Proceedings of IMECE 2004
ASME International Mechanical Engineering Congress
November 13-19, 2004, Anaheim, California USA

IMECE2004-60736

A DECENTRALIZED APPROACH TO SOUND SOURCE LOCALIZATION WITH SENSOR NETWORKS

Isaac Amundson **Peter L. Schmidt** **Kenneth D. Frampton**

Department of Mechanical Engineering
Vanderbilt University, Nashville TN 37235

ABSTRACT

A sound source localization system has been developed based on a fully decentralized sensor network. Decentralization permits all nodes in a network to handle their own processing and decision-making, and as a result, reduce network congestion and the need for a centralized processor. The system consists of an array of battery operated COTS Ethernet-ready embedded systems with an attached microphone circuit. The localization solution requires groups of at least four nodes to be active within the array to return an acceptable two-dimensional result. Sensor nodes, positioned randomly over a 10 square meter area, recorded detection times of impulsive sources with microsecond resolution. In order to achieve a scalable system, nodes were organized in groups of from 4 to 10 nodes. Grouping was determined by the selecting the nodes farthest apart from each other. A designated leader of each group analyzed the sound source arrival times and calculated the sound source location based on time-differences of arrival. Experimental results show that this approach to sound source localization can achieve accuracies of about 30 centimeters. Perhaps more importantly though, it is accomplished in a decentralized manner, which can lead to a more flexible, scalable distributed sensor network.

INTRODUCTION

Motivation and Objective

Recent advances in miniaturization, low-cost and low-power consumption in microprocessors has led to a good deal of research in distributed, wireless sensor networks [1, 6]. One of the thrusts in this research is to create large-scale sensor networks capable of monitoring various environmental parameters or activities. Locating acoustic sources with a distributed sensor network is an example of one such application.

The envisioned system would consist of numerous sensor nodes deployed over an area of interest, perhaps by hand or by autonomous vehicle. These nodes would then self-organize, synchronize and then begin locating acoustic targets. Such a system would be invaluable for defense and security applications, natural disaster situations and other environmental monitoring. While the challenges associated with developing the system infrastructure are numerous (e.g. ad-hoc networking, message routing, time synchronization, self-localization, etc. [7]), the challenge also exists to create scalable algorithms for locating acoustic targets. The primary aim of the work reported here is to develop a scalable localization algorithm by making use of recent developments in embedded systems hardware and middleware for distributed systems.
Traditionally, distributed real-time embedded (DRE) systems run under the command of a central controller. This centralized design model can lead to data collection bottlenecks, because often there are insufficient resources to direct all embedded devices in a system. This is especially true for vast arrays of sensor/actuator nodes. Recent developments in embedded processors have enabled new approaches to decentralized systems. Such decentralization permits all nodes in a network to handle their own processing and decision-making, and as a result, reduce network congestion and the need for a centralized processor. For many applications, such as in military and aerospace where size and power consumption may be limited, a decentralized approach may prove more desirable, while still permitting scalability and fault tolerance.

In general, a distributed system is a collection of computers connected through a network that appears as a single coherent system to its users. Distributed systems come in a wide variety of forms. A corporate mainframe/dummy terminal configuration, a cell-phone network, computerized automobile sensors/actuators and the Internet are all examples of distributed systems. Distributed systems can further be divided into centralized and decentralized categories. A centralized system is a distributed system in which a number of lightweight remote subsystems are driven by a master controller. A decentralized system, or truly distributed system, has no master controller, and each subsystem manages a portion of the entire system's workload.

At present, centralized systems are the most abundant form of distributed systems in the field. The benefits of centralized systems are that they are easier to develop and are generally cheaper in cost. For these types of systems, computations are performed by a single controller, allowing for a much simpler overall design. The individual nodes are also fairly basic, often consisting of merely a sensor or actuator with minimal to no computing power at all. This can lead to reduced node cost and development time.

The major drawback to centralized systems is that they are not scalable. A controller on a distributed centralized system with ten thousand nodes, for example, will perform less efficiently than a controller overseeing only ten nodes. Because scalability is becoming increasingly important in sensor networks as their applicable use in the field grows, a feasible alternative is needed to overcome this processing bottleneck. Another drawback to the centralized design is that if the centralized controller drops off the network for some reason, the distributed nodes will be useless. Not only will they be unable to sense or direct their surrounding environment, but their controller will also be unable to collect and archive events that are occurring during that downtime for later analysis.

Decentralized, or truly distributed systems, are gaining popularity because of their scalability and fault tolerance properties, in addition to reduced cost, and wider availability of both hardware and software COTS products. Rather than relying on a single controller to assume the workload of the entire system, it is becoming easier to implement a decentralized design by employing nodes that are able to do a share of the processing. As embedded technology advances, we are seeing hand-held devices that have the processing capabilities of full-scale computers from only a few years ago. Furthermore, we are seeing the development of inexpensive embedded real-time operating systems and frameworks that are able to drive these distributed nodes.

Decentralized system design can lead to more accidental complexity due to concurrency issues inherent with several distributed nodes attempting to work in concert. This can further lead to increased design time for applications that perform more involved operations. Most of this accidental complexity can be overcome with development tools, pattern usage, and standardized frameworks, but in general a decentralized system will be more intricate than a centralized one.

Related Work

Recently, several research projects have been undertaken by diverse groups interested in using sensors to gather video, acoustic, olfactory, and radio frequency data from a given target. Hayes et al. [5] have developed a scheme for using a group of robots to cooperatively locate the source of a chemical plume. Navarro-Serment et al. [13] have developed a localization algorithm for teams of robots using ultrasonic beacons to communicate with and track their team members.

Hawkes and Nehorai [4] have developed an algorithm for source localization using acoustic vector sensors. While making use of intensity data significantly alters the accuracy achievable, the sensors are not of a mature design and the data reduction is limited to a centralized computing station. Kozick and Sadler [9] have also approached localization in a distributed fashion. By treating small groups of sensors as individual arrays, data reduction computing cost is reduced, along with required communication bandwidth. Their solution method is also intended to run on a centralized computing platform.

Chen et al. [2] have developed an acoustic source localization system using a beam-forming approach to determine source azimuth and distance. Their devices are low cost Compaq iPAQs, connected via a wireless network, but still rely on centralized data reduction with no real time capability.

Yao et al. [18] investigated the use of various beamforming techniques with a randomly distributed sensor array. The results were based on theoretical studies, but one special case considered was the use of Time Difference of Arrival (TDOA) solutions to source localization (identical to those used here). Yao et al. noted that the least squares solution to the TDOA equations is very sensitive and that an over-determined set of equations is desirable. However, the results presented were for far-field sources (i.e. sources outside of the array) while the current work focuses on near-field sources.

Scope and Goals of this Work

While all of the approaches to source localization discussed previously were reliable and accurate, they are not well suited to large-scale sensor networks because they are based on centralized computational schemes. In order to develop a more scalable approach the current work focuses on *decentralized* sound source localization. This approach will consist of several sensor nodes organized into groups. Each node in the network is the leader of a group of from 4 to 10 nodes. The leader of each group collects the times of arrival of all group members and independently estimates the source location base on TDOA equations. Furthermore, in order to perform this task several middleware services must be employed such as time synchronization, group management, and ad-hoc network communication. Each of these subject areas is discussed in the following sections along with a description of the experimental hardware and results.

SOUND SOURCE LOCALIZATION

To calculate a sound source location in two dimensions, a localization equation obtained from Mahajan and Walworth [10] was used. This approach requires that a minimum of four sensor nodes detect the acoustic source time of arrival. The sound source, at an unknown position (u, v), and receivers, at known positions, are assumed to lie on the same plane. If each sensor records the time of arrival of the acoustic wave (t_1, t_2, t_3, and t_4) then the time-differences-of-arrival (TDOAs) can be calculated as follows:

$$\begin{aligned} \Delta t_{12} &= t_2 - t_1 \\ \Delta t_{13} &= t_3 - t_1 \\ \Delta t_{14} &= t_4 - t_1 \end{aligned} \quad (1)$$

The above TDOAs only take into account differences between the first node to detect the sound, n_1, and all other nodes. If d is the distance between the source and n_1, then the distance between the source and the second node to detect the sound, n_2, must be $d + c\Delta t_{12}$, where c is the speed of sound. Similarly, the distances of nodes n_3 and n_4 from the source must be $d + c\Delta t_{13}$ and $d + c\Delta t_{14}$, respectively. Based on this, the distances from each node to the source are

$$\begin{aligned} (x_1 - u)^2 + (y_1 - v)^2 &= d^2 \\ (x_2 - u)^2 + (y_2 - v)^2 &= (d + c\Delta t_{12})^2 \\ (x_3 - u)^2 + (y_3 - v)^2 &= (d + c\Delta t_{13})^2 \\ (x_4 - u)^2 + (y_4 - v)^2 &= (d + c\Delta t_{14})^2 \end{aligned} \quad (2)$$

where x_i and y_i are the known coordinates of the i^{th} node. Expanding the first equation in Eq. 2 results in

$$d^2 = x_1^2 - 2x_1 u + u^2 + y_1^2 - 2y_1 v + v^2 \quad (3)$$

Now, substituting d^2 from the above equation into the remaining three equations, results in a set of dependent equations with as follows:

$$\begin{bmatrix} c^2(t_2 - t_1)^2 + x_1^2 + y_1^2 - x_2^2 - y_2^2 \\ c^2(t_3 - t_1)^2 + x_1^2 + y_1^2 - x_3^2 - y_3^2 \\ c^2(t_4 - t_1)^2 + x_1^2 + y_1^2 - x_4^2 - y_4^2 \end{bmatrix} = \begin{bmatrix} 2x_1 - 2x_2 & 2y_1 - 2y_2 & -2c(t_2 - t_1) \\ 2x_1 - 2x_3 & 2y_1 - 2y_3 & -2c(t_3 - t_1) \\ 2x_1 - 2x_4 & 2y_1 - 2y_4 & -2c(t_4 - t_1) \end{bmatrix} \begin{bmatrix} u \\ v \\ d \end{bmatrix} \quad (4)$$

The above equation can be solved with traditional linear algebra approaches. However, as noted by Yao et al., the solution to this equation can easily be ill conditioned. Therefore, some care must be taken when considering solutions of this kind. Note that with additional TOA measurements, more equations could be added to Eq. 4 resulting in an over determined system. The solution approach taken here will be discussed in detail in later sections.

EXPERIMENTAL PLATFORM

Hardware

The increasing feasibility of embedded systems research is due in part to the wide variety of micro-computational devices available. Sensor network development is now possible using basic hand-held personal desktop assistants, PC/104 devices, and even the more specialized Berkeley Motes [6]. A PC/104 device was selected for this application due to its reduced cost, modular design and ease in programming.

The PC/104 modules used were the Diamond Systems' Prometheus, which includes A/D and D/A data acquisition, a ZF Micro Devices ZFx86 100 MHz CPU, a 100 Mbps 10/100BaseT Fast Ethernet port, 32 MB RAM and 128 MB flash disk storage.

The nodes in the sensor network communicate via Ethernet and are connected through a 3Com router. This may be improved on in the future by either using wireless networking or radio frequency transmission. Because real-time communication was not a requirement for this application, there was no immediate need to implement these methods at development time.

A microphone circuit for sound detection was constructed using a Panasonic-ECG WM-34BY omni-directional microphone, an amplifier of gain 10, and an 8th order Butterworth 10 kHz low pass filter. The circuit was connected to one of the Prometheus' external triggers. The signal from the microphone circuit was passed to a comparator. When the amplitude exceeded a set threshold (100 mV in this case) a 5-volt signal was sent to the interrupt. This triggered an interrupt subroutine within the software, which recorded the sound source time of arrival and began analysis of the sound.

Software

The software that drives the system was written in C++ and compiled with the GCC version 3.2 compiler. The nodes in the sensor network run under an embedded version of Linux, called Flash Linux. The Adaptive Communication Environment (ACE) [15,16] was used to develop the sound source localization application. ACE is an open source framework developed for high-performance and real-time communication services and applications. Network communication was achieved using The ACE ORB (TAO) [17]. TAO is an open source extension to ACE and the Object Management Group's CORBA [14], which arranges client/server communication in an object-oriented manner. The benefits of using ACE and TAO far outweigh the minimal losses in performance by establishing a pattern-oriented structure of program design, while ensuring scalability, robustness and portability.

Time Synchronization

Time synchronization is critical for applications that use time-of-arrival (TOA) measurements. This is because the TOA measurements are made with respect to each node's local clock. If individual nodes are not in agreement as to the global network time, then the resulting errors in the arrival time measurements will directly affect the accuracy of the source location solution. Therefore, in order to ensure accurate time synchronization among nodes the Reference Broadcast Synchronization (RBS) [3,8] approach was selected over the more common Network Time Protocol (NTP) [12].

RBS eliminates most of the non-determinism introduced by other time synchronization methods by only examining the arrival time of a general broadcast packet with a user-level packet capture technique [11]. In fact, any node in a distributed system can send a multicast message over the network. All nodes on the network will receive the message at roughly the same time (packet travel over the network is basically deterministic; generally, packets travel at one foot per nanosecond). By comparing these arrival times, each node can build a registry of all arrival times across the network, and use this registry when calculating reference times.

Another benefit of using RBS is that time synchronization can take place at any time during program execution. If synchronization happens at program initialization, clock skew between nodes will occur over time, resulting in a phase offset. However, if clock synchronization is not necessary until a specific even happens, RBS can be used *after* the event, allowing for a more accurate event time value.

Both NTP and RBS protocols were implemented on the experimental platform. Using NTP, synchronization errors in the range of 5 and 8 milliseconds were recorded. Using RBS, this error was reduced to an average of 5.75 *micro*seconds, and in no instance did this error reach higher than 25 microseconds.

Grouping

To achieve a scalable decentralized design, it was necessary to divide the nodes into groups. If each node in the system were to receive the TOAs from every other system node then the network would be overwhelmed as the number of nodes was increased. The means for creating a scalable approach was to limit the number of TOAs used by each node. Therefore, each node in the system was designated as a group leader that would select a pre-determined number (between 4 and 10 in this study) of nodes to be part of its group. Each node was also a group member in at least one other group. Note that a group containing the total number of nodes in the sensor network is essentially a centralized system, because each node would be performing the work required of the entire system.

For these experiments, grouping was achieved by using the most basic algorithm, whereby each node chooses the $n - 1$ nodes furthest from it, where n is the designated size of the group. However, this technique can skew the results if the selected nodes are either in close proximity to one another, or if more than one happens to fall along a straight line (which results in Eq. 4 being ill conditioned). The approach used here, which resulted in a more robust grouping algorithm and solution to Eq. 4, created groups of nodes that were furthest apart from each other. This resulted in each group covering the maximum area, and having maximum inter-node spacing. This resulted in a reduced sensitivity to TOA measurement error. However, this group spacing was not a critical factor in obtaining satisfactory results.

Sound Source Localization

As described earlier, when a sound source caused the microphone circuit to exceed a preset threshold an interrupt was triggered which caused a subroutine to record the time. Each node that detected the source then sent it's detection time to its group leader. When each group leader had received detect times from all of its group members, or a preset time limit was exceeded, the TDOAs were analyzed to calculate the sound source location.

The TDOAs were used in an exactly determined set of equations as in Eq. 4. Four TOAs were required in order to solve the equations. When the group size was 4 then each group leader would receive just enough TOA measurements for a solution. However, as group size increased, more TOA measurements were available than needed. One obvious approach, and that suggested by Yao et al. [18], was to create an over-determined set of equations similar to Eq. 4. However, it was found that this approach did not yield the best possible solution. The least squares solution to the over-determined equations tended to be sensitive to individual errors. For example, if one node recorded a false or erroneous TOA, it would cause the least squares solution to be skewed.

An alternative approach was to take every combination of the nodes reporting TOAs 4 at a time. As the group size increased, the number of possible combinations of four nodes also increased. It was found through modeling and evaluation of the experimental data, that the best solution was achieved by calculating all possible solutions of n nodes, taken 4 at a time, and then taking the median of that group of solutions. An example of the distribution of solutions is shown in Figure 1. This is for the case when the group size is 6 resulting in 76 unique solutions overall. Note that the average of all of these solutions resulted in an estimated x-position of about 3.42 meters, while the median of all estimated positions was 4.61 meters. The actual x-position of the source in this case was 5.00 meters. Also note that 3 of the solutions estimated the location to be greater than 5 meters from the actual source. These three erroneous solutions are a result of one node with a poor recording of the TOA. The fact that the overall location error remains low is indication of the robustness of the system.

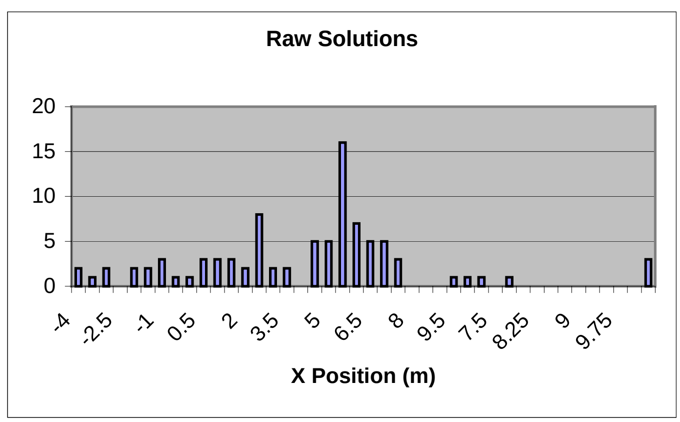

Figure 1. Histogram of x-position solutions calculated by group leaders.

Experimental Set-up

A series of experiments were conducted by randomly positioning 10 nodes over a 10 by 7 meter indoor area. Eight sound source positions were also chosen at random, both within the perimeter of the sensor network as well as outside of it. The positions of the nodes and sound sources can be seen in Figure 2. A starter pistol was used as the sound source, as it provided a loud impulsive noise, likely to carry to all the microphones in the sensor network.

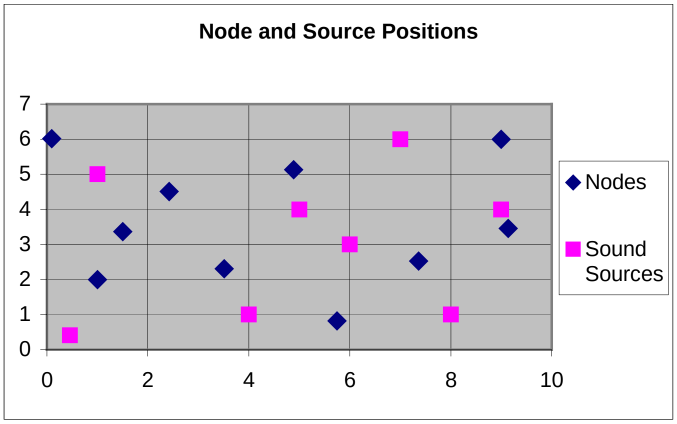

Figure 2. Positions of nodes and sources.

To begin a set of experiments the nodes were powered up and then the ad-hoc networking, time synchronization and group-forming protocols were run. This initialization took about 60 seconds. The starter pistol was then fired at a predetermined source location. Once each node estimated the position of the source, all data from that node (including all candidate solutions, times of arrival, etc.) were transmitted to a laptop that was also connected to the network.

EXPERIMENTAL RESULTS

Experiments were carried out for a 10-node network and with group sizes of 4, 6, 8 and 10. Each node recorded a source location estimate as described earlier. A summary of the error for all sources inside the sensor field perimeter and all node solutions is shown in Figure 3 (the bracketed bars indicate the standard deviation of all solutions). Note that 4 groupings times 8 sources times 10 nodes results in 320 total data points (40 per group point) went into the construction of Figure 3. As expected, the error (defined as the distance from the actual source to the estimated source) decreased with an increase of group size. The minimum average error was 36 centimeters and occurred for a group size of 6. However, no significant gain in accuracy was achieved after increasing the group size greater than 6. Hence, these results imply that no benefit will be gained by increasing the group size beyond 6. It should be noted, however, that in these well-controlled experiments false triggering of individual nodes was a rare event. In a real application, false triggers would be much more common and, therefore, larger group sizes would likely yield a significant benefit. It is interesting to note that Yao et al. found a similar limit on over-determined solutions [18]. These results are slightly worse than the approximately 10-centimeter error achieved by Chen et al. [2]. However Chen et al. used a centralized optimization solver as well as a matched filter TOA detector.

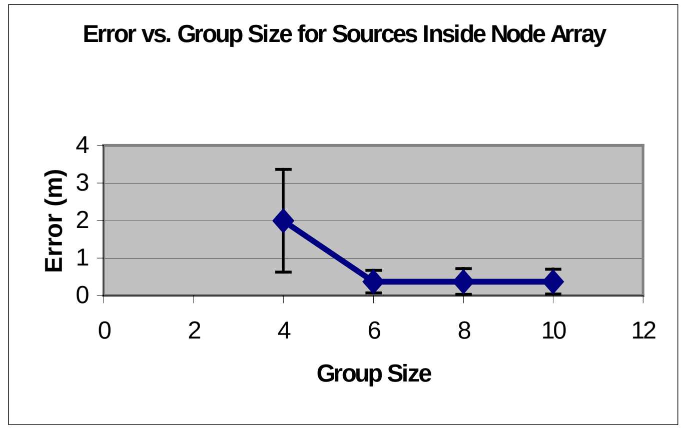

Figure 3. Localization error in meters, based on node group size for sound sources within the node array

The performance of this solution approach was not nearly as accurate when the sound sources were located outside of the array. This is demonstrated in Figure 4, which shows average error of all solutions for all sources outside the array. While the TDOA solution was not very good at locating sources outside the array, it was interestingly good a indicating the direction from which the source originated. This was indicated by the solution to sources outside the array being estimated to be on the edge of the array nearest the source. Yao noted a similar weakness in this approach and only attempted to solve for bearing and not location.

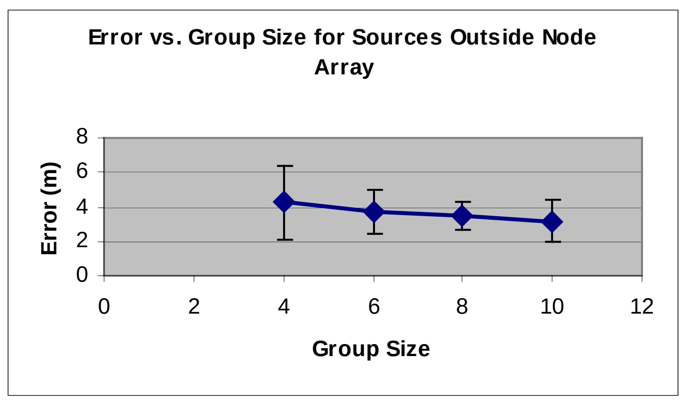

Figure 4. Localization error in meters, based on node group size for sound sources outside of the node array.

As noted previously, the best solution was achieved by a group of six. In this case, the average error was about 30 centimeters. This error can be attributed to several factors including time synchronization offset, operating system jitter in recording the detect time, and sound source positioning error. Time synchronization was found to be accurate to within 25 microseconds and therefore contributes very little to the overall error. However operating system jitter was a significant source of error. In separate tests, the average time delay from the moment a sound reached a node to the time the node actually recorded the detection time, was found to be around 400 microseconds. The maximum delay noted was 520 microseconds. Note that an error of 500 microseconds translates to about 0.17 meters of error for a speed of sound of 343 m/s. This latency was generally unavoidable because of the use of a non-real-time operating system and program multithreading time considerations. It could be improved by using a real-time operating system.

The theoretical TDOAs were calculated for each source and each node. These were compared with the actual measured TDOAs and are summarized in the histogram of Figure 5. Note that the majority of solutions returned an error of less than ten milliseconds. Of about 400 samples, slightly less than 150 reported arrival time differences greater than that. These errors are not particularly good and are mostly the result of using a threshold detection mechanism. Threshold detection devices are known to be rather sensitive to such effects as reflections (experiments were performed in a hard walled room), propagation attenuation, and other acoustic effects. Several methods of improving TOA measurements, and hence TDOAs, are available including the matched filter approach used by Chen et al. [2]. However, despite these relatively poor TDOA measurements, a reasonably good estimate of the source location was achieved.

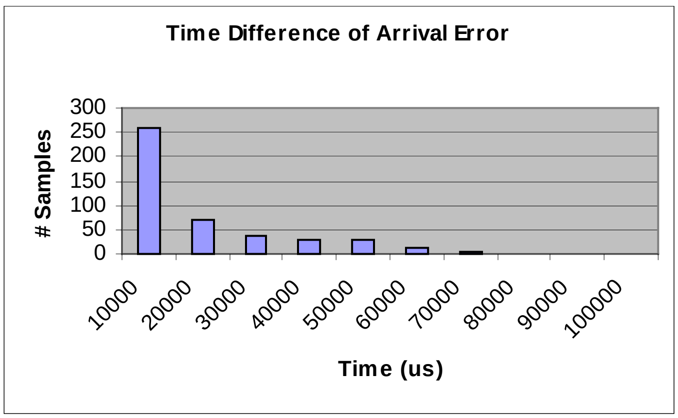

Figure 5. Breakdown of solution error. The majority of solutions realized an error of less than ten milliseconds.

CONCLUSIONS AND FUTURE WORK

An experimental implementation of a decentralized sensor network based acoustic localization system has been described. The system was developed from several COTS PC/104 embedded processors and software was developed to synchronize the processor clocks, provide for network routing, manage groups within the network and estimate the position of acoustic sources via a TDOA solution. Results from these experiments demonstrated an accuracy of 30 centimeters when the group size was six or more. This accuracy was found to be comparable to previous centralized acoustic localization systems.

Although the use of a decentralized distributed system in sound source localization was successful, this application can still be improved upon by varying degrees. We have demonstrated that a truly distributed system is more than capable of handling the workload, and that several middleweight nodes are able to perform the task of a centralized controller. Note that had the number of nodes in our system been increased ten- or one hundred-fold, for example, computation time would have remained the same, whereas in a centralized system this time complexity would have rapidly increased.

This system was designed to localize sound sources in open spaces where line-of-sight is unobstructed. When obstacles such as walls or other large structures are introduced, sound wave reflections and multi-path effects occur, distorting the arrival time of the sound. This can be controlled however, as the amplitude of a reflected sound wave will be reduced, depending on the impedance of the reflecting material.

At present, the sensor network is able to localize a stationary discrete sound source. Future modifications will allow the tracking of sound sources in motion. This will be beneficial because it might not always be desirable to know the location of the start position of a source, but highly desirable to know its current position.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation CAREER program and the Information Technology Research program.

The following people were essential in the design and development of this project: Douglas Schmidt, Aniruddha Gokhale, Balachandran Natarajan, Tao Tao, and Steve Williams.

REFERENCES

[1] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E. Wireless Sensor Networks: A Survey. *Computer Networks*, 38(4):393–422, March 2002.

[2] Chen, J.C., Yip, L., Elson, J., Wang, H., Maniezzo, D., Hudson, R.E., Yao, K., Estrin, D. Coherent Acoustic Array Processing and Localization on Wireless Sensor Networks. *Proceedings of the IEEE, Vol. 91, No. 8, April 2003*

[3] Elson, J., Girod, L., Estrin, D. Fine-Grained Network Time Synchronization Using Reference Broadcasts. In *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation* (OSDI 2002), Boston MA, December 2002.

[4] Hawkes, M., Nehorai, A. Wideband Source Localization Using a Distributed Acoustic Vector-Sensing Array. *IEEE Transactions on Signal Processing, Vol. 51, No. 6, June 2003*

[5] Hayes, A.T., Martinoli, A., Goodman, R.M. Distributed Odor Source Localization. *IEEE Sensors*, June 2002

[6] Hill, J., Culler, D.: Mica: A Wireless Platform for Deeply Embedded Networks. *IEEE Micro*, Vol. 22, No. 6, 2002, pp. 12–24.

[7] Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., Pister, K. System architecture directions for networked sensors. *Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and*

Operating Systems (ASPLOS-IX), pages 93–104, Cambridge, MA, USA, November 2000. ACM.

[8] Karp, R., Elson, J., Estrin, D., Shenker, S. Optimal and Global Time Synchronization in Sensornets. *Technical Report CENS Technical Report 0012*, Center for Embedded Networked Sensing, University of California, Los Angeles, April 2003

[9] Kozick, R.J., Sadler, B.M. Source Localization with Distributed Sensor Arrays and Partial Spatial Coherence. *IEEE Transactions on Signal Processing, May 14, 2003*

[10] Mahajan, A., Walworth, M. 3-D Position Sensing Using the Differences in Time-of-Flights from a Wave Source to Various Receivers. *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 1, February 2001.

[11] McCanne, S., Jacobson, V. The BSD Packet Filter: A new Architecture for User-level Packet Capture. Presented at *1993 Winter USENIX conference*, January 25-29, 1993, San Diego, CA

[12] Mills, D.L. Internet Time Synchronization: the Network Time Protocol. In Zhonghua Yang and T. Anthony Marsland, editors, *Global States and Time in Distributed Systems*. IEEE Computer Society Press, 1994.

[13] Navarro-Serment, L.E., Paredis, C.J.J., Khosla, P.K. A Beacon System for the Localization of Distributed Robotic Teams. *Proceedings of the International Conference on Field and Service Robotics (FSR '99)*, August, 1999

[14] Object Management Group, *The Common Object Request Broker Architecture: Core Specification*, 3.0.2 ed., December 2002

[15] Schmidt, D. C., Huston, S. D. C++ Network Programming, Volume 1: Mastering Complexity with ACE and Patterns. Addison-Wesley, Boston MA, 2002

[16] Schmidt, D. C., Huston, S. D. C++ Network Programming, Volume 2: Systematic Reuse with ACE and Frameworks. Addison-Wesley, Reading MA, 2002

[17] Schmidt, D. C., Levine, D. L., Mungee, S. The Design and Performance of Real-Time Object Request Brokers. *Computer Communications*, vol. 21 pp. 294-324, April 1998

[18] Yao, K., Hudson, R.E., Reed, C.W., Chen, D., Lorenzelli, F. Blind Beamforming on a Randomly Distributed Sensor Array System. *IEEE Journal on Selected Areas in Communications* Vol. 16, No. 8, pp. 1555 – 1567.

