SmartLOCUS: An autonomous, self-assembling sensor network for indoor asset and systems management

Cyril Brignone, Tim Connors, Geoff Lyon, Salil Pradhan
Mobile and Media Systems Laboratory
HP Laboratories Palo Alto
HPL-2003-41
June 7, 2005*

Indoor localization technology is a prerequisite to location aware services within building infrastructures. For large buildings, or frequently remodeled indoor spaces, a scalable, easily deployed localization system is required to avoid recurring costs of installation. Current localization systems require extensive manual intervention during installation. Additionally, localization computations are centralized, rendering them inflexible and not easily scalable. In this report, we describe a localization system formed by a wireless location aware network that is self-assembling. The nodes compute their locations autonomously in the face of topology changes and the localization algorithm is distributed enabling new nodes to localize themselves. These features make the system easy to deploy and scalable. We also suggest how this system can be used in commercial applications such as asset tracking.

* Internal Accession Date Only
Approved for External Publication

SmartLOCUS: An autonomous, self-assembling sensor network for indoor asset and systems management

Cyril Brignone, Tim Connors, Geoff Lyon, Salil Pradhan

Sentient Environment Department,
Mobile and Media Systems Laboratory,
Hewlett-Packard Laboratories,
1501 Page Mill Road
Palo Alto, CA 94304, USA

Abstract: Indoor localization technology is a prerequisite to location aware services within building infrastructures. For large buildings, or frequently remodeled indoor spaces, a scalable, easily deployed localization system is required to avoid recurring costs of installation. Current localization systems require extensive manual intervention during installation. Additionally, localization computations are centralized, rendering them inflexible and not easily scalable. In this report, we describe a localization system formed by a wireless location aware network that is self-assembling. The nodes compute their locations autonomously in the face of topology changes and the localization algorithm is distributed enabling new nodes to localize themselves. These features make the system easy to deploy and scalable. We also suggest how this system can be used in commercial applications such as asset tracking.

1. Introduction

Location aware services have proven to be very useful. Although adequate for outdoor positioning, GPS receivers do not usually function indoors. In addition, the accuracy requirements for interior applications are quite high, i.e., on the order of decimeters. For these reasons GPS is not an adequate solution for indoor positioning.

The typical solution to the indoor sensor localization problem consists of installing a physical wire with one end terminating at a known fixed position. A centralized database is manually programmed with a mapping between sensors and wires. Frequently, when an alternative sensor is desired at some currently serviced location, the wire is of the wrong type and must be replaced. Similarly, when a new location is to be serviced, a new wire must be installed. Additionally, changes must be reflected in the central database with an explicit manual update. Note too, that these activities involve both IT and construction skill sets, complicating the process and likely requiring a multi-disciplinary workforce. Although this approach solves the localization problem, it is prohibitively expensive.

Tracking physical assets is even more problematic, since inventories and equipment are often easily transported by building occupants without the knowledge (or permission) of building infrastructure maintainers. Unless the central database is fastidiously updated, physical assets become virtually lost.

A SmartLOCUS system is a collection of small embedded platforms most of which are attached to an asset, sensor, or other equipment whose location is important. Each platform or node is designed to automatically track its own location. Each node knows when it is moving and when its position is stable. Stable nodes help moving nodes determine where they are. Typically, a few nodes, called *infrastructure nodes*, are deployed with the expectation that they will move very infrequently. Other nodes are called *mobile nodes* and are expected to move frequently including exiting a space and reentering at a later time.

Since the technology for determining position is based partially on radio transmissions, these radios can be used to transmit sensor and location information to a back-end application. The back-end has no need to record the location of any SmartLOCUS node as they all know their own position. As long as several nodes are position stable, even an infrastructure node can be moved with no rewiring or database maintenance.

A SmartLOCUS system is self-assembling. When a new node is introduced into a space, it automatically finds its neighbor nodes and, using information provided by them, determines its own location within the space. In this way, new assets, sensors and other equipment can be trivially deployed into a space. A special case exists for the first few nodes to become active within a space. We describe a simple algorithm below to handle this case.

2. SmartLOCUS Architecture

The SmartLOCUS architecture is one of a number of component technologies being investigated by the Sentient Environment Department. Taken together, these technologies enable us to demonstrate end-to-end solutions in the building management space. A collection of ad-hoc networking nodes form the SmartLOCUS system, similar in concept to a number of ad-hoc networking solutions from academic research groups. However, the key differentiating features of SmartLOCUS are location awareness (with an accuracy of a few centimeters) together with self-assembly and autonomous management attributes.

Figure 1: Prototype SmartLOCUS nodes

Figure 1 shows two appliances each incorporating the first prototype hardware. The left hand picture is a stand-alone SmartLOCUS device suited for fixed infrastructure nodes. The righthand picture shows a SmartLOCUS that has been attached to an HP iPAQ via an expansion sleeve. This latter configuration represents a mobile node and suggests possible application usages such as pinpointing location on a displayed map. Figure 2 shows a prototypical configuration of SmartLOCUS nodes implementing a combined suite of monitoring applications, including environmental, asset, security, systems control and in-building navigation. Depending on user requirements, data-management and geo-visualization tools enable custom reconstructive views of the building including up-to-date sensor and location information. As an example of customized views consider that security staff may have access to people and object tracking. However, only object tracking may be available to a service technician trying to locate a faulty printer.

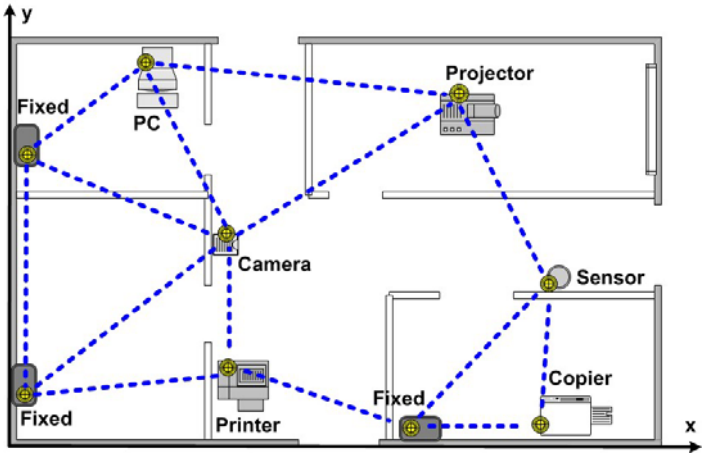

Figure 2: Building scenario example

2.1 Hardware platform

The SmartLOCUS hardware platform (Figure 3.) consists of two circuit boards. The main board is powered by an ARM7 microcontroller with sufficient Flash Memory and SRAM to provide a basic computation solution. A CPLD is used to perform additional digital operations and to provide interfacing to a wide variety of sensors. These sensors (and other IO devices) are outfitted via an auxiliary circuit board. This computation board has been used for several projects. The auxiliary IO board for SmartLOCUS incorporates a low power FSK radio transceiver module for data communications; several serial ports; a "OneWire" sensor network interface chip from Dallas Semiconductor; and an ultrasound transceiver, which is used for distance ranging between different nodes. As seen in Figure 1, these components may be embedded into an array of other appliances.

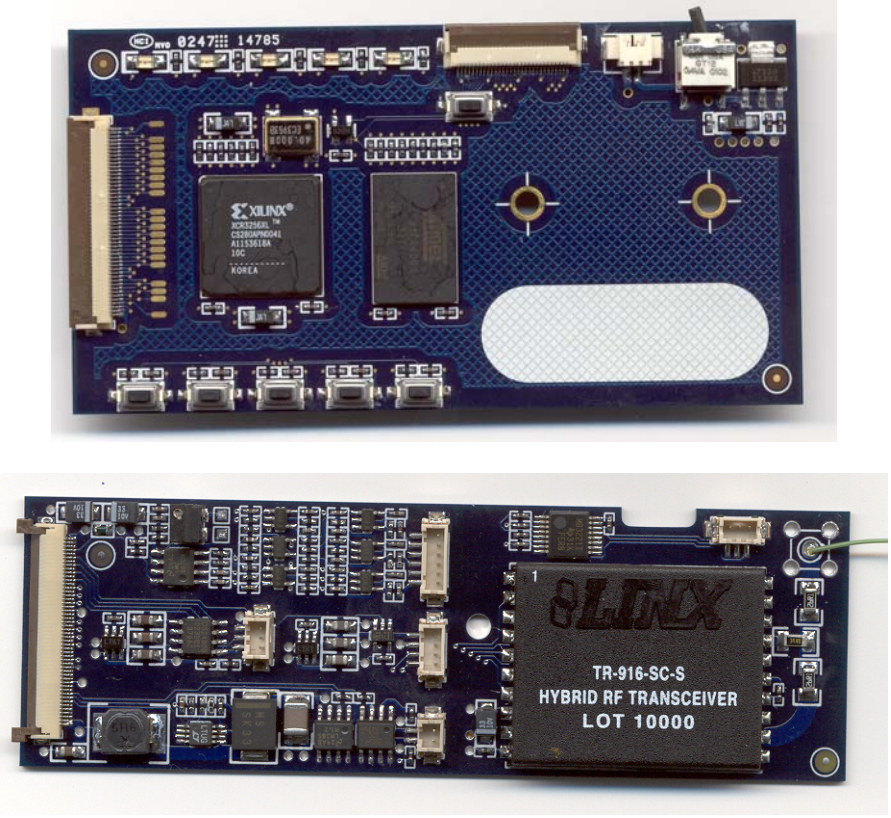

Figure 3: SmartLOCUS hardware platform, processor board and IO board.

2.2 Embedded software components

SmartLOCUS nodes use a modular embedded operating system, the e-Kernel, which has been configured to include the relevant system level functions required for multi-tasking, inter-nodal communications, distance ranging, position determination and sensor interfacing. Application code, in C or C++, is compiled and linked using the open source GNU toolset. Code download, flash programming and de-bugging are facilitated using a commercial JTAG interface controller.

2.3 Sensor interfacing

We customize the SmartLOCUS platform to a variety of sensors and sensing sub-systems. In traditional sensor systems, such custom interfacing is often performed individually, device at a time. Only raw data is transferred and then interpreted at a centralized location. With SmartLOCUS, we are considering the adoption of standardized interfaces to smart sensing devices (e.g. IEEE P1451) and actively researching methods of creating and distributing area based rules for localized control policies, where most of the sensor network traffic is processed and analyzed within localized regions.

3. Related Work.

Before describing the SmartLOCUS system in detail, we briefly describe two similar projects.

3.1 The Bat Ultrasound Localization System.

As does SmartLOCUS, the Bat System [1] from Cambridge University uses ultrasound to measure distances. Nodes whose positions are to be measured are outfitted with a small US transmitter, called a Bat. A fixed array of receivers listens for these transmissions. By using the relative time of arrival among multiple receivers together with their known fixed locations, a position for the Bat can be determined. This technique requires deploying receivers in advance and determining their positions by some other mechanical measuring system. Also the timing information is collected by many receivers. The Bat System assembles this timing information at a central computation site to do the actual calculations.

3.2 The Cricket Location-Support System.

The Cricket System [2] from MIT uses time of flight of both RF and US signals to measure distances. The distances measured are those from various "beacons" to "listeners." The listeners are mobile and receive both the RF and US signal from each beacon. A listener can then determine which beacon is the closest. In this system, each listener must know in advance where all the beacons are. As in the Bat System, the beacon positions must be determined though some alternative location technique. Also the accuracy of the location depends on the density of the beacons within the space.

4. SmartLOCUS Localization Algorithms

In this section, we discuss in detail the various algorithms used to achieve autonomous localization.

In the general case, the distinction between infrastructure and mobile nodes is logical only. The localization platforms can be physically identical and automatically recognize their own status as infrastructure or mobile. If an infrastructure node happens to move, it should stop helping other mobile nodes determine their location. When its motion stops, it can return to the role of helper for mobile nodes. Note that, even without some kind of motion detector, such as an accelerometer, a node can still detect that it is moving by comparing a newly calculated position with a previous one. However, there could be some lag in recognizing this condition. Since the first prototype SmartLOCUS lacks any form of instantaneous motion detection, we temporarily require that infrastructure nodes not be moved.

4.1 Inter-node distance calculation

SmartLOCUS uses the speed difference of an ultrasound (US) signal and a radio frequency (RF) signal to calculate distances between nodes. An US signal propagates at the speed of sound or 340 m/s. A RF signal propagates at the speed of light or 300,000,000 m/s. Because the RF signal is so fast compared to the US signal (900 000 times faster) we use 0 as an approximate travel time of an RF message from one node to another.

Figure 4 portrays a distance request sequence between two nodes A and N. As the first step in computing its location, node N sends a broadcast RF message requesting distance information. It is the reply to this request that is used to measure distance. Upon hearing the request, node A replies with the following.

- A RF message containing its own position and its own identity. This appears in Figure 4 as (xA, yA) and idA.

- A US blip consisting of 64 cycles of sound at 40 kHz.

The RF and US replies are sent at the same time. The difference in time of arrival, Δt, allows node N to calculate its distance to node A. This distance, called dAN, is given by dAN = Δt * speed of sound.

Since several nodes may hear the distance request, each one performs a random back-off in time before replying. This prevents collisions among the replies. For additional robustness, each replying node repeats their reply several times over a period of 15 seconds. Note also that the RF and US reply can be observed by nodes other than the original requestor. Any node hearing a reply can also calculate their distance to the replying node.

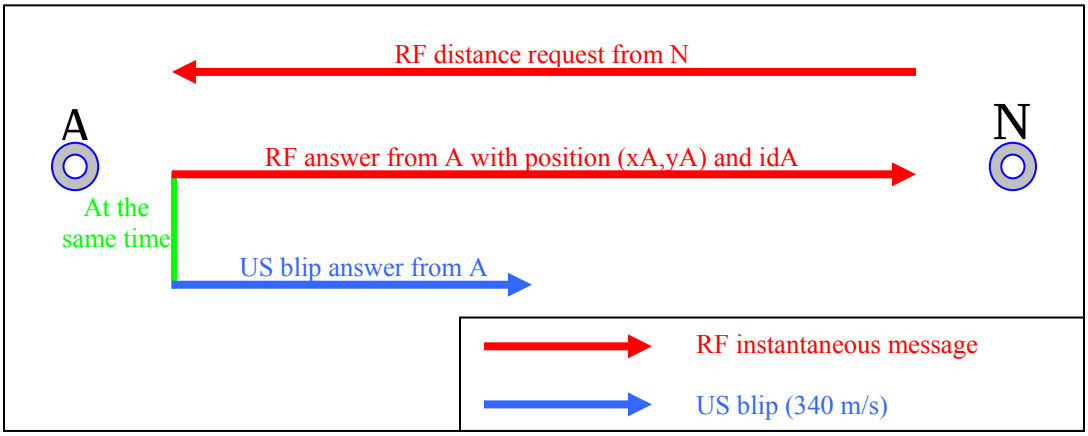

Figure 4: Distance request sequence

Mobile nodes calculate their location using only one distance measurement for each neighbor. In this case, the distance accuracy is typically 20 cm.

For the location initialization of an infrastructure node, the distance measurement to each neighbor is performed multiple times to get an accuracy of about 1 centimeter. Because of environmental conditions such as noise, a typical 10 measurement sequence between two nodes has a distribution as shown in figure 5. The distance chosen is the center of a 3 centimeter window located where the measurements have the highest density.

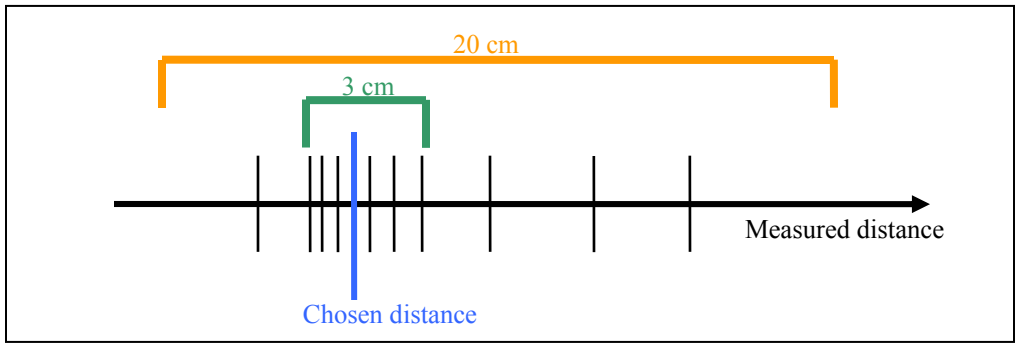

Figure 5: Typical spread of 10 distance measurements between two nodes.

This section has discussed using RF and US measurements to calculate distances between nodes. An emerging technology, called Ultra Wide Band or UWB [3], has the potential to replace the described methods. Designed as a data communication mechanism, UWB has the fortuitous property that distance measurements can be easily obtained as a side effect of communication. Thus UWB could replace both the RF and US components of a SmartLOCUS platform while still providing communication and distances. The bulk of the SmartLOCUS localization algorithms would remain the same. These are described next.

4.2 Node location calculation

Once the distance calculation has been done, node N knows the following information about each of its neighbors.

- Their identity.
- Their position and distance away.
- The distances between neighbors as calculated from their positions.

Figure 6 portrays a node N with two neighbors A and B. Given the positions of nodes A and B together with the distances to them, node N can determine its position as one of two possibilities. Information from a third neighbor is required to disambiguate amongst these choices. Therefore, if all nodes are in one plane, it requires three neighbors to locate node N to a unique position. (Note that the three neighbors cannot be collinear.) Similarly, to locate node N within a three dimensional space requires four neighbors, not all in one plane. For simplicities sake, we discuss the planar environment only. Extension to the three dimensional case is trivial, albeit tedious.

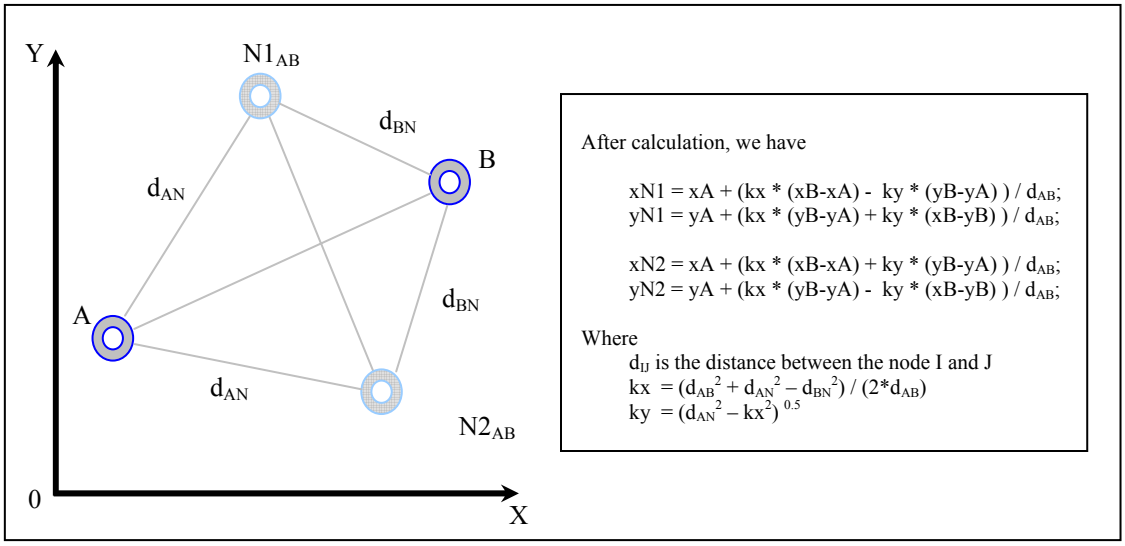

Figure 6: Two possible locations, N1 and N2, for a node given its distances to positioned nodes A and B.

The positioning calculations we use could be easier if we employed the Law of Sines and the Law of Cosines. However, since our hardware platform is simple and low power, the runtime environment does not support trigonometric functions. (In fact, it doesn't even support a floating point data type – we use fixed point.) The available operations are simple arithmetic together with square root. These operations allow us to use the Pythagoras Theorem to calculate positions.

One difficulty arises in position calculation due to the approximate nature of the distance measurements. To address this difficulty, we take advantage of the fact that we need three neighbors to locate a given node. Recall that that in the previous discussion, the third node disambiguated amongst two possible locations. In an information theoretic sense, this is only one bit of information. However, the data supplied by the third neighbor carries significantly more information. To capture the full information available, we use each neighbor node in symmetrical roles. This is explained next.

Consider a node N with three neighbor nodes, A, B and C. There are three possible pairings of neighbor nodes, AC, AB and BC. For each pair, one could calculate two possible positions for node N as was shown in Figure 6. Exactly one of these possibilities would be correct. Given three pairs, there would be six calculated positions for node N, where three of them must be correct, and therefore the same, while the other three of six must be incorrect. To locate node N is simply a matter of choosing the three out of six calculated positions that are the same. However, this assumes that all information is infinitely accurate. In reality, the distance measurements between nodes seldom are. In practice, the three calculated positions which should be the same are merely close.

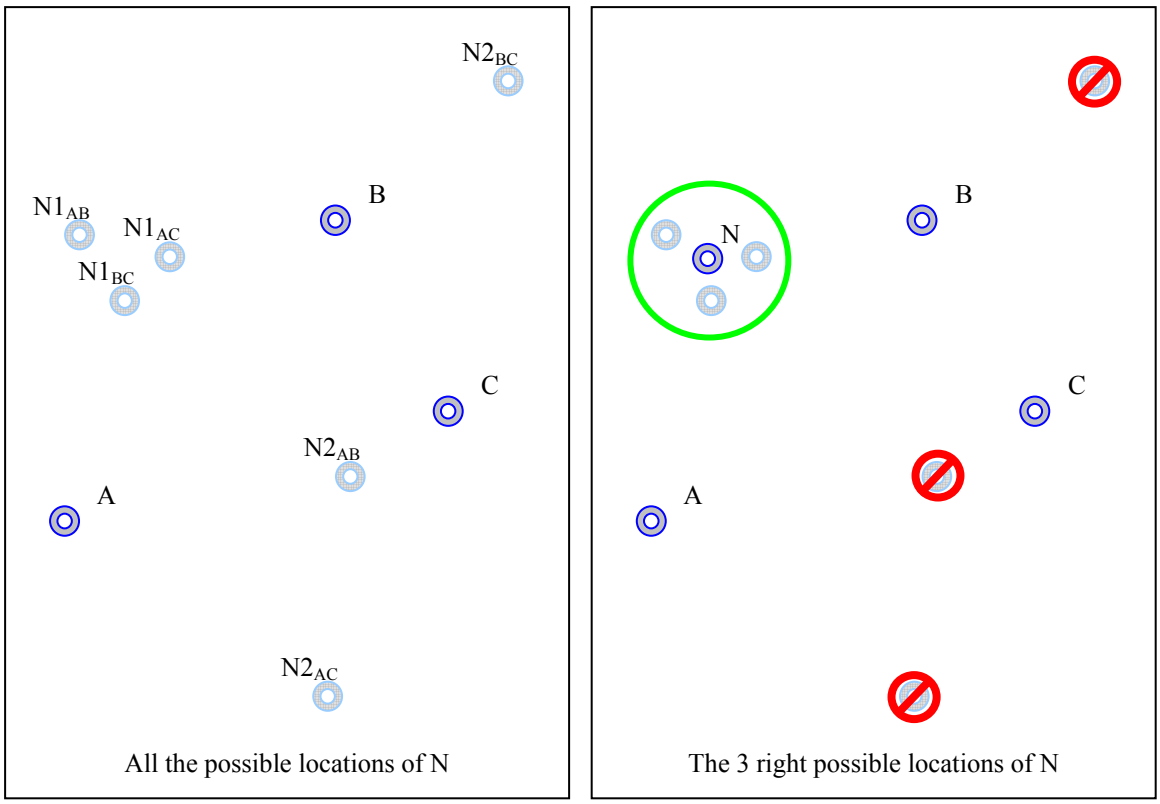

Figure 7: Computation of the position of a node N using the three positioned nodes A, B and C

This situation is depicted in Figure 7. The leftmost diagram shows the three nodes A, B and C. In addition, it shows the six possible locations for node N derived by considering the location of N relative to each of the pairs AB, AC and BC in turn. The rightmost diagram repeats the leftmost and marks the incorrect possibilities. In addition, the rightmost diagram circles the three possible locations for N that, in a perfect world, should be coincident. Even in the face of slight inaccuracies, one can eliminate the incorrect possibilities, since they are so different. We take as the position for node N the average (or center of gravity) of the three nearly coincident possibilities. By using the average, this position incorporates uniformly all of the information contained in the neighbor data.

This algorithm can be extended to the case where there are more than 3 neighbor nodes which have already located themselves. Given k such neighbors, there are (k-1) + (k-2) + . . . + 1 possible pairs of neighbors each generating two possible positions. Because of the combinatorics, considering more than 3 or 4 neighbors is impractical. Later, we discuss a technique to select the "best" subset of available neighbors.

4.3 Infrastructure node location initialization

Several initialization algorithms were tried for this project. The present one provides a simple way to setup any local coordinate system. In the future, it will be possible to translate this local coordinate system within some global system (latitude and longitude for example).

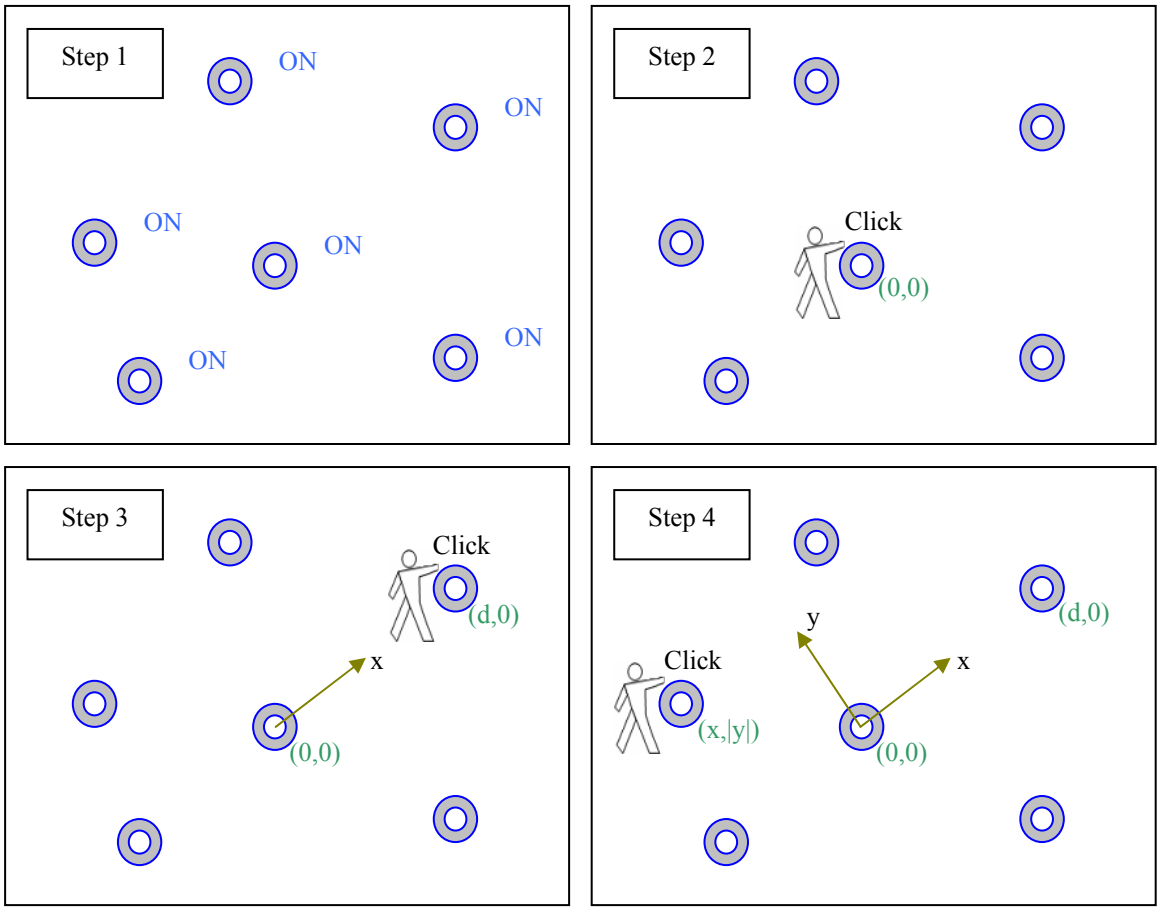

Figure 8: Initialization of infrastructure nodes to create coordinate system

SmartLOCUS platforms that are designed to be infrastructure nodes have a coordinate reset button. The following steps initialize a local coordinate system.

- Step 1: turn on all nodes.
- Step 2: Choose a node to be the origin 0 and push its coordinate reset button.
- Step 3: Choose a node on the positive x axis and push its coordinate reset button.
- Step 4: Choose a node that should have a positive y coordinate and push its coordinate reset button.

All other nodes will determine their own location within the newly created coordinate system (as described previously). These four steps are shown pictorially in figure 8. Note that there is no need to measure or calculate distances. The installer merely presses the same button once on each of the initial nodes.

Figure 9 shows the state diagram used by infrastructure nodes during initialization. When a node is turned on, it simply listens for chatter from its neighbors. State transitions occur as follows:

- If a node hears at least 3 neighbors who know their position, it will calculate its own position and will begin answering position requests from other nodes.
- If the coordinate reset button is depressed, the node will send a distance request. Only nodes knowing their location answer.
- If no neighbor answers, the node sets its position to (0, 0), it is the origin.
- If the only responder is a neighbor at (0, 0) and the response(s) show this neighbor to be D units away, then the node sets its position to (D, 0). This node lies D units along the x axis from the origin.
- If exactly two neighbors respond with their coordinates, the node uses the distances obtained from these responses to calculate its two possible positions as depicted in Figure 6. The one with $y > 0$ is chosen as the location.

The three nodes to have their coordinate reset button pressed establish a coordinate system. All others will calculate their coordinates as explained in paragraph 4.2.

Once initialized, infrastructure nodes recalculate their position every 10 minutes. A history of measured locations is maintained and the location reported to other nodes is adjusted using the history data. In our current implementation, nodes use the average of the last five measured locations if all of them are within a circle of 20 cm radius.

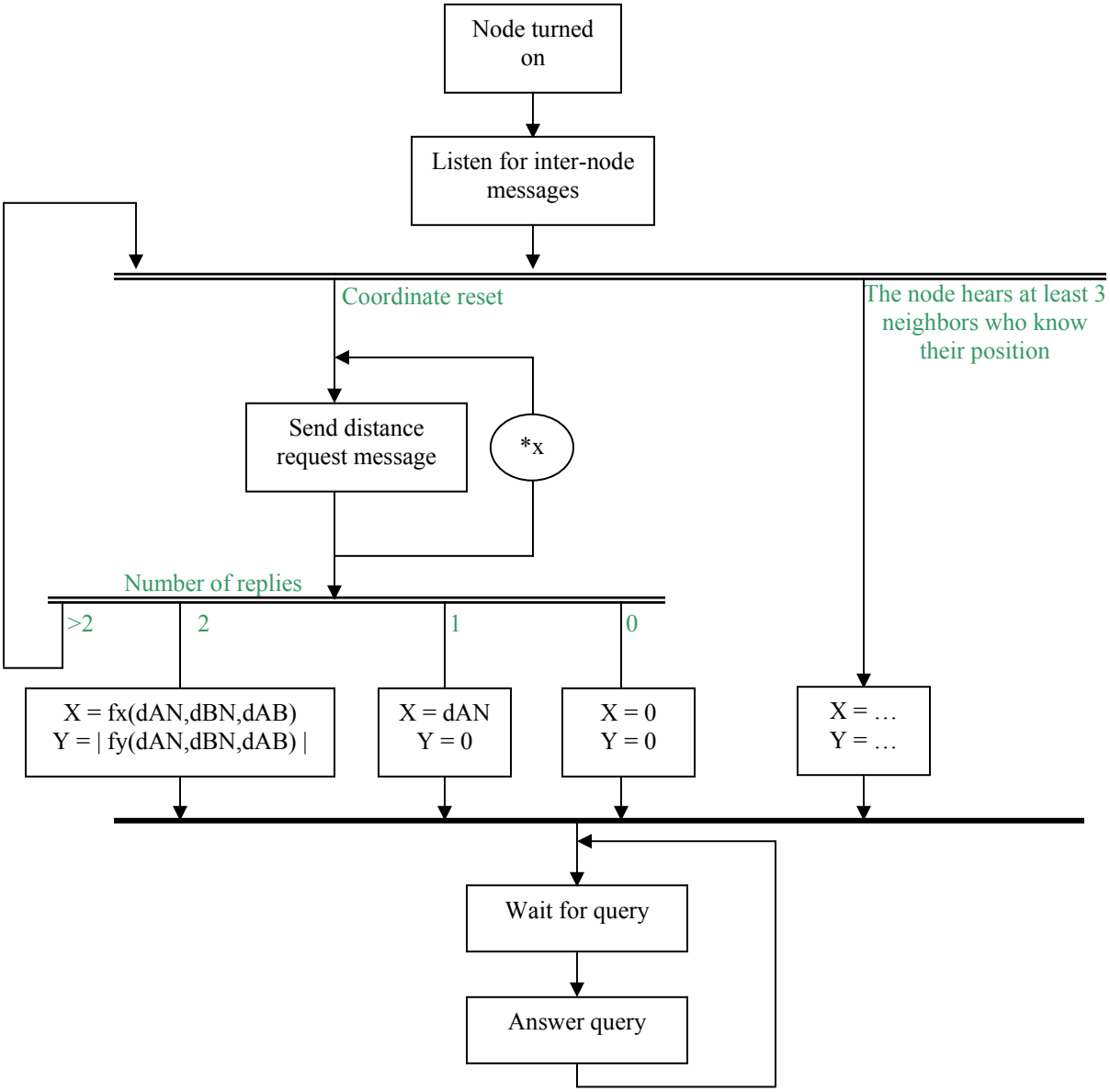

Figure 9: SmartLOCUS infrastructure node initialization diagram

4.4 Neighbor filtering

Using a distance request message, a node can determine its distance to all of its location aware neighbors. In our current implementation, we trim this list of neighbors down to the minimum (k = 3) required to calculate the local position. The reduced list consists of those neighbors whose location information is likely to produce to the most accurate result for the local position calculation. To accomplish this, each node calculates a position accuracy factor, called a. In essence, this factor indicates how far removed a node is from the set of nodes that established the coordinate system. It is an integer

number between 0 and 7. Accuracy factors and geometry are both considered when choosing the reduced list of neighbors. Accuracy factors are described next.

- a = 0 means the node should not help other nodes to initialize. This is the case for a mobile node or for an infrastructure node which does not know its location.

- a = 1 means the node is one of the 3 initial nodes defining the coordinate system. They are the ones with the highest accuracy.

- a = 2 to 7 means that at least one of the neighbors used by the node to initialize itself had an accuracy of a-1. In other words, the accuracy factor is one step worse then the **most** accurate neighbor.

- If a node initializes using a neighbor with an accuracy a = 7, its accuracy will still be 7.

Since errors in distance measurement are independent of the actual distance, longer distances have a smaller percentage error. Also, triangulation with very small angles magnifies errors. With that in mind, we describe the filtering algorithm. Let S be the list of location aware neighbors whose distances are known and which is ordered by decreasing accuracy factors. Then:

- The first chosen neighbor N1 is the one with the best accuracy.

- The second chosen neighbor N2 is the next node in S more than 2 meters from N1.

- The third chosen neighbor N3 is the next node in S more than 2 meters from N1 and N2, and such that the three angles of the triangle N1, N2 and N3 are greater than 25 degrees.

4.5 Mobile node location

In our current implementation, mobile nodes calculate their location once every second. To do so, they broadcast a distance request. The infrastructure nodes answer back once every second for 10 seconds. After, if the mobile node is still in the room, it will broadcast again a distance request.

The mobile nodes do not keep any data history. Every second, they check the distance data they have. If there are at least 3 recent (one second or less) distances recorded, it calculates its location from theses. Otherwise, it broadcasts a distance request message.

5. Conclusion

This paper has presented an overview of the SmartLOCUS system. The system creates an ad-hoc network of nodal communication points that can be attached and attributed to various areas or objects of interest within an indoor space. The key attributes of self-management and location awareness form a powerful base onto which many possible applications can be established.

References

[1] Harter A., Hopper A., Steggles P., Ward A., and Webster P. The Anatomy of a Context-Aware Application. In Proc. 5th ACM MOBICOM Conf. Seattle WA. Aug 1999.

[2] Priyantha, N, Charkraborty, A, Balakrshnan, H. The Cricket Location-Support System. In Proc, 6th ACM MOBICOM Boston MA. Aug. 2000

[3] IEEE 802.15 Low Rate Alternative PHY Task Group (TG4a) http://www.ieee802.org/15/pub/TG4a.html