Self Organizing Wireless Sensor Network*

Katayoun Sohrabi, Jay Gao, Vishal Ailawadhi, Greg Pottie
Electrical Engineering Department
UCLA
Los Angeles, California, 90095
sohrabi@ee.ucla.edu gao@ee.ucla.edu vishal@ucla.edu pottie@icsl.ucla.edu

Abstract
Wireless sensor networks will enable low-cost connections between the physical world and telecommunications networks, with applications including security, industrial automation, remote exploration, and medical monitoring. After briefly discussing the underlying technology, we describe a protocol suite for self-organization of sensor networks, in which the number and topology of the sensor nodes are a priori unknown, and no timing has been made available. The scalable protocol very quickly establishes a TDMA and frequency assignment schedule, and then proceeds to establish energy-efficient routing. Simulation results will be demonstrated. We are in the process of porting the protocol to our network of 40 sensor nodes, and incorporating limited mobility features.

1 Introduction
With continued advances in integrated circuits technology, it is now possible to build very compact and low-cost devices that include sensing, signal processing, and wireless communications. Networks of such devices can then be constructed for such diverse purposes as security, machine monitoring, remote medical patient care, inventory control, and indeed many others in which a low cost connection between networks and the physical world can be made. It is imperative in such circumstances that the sensor network be self-organizing and energy efficient, since with a large number of nodes the installation cost would otherwise be large, and the energy storage system (hardware and maintenance) could also place a floor on cost. Under a joint research program of UCLA and the Rockwell Science Center, prototype platforms have been constructed which can be used for both gathering sensor data in support of target classification and location algorithm development, and to test the self-organizing network protocols to be described below. In this paper we describe an autonomous and scalable boot-up procedure, an energy efficient routing algorithm, and a protocol to allow interaction of pedestrians with the static sensor network. All of these operate without the need for manual configuration, prior information on the number or identities of the network elements, or synchronization.

The basic technological trend is towards integration of complete systems in single integrated circuits or multichip modules, with novel substrates. Micro electro-mechanical

*This research is supported by DARPA Contract number F04701-97-C-0010
MEMS techniques enable the fabrication of a wide range of sensors and actuators using standard CMOS processing, with the possible addition of one or two deposition steps. With the fabrication of suitable pads, these can be die bonded onto CMOS chips which perform the necessary signal processing. Additionally, the art of CMOS radios has considerably advanced in recent years, so that it is now possible to construct a complete radio from RF to baseband on one substrate. This may include MEMS RF components such as spiral inductors, or some small number of discrete passive components to reduce the energy consumption of the RF circuits. It is also possible to make use of the bonding wires or passive components embedded in ceramic packaging to improve the efficiency of the radios. Polymer substrates provide possibilities for both energy generation (through piezo-electric effects or photo cells) and sensing. Thus, a wide range of techniques are now available for production of low-power systems that combine sensing, actuation, communication and signal processing.

Energy efficiency is of great concern in sensor networks, since maintenance and hardware costs can easily be dominated by battery replacement. The energy budget is largely dominated by communications cost. The power consumption of CMOS circuits is proportional to the product of the transistor area, clock rate, and the square of the voltage swing. RF circuits need far higher clock rates, bigger transistors, and larger voltage swings than the baseband digital circuits. Thus, by reducing the number of bits to be transmitted by doing processing locally, we can radically reduce the power consumption of a sensor network. Further, in most applications, most of the time there is nothing of interest to report, yet we must be constantly vigilant for events. The solution is to dedicate some specialized low-power circuits to be looking for events using simple processing (e.g. energy detection) which has a high false alarm probability but the desired missed detection probability. Then more sophisticated processing is triggered, perhaps using general purpose processors. Only after this processing has been done will a sensor node communicate with its neighbors. If the detection certainty is high enough, it may inhibit its neighbors from communicating, and send the decision back to the remote user. Alternatively, it may excite its neighbors to become involved in cooperative processing to achieve a higher level of certainty before engaging in long-range communications. By means of this processing hierarchy, both the signal processing and communications energy costs can be minimized, while meeting false alarm and detection probability targets. A modular test platform has been developed in support of the development of such algorithms [1]. We are investigating a wide range of algorithms for target tracking and classification, including distributed node location algorithms, distributed beamforming, multi-node blind deconvolution and target separation, and FFT, wavelet, cepstral, and neural classifiers.

The networking protocols must also be designed with the energy constraints in mind. We must support a variety of functions: self-organization (since the network starts with no knowledge of neighbors or timing), energy efficient routing, communication with mobiles that interact with the network, and cooperative organization for local data fusion or beamforming. In this paper, we will describe all but the last function which is still under research. Section 2 describes the self-organizing algorithm and MAC, section 3 describes the routing algorithm, section 4 gives the extension to mobility, and section 5 presents our conclusions.
2 Link Layer Issues

The sensor network must be able to self organize at the lowest level, i.e. the member nodes must be able to form a connected multi-hop network dynamically and autonomously. The key design issues are the available bandwidth (which is large, compared to the per hop data rate) and available energy (which is small). The BOOTUP algorithm conserves energy by reducing the number of messages passed to build the network. This is done by combining the connectivity discovery phase with the link assignment phase and removing the requirement for network-wide slot synchronization. The resulting solution is a distributed and scalable algorithm which builds a flat hierarchy. The final product of this organization algorithm will be a connected multi-hop network, where each node follows a TDMA-like time schedule.

2.1 Self Organization Problem

The BOOTUP algorithm provides a means for a random collection of nodes to determine, collectively, local versions of the connectivity matrix of the network, and to establish links based on this information. The nodes do not have any prior information about the time reference, or location of other nodes or total number of nodes when the algorithm starts.

Assume there are R orthogonal channels available to our network. (e.g. through some combination of time, frequency, or code division). Consider a network of N nodes, with connectivity matrix $C = [c_{ij}]$. A link $l_{ij}$ exists from $i$ to $j$, if $j$ is able to receive signal from $i$ and $c_{ij} = 1$. Otherwise $c_{ij} = 0$. Let this network have a total of $L$ links, numbered from 1 to $L$. Two links $l_1$, from node $i$ to node $j$, and $l_2$, from node $k$ to node $m$, are defined to be interfering if at least one of the members of the following set $c_{kj}, c_{im}$ is equal to one. Now an interference free organization of the network will be a mapping, such that in the entire network no two interfering links have the same channel $r$ assigned to them. The size of this organization is the total number of distinct channels assigned. Finding a mapping where the total number of assigned channels is minimum is an NP complete problem for general topologies. Fortunately, we only require an interference-free mapping, regardless of its size. The important constraint is that this mapping must be found using minimal energy. Since the major energy cost is due to communicating messages between nodes, our algorithm must find ways to reduce the number of passed messages.

To summarize, the self organization procedure must form a connected multi-hop network, starting with no connectivity information or timing reference. The algorithm must also save energy. In order to do this, the algorithm enables nodes to find their neighbors by means of exchanges of a limited number of messages over the air. The messages enable nodes to change local information, and assign channels to the discovered link in a distributed fashion.

2.2 Channel Access Mechanism

Before description of the BOOTUP algorithm is given, the channel access mechanism for the sensor network must be described. In the sensor network nodes communicate intermittently, and their radios, which are a major energy consumer, need not be turned on at all time. This leads to a TDMA-like channel access mechanism for the sensor
network. Unfortunately, generic TDMA systems are synchronous systems, where all the nodes are slot synchronized (a costly endeavor). However, since a node only needs to know the transmission and reception epochs of its neighbors, and since these coordinates are determined at the time a link is formed, there is no need to form network wide synchronization, nor for a single TDMA schedule. Thus nodes do not reckon their frame structure on a single frame epoch. Also the internal structure of this frame structure, which we call a "SUPER FRAME", will be different for each node. Figure 1 illustrates the concept of the "SUPER FRAME".

![SUPER FRAME structure](image)

Figure 1: SUPER FRAME structure.

### 2.3 BOOTUP Protocol

In many classical organization algorithms two passes are needed over the entire network. In the first, all nodes follow a certain pre-assigned order, to broadcast their existence. By the end of the first pass, all nodes have a good idea about their local connectivity. The second pass enables the nodes to make decisions about the allocation of channels to links, and discover and remove interfering assignments. All the nodes must be listening to the channel for entire duration of both passes.

In our algorithm, the nodes turn off their transceivers, and only listen and transmit on the channel for a fraction of the duration of the time the self-organization procedure runs. There is also no a priori time synchronization. Our solution enables each node to find its radio neighbors one at a time. At the same time a link is formed, a channel is mutually agreed upon (based on the connectivity information of the two nodes involved) and assigned to the new link. This approach may result in less than optimum bandwidth utilization. However, since bandwidth is not a premium, we will accept this in exchange for energy savings that result from passing very few messages.

All nodes in the network run identical copies of the BOOTUP algorithm upon their respective wake-up times (which we assume are randomly distributed). The algorithm requires each node to transmit and wait to receive in a certain sequence a set of four message types called TYPE1, TYPE2, TYPE3, and TYPE4. TYPE1 message is an invitation message, and the rest comprise a three message handshake. By the end of a successful handshaking transaction, the two nodes involved will have information about
the TDMA schedule assignments of each other (effectively the one-hop away connectivity information), and also will have agreed upon a pair of time slots and an operational frequency band as the channel assigned to the link between them.

We assume each radio transceiver is able to tune to different frequencies while engaging in normal communications with neighbors (i.e. during the TDMA frame). However the entire BOOTUP algorithm takes place on a single fixed frequency, known to all the network nodes. A channel here is an ordered triplet \((i, j, k)\) where \(i\) and \(j\) are slot orders, and indicate a pair of slots within each node’s SUPER FRAME, and \(k\) is a frequency index, indicating the carrier frequency to be used for radio transmission and reception over the pair of slots. The channel assignment determines when and over what frequency the two nodes communicate.

Each node listens for an invitation for a random duration upon waking up. If nothing is heard, it sends an invitation and listens for a response. If nothing is heard, another invitation is sent after a random period. This continues, until a response is heard, at which point a three message handshake follows, and the pair assign a channel to this new link. Once a node has a neighbor, it will spend a portion of its time in TDMA mode, communicating with its neighbor(s), and a portion in BOOTUP mode to search for new nodes. The algorithm is given in Figure 2. Detailed description of the algorithm is given in [8].

![BOOTUP algorithm](image)

**Figure 2**: BOOTUP algorithm.

### 2.4 Performance of the BOOTUP algorithm

The algorithm is implemented in the PARSEC environment [9]. Its performance in terms of the amount of time and energy required to connect the network was investigated. Figure 3 gives the average performance for the time until the network becomes connected and also the duration the radios are powered on. Note that if the radio burns the same amount of energy while transmitting and receiving, then the duration of radio power up time will be good indication of the consumed energy. For purposes of comparison, the duration of the radio power-up time, for a single run of the Linked Cluster Algorithm,
LCA [7], is also included in this graph. Note that, the LCA assumes node synchronization at the slot level. It also requires knowledge of the total number of nodes, $N$, as well as an agreement between all the nodes about the location of transition time of node 0, i.e. frame synchronization. These are all requirements which do not have to be met in the BOOTUP algorithm.

It is clear that the algorithm is scalable. Also note that it connects the network quickly. The algorithm is able to organize the network without any node having prior knowledge of other node timing reference, as can be seen in Figure 3. We have been able to connect networks of up to 500 nodes with this algorithm. The energy saving advantage of the algorithm, is evident for large numbers of nodes, as can be seen from Figure 3.

3 Multi-hop Routing Issues

Energy management will be the primary routing consideration because it has a direct impact on network topology. Although most applications involved stationary sensor nodes, battery depletion can introduce some unpredictable changes in network connectivity. Since radio communication is one of the most energy consuming operations, topological change patterns in the network can be highly correlated to the radio traffic loading.

Sensor network traffic is highly asymmetric. Traffic flows in two primary directions: (1) **user-to-sensor**, which consists of command messages issued by the network user, through a base station or long range unit, to subsets of sensor nodes as query for status reports or trigger for some network functionality, and (2) **sensor-to-user**, which includes reports for target detection, activity, cooperative sensor function results, or in rare cases, even raw sensor data. Since the sensor-to-user flow dominates, a corresponding asymmetry in energy consumption is created, making the immediate neighborhood of the base station more susceptible to energy depletion. Therefore, an important task in traffic management is to maintain the network connectivity near the base station. Also, most sensor-to-user messages are prioritized based on the nature of its payload, which require
different QOS in terms of routing delay and reliability.

To meet the robustness and priority routing demands, a multi-path approach is pursued. However, since low overhead is still very important, instead of building node or link-disjoint routes, which offers the highest degree of robustness but an equally high cost in terms of overhead, we simply construct multiple overlapping spanning trees, each originating from one 1-hop neighbor of the base station. Each tree can provide a single route for each of its member nodes, and since most nodes will belong to multiple trees, multiple routes to the base station are made available to most sensor nodes. A hidden advantage to this approach is that by choosing which tree to use to route, each node can control which neighbor of the base station will relay the packets, therefore, gaining indirect control on traffic loading and energy consumption near the base station.

The construction of minimum-hop spanning trees follows simple distributed algorithms, with some slight modification to restrict its size. For any tree, say, tree $i$, and one of its member nodes, say, node $A$, two parameters will be computed for routing purposes: (1) a path metric, $M_i$, which is the sum of additive QOS parameter on all links and nodes between node $A$ and the base station on tree $i$, and (2) a tree life estimate, $L_i$, which measures the estimated number of radio transmissions that can be supported on tree $i$ before energy depletion occurs.

By using additive metrics, we assume that larger metric values correspond to lower QOS. Another related parameter would be $w_i$, which is a weight coefficient assigned to the $i$th packet generated by node $A$. Let $p_i$ denote the tree selected by node $A$ to route the $i$th packet. For performance evaluation, we assume node $A$ belongs to $m$ spanning trees, and define the average weighted metric to be:

$$\bar{M} = \frac{1}{N(m)} \sum_{i=1}^{N(m)} M_{p_i} w_i$$

(1)

where $N(m) = \sum_{j=1}^{m} L_j$ is the estimated number of radio transmissions that the combined resource of the $m$ trees can support. $\bar{M}$ also represents the average weighted QOS provided to each packet. We can then define a resource matrix:

$$r = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} M_1 \ldots M_1 & M_2 \ldots M_2 & \ldots & M_m \ldots M_m \\ 1 \ldots 1 & 2 \ldots 2 & \ldots & m \ldots m \end{bmatrix}$$

(2)

Each column of $r$ represents the capacity to route one packet from node $A$ to the base station. Note that $\forall i, r_{1,i}$ is the metric value associated with $r_{2,i}$, the tree ID number. Let $q_i(p_i)$ be any permutation of $\{1, 2, 3, \ldots, N(m)\}$ such that $N(p_{i-1}) \leq q_i \leq N(p_i)$, so that it maps tree decision $p_i$ onto $r_i$. Then, Eq (1) is equivalent to:

$$\bar{M} = \sum_{i=1}^{N(m)} w_i r_{1,q_i}$$

(3)

### 3.1 Sequential-Assignment Routing (SAR)

The goal of our routing algorithm is to minimize the quantity in Eq (3). Since the assignment of each packet to a path has to be made in real time, without the opportunity...
to observe the entire sequence of $w_i$ which is random, the minimization problem belongs to the class of sequential stochastic assignment problems described in [5]. The optimal policy described in [5], though it can be translated easily from maximization to minimization, requires an iid assumption on $w_i$, knowledge of its distribution, and calculation of many integral expressions. Although the iid assumption is later removed in [6], the calculation still cannot be made energy efficient. So we only assumed having knowledge of the cdf, $F_W(w)$, of $w_i$ and adopted the following suboptimal sequential assignment routing (SAR) policy:

For any node, say $A$, assume, without loss of generality, that $M_1 \geq M_2 \geq \ldots \geq M_m$, so that $r_{2,j}$’s are also in descending order. If by the best estimate, the network is determined to have resource capacity to route a total of $N$ packets and that for the $i$th packet, $w_i \in (F_W^{-1}(\frac{i-1}{N}), F_W^{-1}(\frac{i}{N}))$, where $1 \leq j \leq N$, then the optimal tree assignment will be $r_{1,j}$ with metric value $r_{2,j}$.

To maintain routing table integrity whenever failures occur, a local computation will be triggered to remove the invalid route. Also, path metrics and tree life estimates need to be kept on track with current network conditions because changes in the radio channel or additional energy consumption within the network may occur after the initial route computation. One way to accomplish this is to periodically trigger a route recomputation from the base station.

### 3.2 Simulation Results

We compare our algorithm to the classic Shortest Path algorithm, which will route all packets on the best possible path, regardless of priorities. A network with one base station and 37 randomly deployed sensors is simulated. We assume network traffic is generated by a subgroup of 5 sensors which detected extended target activities. Each member of the subgroup has equal probability of generating packets, and each packet has three possible priority settings, \{low, medium, high\}. Low and medium priority packets are assigned a weight coefficient of 0 and 1, respectively. Low priority traffic account for 30% of all traffic. Each link is assigned with one of two possible metric values to reflect different QOS. For the network simulated, a "good zone" is defined so that all links terminating in this zone have metric of 0; otherwise, the metric will be 30. Radios are assumed to operate at fixed power level and the per packet energy consumption is 10 energy units. Each node has metric of 1.0 and an identical power source with $10^6$ units of energy. Global metric recomputation is initiated by the base station after every 5000 sensor packets were received. There are two major traffic parameters: $p_h$, the probability that a packet generated has high priority, and the $w_h$, the weight coefficient assigned to it. Higher $p_h$ means more high priority packets will be sent through the system and their impact on the overall QOS will be significant. In our simulation, average weighted metric is compared for the SAR and the shortest path algorithm, under different high priority traffic settings.

In Figure 4, we compare the average metric for $p_h \in [0.05, 0.4]$. As expected, $\bar{M}$ increases when more high priority packets are generated. For $w_h = 50$, SAR is 38% to 40% better than shortest path; for $w_h = 10$, SAR is about 37.5% better. For both SAR and shortest path, the metric values grow roughly linearly with $p_h$. SAR algorithm is able to produce a lower metric because it preserves better paths by limiting usage, thus making them available for high priority packet. On the other hand, the shortest path algorithm makes
no such distinction based on priorities. It assigns many low-priority packets to the best path, which degrades the performance in the long run.

Another parameter of interest is the global recomputation interval. Frequent recomputation can significantly reduce network capacity. Figure 4 also shows that for both algorithms, the average network capacity takes a 20% loss when the frequency of recomputation increases by one order of magnitude. However, no significant difference in network capacity can be observed between SAR and Shortest Path.

![Figure 4: SAR & Shortest Path Comparison](image)

4 Mobile MAC Algorithm

The goal of the mobile MAC algorithm is to enable connectivity between a limited number of mobile users and an existing sensor network. For example, a pedestrian may wish to download data or issue instructions to the network via a handheld-device. This interaction should take place with the mobile having no prior knowledge of the network topology or channel assignments, and with minimal modification of the protocol of the sensor network. The protocol to be described below accomplishes these objectives by using the BOOTUP algorithm’s invitation messages, and lies above the MAC of the stationary network. Since the mobile is more likely to have possibility of renewing its energy source, the protocol concentrates on conserving the energy of the stationary nodes, in contrast to cellular systems in which the stationary devices have the more reliable energy supply.

4.1 Algorithm Elements

The mobile assumes full responsibility for establishing and breaking connections. It makes use of the existing MAC, with the slight change that responses from mobiles are assigned higher priority, through preallocation of one slot per stationary node to handle mobile connections. This enables a more rapid build up and down of connections. The primary messages are:
Broadcast Invite (BI) The stationary node invites other nodes to join.

Mobile Invite (MI) The mobile responds to BI to request a connection.

Mobile Response (MR) The stationary node accepts or declines the MI request.

Mobile Disconnect (MD) The mobile informs the stationary node of a disconnect; no response is needed.

4.2 Protocol

Notice that in contrast to cellular hand-off protocols [2,3], very little information exchange is required. The mobile keeps a registry of nodes within communications range, which it periodically updates by means of interception of BI messages. Whether or not a connection is made depends on the status of its other connections, location and movement information inferred from the history of the connection and communications range registries, or other factors that may be important to a particular application. The stationary nodes have no responsibility for tracking any information regarding the location of the mobile node or the quality of the link.

The protocol amounts to how the mobile manages its two registries, which due to the heavy attenuation in ground to ground propagation will not be very large. Here, by means of eavesdropping on BI messages the mobile begins filling its range registry with SNR’s, node ID’s, and other information on the local neighborhood. If the node is new, but the registry is full, a comparison is made based on the SNR to possibly knock out an old entry. Likewise, based on the SNRs in the registries a decision can be made to send out an MI to a listed node. For nodes that are already listed, reception of a BI message gives the opportunity to update link quality information, and decide whether a connection should be established or discontinued. An MI message, if issued, includes information regarding possible channel assignments, and will be issued only when strong SNRs are available to reduce the likelihood of frequent connects/disconnects with the same node.

Since in the application at hand the mobile node is likely to perform the role of an information sink, if the stationary node receiving an MI can establish a link (i.e., a time slot is available), then an MR message is issued. Otherwise, a short packet declining the invitation is sent. If the stationary node has between transmission of its BI and reception of the MI experienced some disaster, and no response is received, the mobile will after an appropriate period assume that no connection is possible. A mobile will likewise send an MD message to delete a connection if no traffic has arrived from a connected node for some period of time. MD messages will also be produced when SNR’s on links drop below a threshold. The stationary node then deletes the mobile node from its registry of connections.

We have simulated connections of mobile users to randomly placed nodes with satisfactory results. The boot-up, routing, MAC, and mobile MAC protocols have been tested together in a unified simulation environment. We are working towards a more flexible assignment of slots in the TDMA structure for the mobile/stationary node interface, and on porting all the algorithms to our sensor node platforms.
5 Conclusion

We have described a protocol suite that provides self-organization for a randomly deployed sensor network, with the aim of establishing energy efficient communications. Remaining tasks include porting the code onto the sensor nodes, and tuning the parameters according to testing in the field, as well as development of specific network procedures to deal with cooperative processing requirements such as data fusion among sensor nodes and array beamforming. The hardware effort is proceeding well, with the recent demonstration of self-organization of a 40 node network for data collection, and progress on integrated specialized processors for beamforming. With a number of recent government programs, a sensor network research community is developing, bringing with it the prospect of many exciting ideas from the hardware through to applications.

References


