed randomly, form an ad-hoc network and establish a communication link amongst them that makes possible the application of various collaborative signal processing algorithms. We briefly describe the some recent developments in distributed signal processing with applications in wireless sensor networks (WSNs). It is from this application point of view that we review existing protocols for WSNs. We claim that the proposed architecture can form scalable ad-hoc networks with self-organizing medium access control (S-MAC). Further, we argue that it can also support source mobility and to certain extent is fault-tolerant. It will be shown that several variations are also possible trading-off latency to hardware complexity (rather availability).

Introduction:

Continued growth and advancements in fabrication of micro-electro-mechanical systems (MEMS) and actuators makes it possible to develop low-cost sensors in huge numbers. These sensors could drive the future of embedded systems and span a wide range of applications. For example, sensor networks could be employed to monitor wild life habitat or control the operating conditions of a nuclear power plant or just can be used to guide a robot operating under hazardous conditions. They could also have applications in unmanned surveillance of potential enemy lands.

As the number of sensors is large and as they are sensing almost the same physical phenomena (like temperature, pressure or acoustic energy or chemical concentration), the data generated by these sensors is highly correlated and it might be much more advantageous to process the data gathered by them before sending it for further analysis to extract any useful information. For example, sensors might be monitoring the earth temperature at a particular sight. Then the average of all the data collected might give a better estimate of the temperature of the region than a single sensor data would. This idea of extracting useful information from the sensors is widely known as collaborative signal processing. Recently, several algorithms in distributed computing were found attractive for use in sensor networks [1,2]. The reason being local communication requires less energy as opposed to global communication (a.k.a centralized vs. distributed) which consumes most of the precious resources. In this report, we propose an architecture that is based on cross-layer design intended to work with distributed CSP algorithms.

Source Location Estimation Algorithm:

Below we describe the source model and CSP algorithm to obtain the estimate the location of the source:

Omni-directional sensors are distributed over the field randomly. All the sensors within the radius of 15 units are considered active and will collaborate with other active sensors to identify the location of the source, in this case, an acoustic source. It is also assumed that no sensor is close (as much as 1 unit) to the target (or source).

The impinging acoustic energy is measured at the sensor. This energy is inversely proportional to the distance between the source and the sensor. To be more specific, the following model is considered:

$$s_j(i) = \frac{A}{(\theta_1 - x_j)^2 + (\theta_2 - y_j)^2} + e_j(i), \text{ where}$$

- $s_j(i)$ is the i^{th} sample of received energy at the j^{th} sensor
- N is the total # of samples being observed at each sensor
- M is the total # of sensors (active) in total
- $\theta_1, \theta_2, x_j, y_j$ are the x - and y - coordinates of the source and the j^{th} sensor respectively.
- $e_j(i)$ samples are i.i.d AWGN with zero mean and variance σ^2

We estimate the parameter by minimizing the least-squares function and since the parameters enters non-linearly into the equation it is non-linear least-squares estimation problem.

$$f = \frac{1}{MN} \sum_{j=1}^M \sum_{i=1}^N \left[s_j(i) - \frac{A}{(\theta_1 - x_j)^2 + (\theta_2 - y_j)^2} \right]^2.$$

In distributed/ decentralized processing, we minimize communication between sensor nodes and centralized processing unit by refining the parameter estimates based on the neighbor's rather than processing the accumulated data.

Let $\theta_{1,j-1}, \theta_{2,j-1}$ be the previous estimate, and r_j (a function of the data) be the current estimate of the distance between the source and the j^{th} sensor.

Then, $\phi_j = \tan^{-1}(\frac{y_j - \theta_{2,j-1}}{x_j - \theta_{1,j-1}})$ then,

$$\theta_{1,j} = x_j + r_j \cos(\phi_j),$$

$$\theta_{2,j} = y_j + r_j \sin(\phi_j)$$

Below, we describe a network architecture that would allow the application of the CSP algorithms of the above type.

Architecture:

The sensor networks fundamentally differ from cellular networks or any other networks in the sense that energy efficiency is the single most factor in the design of the protocols. In Cellular Networks and Mobile Ad-hoc Networks (MANETs), the design criterion to be met is QoS and not performing the task with minimal energy. Furthermore, WSNs are unattended, they often remain in hostile environments and their battery power is an extremely precious resource. Also, the very objective in using sensor networks is to sense a phenomenon and process the data. There is lot of communication between and/or among sensors and/or a base station. The sensors can be mobile or stationary and even the phenomena being observed could also be mobile. Thus, the design of a WSN architecture is particularly dependent on the application being considered [3-5]. Now, we state the requirements of the algorithm to be met by the architecture:

- *Ad-hoc network*

The nodes are scattered randomly, there is no pre-defined topology for the network. Even if there is a base station, the nodes communicate with the base station only at the end of the estimation process. Hence, the WSN is an ad-hoc network.

- *Self-organizing MAC*

No particular node plays the role of a Master or Cluster head: so to speak. Upon deployment, the networks themselves have organize and forma network. We can use protocols similar to Bluetooth which support up to 7 slaves connected a Master (piconet) and multiple piconets can be inter-networked to form larger networks. However, in the cluster/ Master type of frameworks, all the nodes connected to them communicate with the Master/ Cluster head, thereby consuming more energy and does not support the distributed-CSP (d-CSP).

- *Scalable*

The node density could vary from region to region as the sensor moves. Thus, the Architecture should be scalable in the sense of forming a network with 100 nodes or 1000 nodes!

- *Node insertion (new node)*

It might be the case that, if the source is moving, new nodes might have excellent receive signal strength (RSS) than some nodes which are already a part of the network and nodes having high RSS and thus high SNR would lead to better estimates of the source. Besides that, the network may start with only two nodes and could dynamically grow with time. This an operation such as adding a new node to the existing network.

- *Node deletion (node out-of-range)*

Some of the nodes might be at the verge of losing battery power or it is just that their RSS drops below a pre-defined threshold in which case these can no longer be useful in the d-CSP.

- *Directed graph (cyclical communication)*

The source location estimation problem can be thought of solving a non-linear optimization problem. We can regard the parameter updates as, typically, an initial estimate being modified by an additive corrective term that depends on the current data as well as the previous estimate of the parameter of interest. Thus, the parameter estimate has to be cycled from one sensor to the other. Thus, the network forms a directed-closed-graph. Several other topologies are also possible and that is certainly a topic worth considering (it might lead to better topologies for a given instance of the network)

The reader is referred to an excellent review paper comparing various protocols [6]. Now, we discuss whether there exist any suitable protocols. Almost all protocols rule-out the possibility of using complete contention-based protocols like CSMA-CA used in 802.11 for the simple reason that the nodes have to listen to the channel always and drain the battery of the nodes very quickly. However, we have to use some form of contention-based protocols during the network management frame. Alternatives to contention-based protocols are contention-less protocols like FDMA, TDMA or CDMA or a combination of them. Some trade-offs exist among them: for e.g., TDMA would be the simplest in terms of hardware and most energy efficient as the nodes would wake-up only during

transmission or reception phases. CDMA based communication would be most helpful if we can afford huge bandwidth but can not tolerate latency, where as, FDMA also offers low latency it may require complex hardware compared to TDMA. Different protocols use different combinations of contention-based and contention-less protocols to achieve S-MAC. The most promising protocol proposed for WSNs is Self-Organizing MAC (S-MAC) [3].

It divided the whole communication in to two parts: one is the network management phase and the other is the actual communication phrase during which actual data transmission and reception happens. However, independent links could be formed and they may never be able to join the full network. This may need some monitoring or knowledge about local estimates of the sources. We intend to modify the S-MAC framework to suit the current requirements. Similarly, many algorithms in literature does form either cluster-based on protocols or would form isolated sub-nets. Henceforth, we take a new look at the problem with ideas borrowed from S-MAC. Before that we make the following assumptions:

The Protocol:

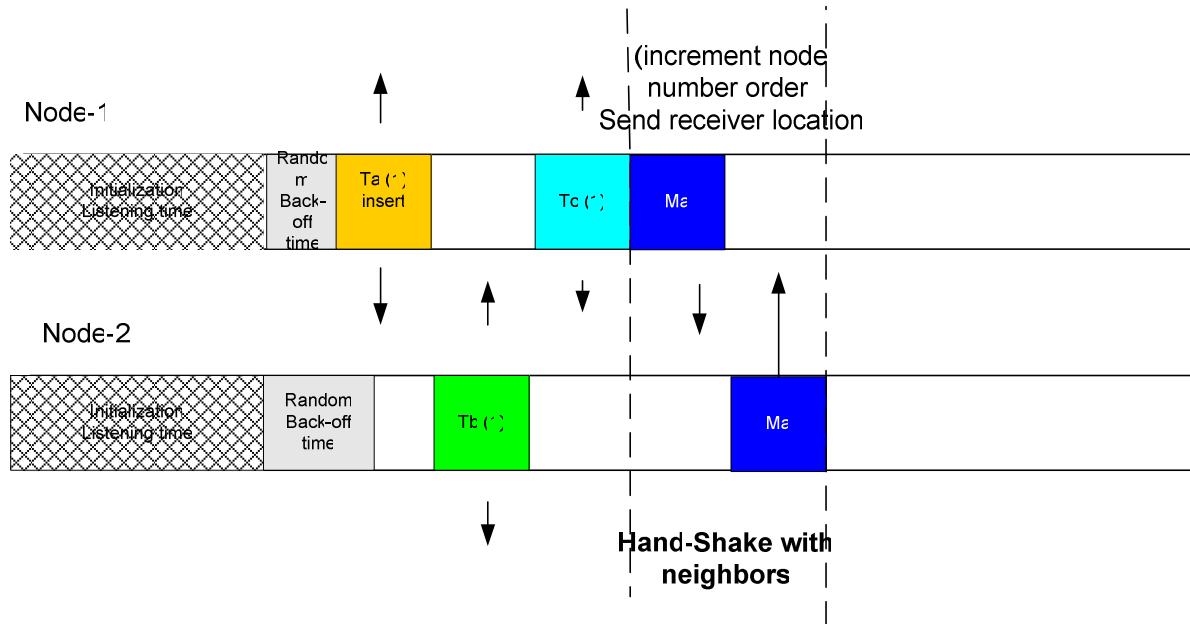
Like, S-MAC, we divide the network communication into to two phases: network management phase (NMP) (a.k.a boot-up phase, set-up phase) and data communication phase (DCP, steady-state phase). During the NMP, the network is either formed or is updated by inserting new nodes into the network or deleting nodes that no longer have a good RSS. Since a node has to perform a signal processing task only once in a cycle, it is

very appropriate to choose a TDMA schedule for contention-free access of the channel in the DCP phase. However, the length of the TDMA frame and the slot allocations for each of the nodes that are already included in the network has to be determined. This is one of the tasks of NMP.

Network Management Phase (NMP):

In the very beginning, no network is formed and all the active nodes are wanting to be associated with the network that is yet to be formed. These nodes would hear for any ongoing messages (it will be clear later what these bands are). After waiting for a certain amount of time that is much larger than a typical NMP super frame plus DMP super frame (will be defined what these super frames are in a moment). Then each of nodes would wait for a certain random back-off period and upon hearing a clear channel, the node that has the smallest random back-off time would broadcast a TA-insert type frame indicating that the node wants to be inserted in the network. Since this node did not hear any beacon frame or any other communication frames, it knows that it is the first one who is initiating the network. So it makes this information explicit by setting a flag field to one. The payload also contains the location of this node (lets say node-1). All the neighboring nodes upon hearing TA-insert-1 frame node-1, would direct a TB-frame with payload containing the self-location. Then node-1 determines closest neighbor by comparing the distances between all potential connected-neighbors. Then node-1 broadcasts to all the neighbors that it wants to be associated with node-2 (just a name given) and node-2 updates its local routing table that it would receive from as well as send to node-1 and vice-versa and a hand-shake (MA type frames) happens between these two

nodes in the freshly formed in the network. Based on pre-defined constant describing the number of nodes that can be added or deleted during the NMP, the process may continue or one cycle of communication (exchanging MB type frame). At the end of this phase, a beacon frame is transmitted by all the nodes (at this point only two). In the Figure below we show the timing diagram of the above process.



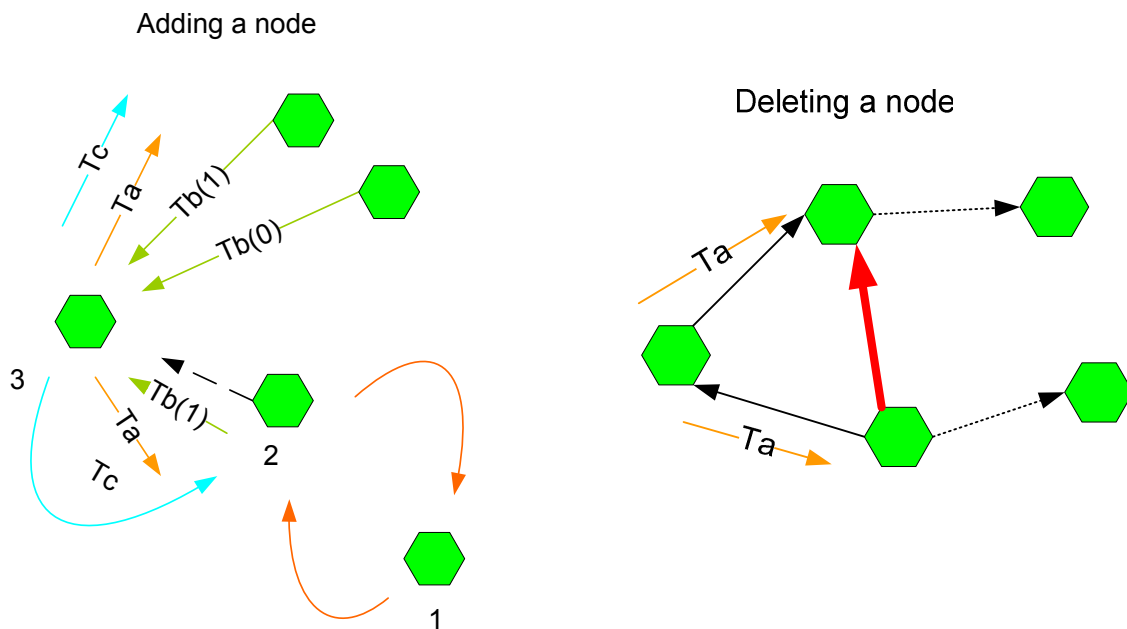
Suppose that the network exists and that node-1 and node-2 have established a link. And a beacon frame has just elapsed. Upon hearing the beacon frame, node-3 (arbitrary node), waits for a random back-off period and if the channel is clear, sends a TA-insert-0 frame (now node-3 knows that the network exists since it heard the beacon frame). Now either node-1 or node-2 or both can respond to node-3's request by sending their respective TB frames (on different frequency bands to avoid collisions). Upon receiving multiple TB frames with location information and the number of nodes that currently exist in the

network, node-3 choose the neighbor which results in minimum of the sum of the energies required for reception from the responding node and transmission to the transmitter locating in the responders routing table. Once node-3 chooses its neighbor, it would send another broadcast frame TC to all the neighbors that node-3 is being associated with node-1 (for e.g.). Upon hearing TC frame node-3, node-1 updates its local routing table. The receiver location in the routing table is now replaced with node-3's location.

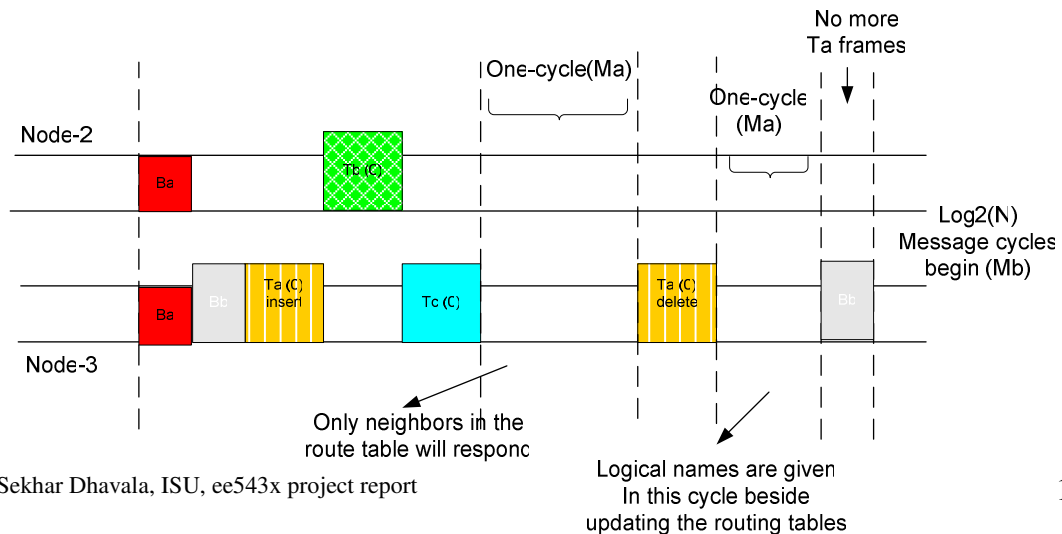
Now node-3 sets its logical index to 1, increments the total number of nodes it received from node-2 in TB frame and broadcasts MA frame (there is no actual content but there is only information about the logical index of the sender and increment total number of nodes message along with location of node-2). Now all the active nodes would hear on MA band for updated. Node-2 receives MA from node-1 (earlier it was node-3) and compares the location information in the payload with its position. Upon successful comparison, it increments the logical index in the payload by one and assigns this number to itself and also increments the total number of nodes in network to three. Now node-2 remained node-2. Likewise, node-1 (earlier) will be renamed to node-3. Now the link structure is 1-2-3-1. This would always be the case. At the end of each TC frame, the logical indices might represent different physical nodes. Now we have added node-3 to the network which is now being refereed to as node-1. Each node would again wait for TA frame either insert or delete request and if they don't hear anything in this TA interval, everybody starts actual data frame. Right from the beacon interval to the commencement of DCP phase, we call it as NMP super frame. And a DCP super frame

contains $\log_2(N)$ MB message frames. Now node-1 starts processing data and sends data to node-2 and sleeps. During this time, node-3 falls asleep. While node-2 starts transmitting data, node-3 awakens and starts receiving data.

Similarly a node can be deleted from the network. A node that does not want to participate would place a TA-delete frame and upon hearing TA-delete those nodes which have this location information in their local routing table will delete the entry. They update the routing table based on the routing table of the node that placed the delete request. Addition of a node and deletion of node are shown in the following figure:



In the Figure below, we show the timing diagrams of node insertion and deletion.



DCM begins at the end of the TA waiting period after N Ma frames. By this time, every node knows how many nodes are present in the network and their logical indices. This enables each node to determine when to wake-up for data reception and data transmissions. The energy efficiency vs. latency factor very much depends on the following factor.

- Velocity of the source
- Number of cycles in the DCP super frame
- Number of nodes that can be added or deleted during NMP super frame

The above protocol works under certain assumption:

Assumptions:

- Time synchronization: Since we are using TDMA for contention-free access, it is very much implied that good time synchronization is available.
- Dense Sensor network: If the density is low, then a very degenerate network could be formed in which case a different topology of the graph may work better. For example, if all the sensors are aligned in a straight line, the first node and the last are supposed to form one end of the link which may break-down.
- Further, we are also assuming that the sensor node is capable of receiving multiple bands at the same time. Like, multiple TB frames or TAinsert and TDelete frames.

Violating any of these assumptions could lead to very unusable protocol.

Conclusions:

We proposed an architecture for distributed source localization in wireless sensor networks. Instead of designing each of the layers (like MAC, Application Layer etc..), we followed a cross-layer design allowing us to customize the protocol for the application. This is particularly needed because, resources are very scarce in sensor networks and utmost care has to be taken to ensure that they are efficiently utilized.

The proposed protocol forms an ad-hoc network by randomly choosing two nodes initially and then growing the network grows by adding a node at a time. Though latency may be affected, the protocol could manage the formation of the network even if the number of nodes is highly variable. This insertion and deletion of nodes also enables us to accommodate moving sources. We use a TDMA based channel access mechanism for actual data communication. Using TDMA results in lot of energy savings (as opposed to CSMA-CA) and TDMA is the most natural choice for the application. The TDMA schedule and network formation are accomplished by a combination of contention-based channel access as well as FDMA. Using FDMA reduces the over all latency and helps in achieving a balance in energy savings and latency. Each node maintains a routing table containing the location information about the sender and the receiver. It would also contain its own logical index as well as the total number of nodes being present at that point of time. This information is used in TB, TA, TC and MA frames. Several variations are possible and it would be required to study how to reduce the latency in the NMP phase. Once the protocol evolves and is robust to node failures etc., simulation studies would be performed to verify the arguments.

References:

1. Michel Rabbat and Robert Nowak, “Distributed Optimization in Wireless Sensor Networks,” ICASSP’04, 2004.
2. Angelia Nedic and Dimitri P. Bertsekas, “Incremental subgradient methods for nondifferentiable optimization,” SIAM J. on Optimization, Vol. 12, pp. 109-138, 2001.
3. Soharabi K, Gao J, Ailawadhi V and Pottie G.J, “Protocols for self-organization of a wireless ad-hoc sensor network,” IEEE Personal Communications, Vol. 7, Issue 5, pp. 16-27, 200
4. Heinzelman W.R, Chandrakasan A and Balakrishnan H, “Energy-efficient communication protocol for wireless micro sensor networks,” Proceedings of the 33rd Annual Hawai Intl. Con. On System Sciences, pp. 3005-3014, 2000
5. Lindsay S, Raghavendra C. S, “PEGASIS: Power Efficient Gathering in Sensor Information Systems,” ICC, 2001
6. Praveen R, Ravi M, Shashidhar G and Udit S, “A survey on Sensor networks,” CS6392 course project report, UT-Dallas

Web-resources:

Literature Archive:

<http://www.research.rutgers.edu/~mini/sensornetworks.html>

Links to sensor networks:

<http://www.csif.cs.ucdavis.edu/~bharathi/sensor/snw.html>

Research at MIT:

<http://nms.lcs.mit.edu/publications/>