An Iterative Approach to Locating Simple Devices in an ad-hoc Network

David P. Robinson, Ian W. Marshall.
BTexact Technologies

Abstract

This paper considers the problems associated with location determination in wireless ad-hoc networks. It presents a simple algorithm allowing a distributed set of devices to independently determine their location given only the distances between themselves and neighbouring devices if some devices in the network (location aware devices) have external location information (such as GPS or being at a known fixed location). The algorithm allows each device to make a guess of location that it then communicates to all neighbouring devices. Depending on the information received from neighbours each device may then modify its guess. In this way, over time, the network as a whole is able to identify a set of positions that satisfies all the available distance data. The algorithm is shown to be resistant to errors in measuring the distances between devices.

1 Introduction

Wireless ad-hoc networks are self-organising, rapidly deployable and require no fixed infrastructure, comprising of wireless devices [1]. One simple application of such networks is in low-cost, low-maintenance sensor networks. A great deal of work has been carried out recently in investigating such networks, in particular in the area of routing protocols ([2]). Routing protocols are almost universally superior if information about a device's location is available to it. The "Terminodes" project [3] has investigated many problems inherent with wireless, ad-hoc networks extending the work on "Active Badge" [4], [5] and "Smart Badge" [6] into new areas.

Accurate determination of the location of devices in such networks has been historically largely ignored, with location either being assumed to be known by use of a global positioning system (GPS) receiver (e.g. [7]), or by requiring only low accuracy of location [4]. Traditional trilateration or triangulation approaches to accurate location determination rely on the device being located having a line of sight to one or more base stations with known location (e.g. [8]). Recently there has been some work into location determination, stimulated by findings presented by Capkun et al. [9]. This work has highlighted the fact that it is possible for devices in a network to determine their own location by measuring only the distance to nearby devices. In all likelihood, there are several methods by which devices are able to self-determine their location. Two approaches to the problem have been investigated. The method to be presented in this paper is markedly different to that of Capkun et al. [9] in that it is an iterative approach whereby each device guesses a position, informs its neighbours and then re-guesses position. The second method allows ambiguities present in the solutions of simultaneous equations to propagate through the network and will be detailed elsewhere.

2 Method

This paper presents the results from a series of simulations. In all cases, an ad-hoc network of devices was constructed. All devices within the network are considered relatively simple, having limited processor power and memory. Each device is also assumed able to communicate with other nearby devices within a certain range of communication. It was also assumed that all devices were able to determine the distance between themselves and neighbouring devices with which they can communicate. Within each network simulated, there are a small proportion of devices that are able to locate themselves absolutely at all times. These represent a sub-group of devices with some extra capability such as GPS. All other devices in the network must locate themselves relative to these devices. The goal of this research was to allow a large portion of the network to locate themselves by transmitting limited amounts of data and by carrying out only simple calculations. If it were possible for each device in the network to guess its location and then transmit its guess of location to its neighbours who then improve their guess of location before repeating the process, then the location guesses of the network as a whole will eventually converge to the real locations of each individual device. The problem to be solved is how do individual devices determine how to improve their guess of location when they have only information from their immediate neighbours available to them.

Imagine a set of beads connected to each other on a tabletop by springs of various lengths. Several of the beads are fixed in certain locations on the table, the rest are free to oscillate. Let us assume that there is some equilibrium set of positions where the beads are in such a position so that every spring is its natural length. If the beads are displaced from this equilibrium, they will oscillate as their connecting springs are stretched or compressed, but after some time will return to their equilibrium positions. Each bead is only influenced by its own spring connections and hence only by their immediate neighbours. By likening an ad-hoc network of devices to this mechanical system, it is possible to create a method of device location that would be applicable to relatively low-cost networks. All each individual device needs is some form of apparatus to determine the range to neighbouring devices and some means of communicating to those devices.

At each iteration, all devices send a brief communication to their neighbours. That communication contains information about where that node thinks it is located and whether it has made that decision by successive guesses of location or as a result of being informed of its position by an external source (such as GPS). From the information each device receives, it can calculate what the distance between itself and each of its neighbouring devices would be if all the guesses of location were correct. This 'virtual distance' can then be compared to the real measured distance between devices. In this way, a terrain of the 'perceived error' for each device can be constructed by the device itself and the next guess of location for that device can be at a lower perceived error. Changing the guess of location of a device will influence the terrain of perceived error in all neighbouring devices. The implication of this is that the latest guesses of location that each device has made will now not necessarily be in the region of low perceived error observed in the previous set of location guesses. Thus, each device will have to make further guesses of location.

There are several potential problems. Devices may get trapped in a set of position guesses that are not correct but represent an area of reasonably low error or devices may chose positions in successive iterations that result in an oscillatory behavior between nodes developing. To overcome these problems there must be some probabilistic method of 'resetting' the device's position guess away from its current guess. As well as moving a guess away from a local minimum, this has the effect of breaking any developing oscillatory motion, solving both problems. This reset clause must only be activated when a superior solution is not being obtained, but should be activated as soon as it is clear this condition is met. The exact question of when a reset clause should be activated was of great difficulty but a set of conditions that work for the networks simulated was found.

3 The simulations

A decision was made at the outset that the simulations produced would be as close to a real-life embodiment of the technology as possible. It is envisaged that the methods outlined here would be used in a rapidly deployable, low-cost, self-organising sensor network. Such a system could be deployed in areas that are largely unmonitored due to their inaccessibility, or harshness of environment. Similarly due to the ad-hoc nature of the network, a network could equally be deployed as a low-cost alternative to current centrally maintained sensor networks such as those seen in some office environments (e.g. 'active bat' [10]). It was decided that all simulations should run on a standard square grid with side of length 400. The grid was populated with two types of device, one that knew its location at all times and another that had to determine their location. All devices have a range of communication of 50 and the devices that have to guess their location are distributed randomly onto the grid. Several questions needed to be answered in order to gain an understanding of what the important factors for allowing a device to locate themselves accurately were. An in-depth discussion of the algorithm is too long for this publication, so the results section deals with the questions posed when considering the ability of the algorithm to perform under different conditions.

4 Results

This section presents results to many of the experiments carried out. The performance of the algorithm is assessed on two counts, the quality of the solution obtained and the speed with which that solution is obtained. A satisfactory solution is defined as being a solution where all devices have a perceived error that is smaller than some low threshold value. The quality of the solution obtained is defined as being the percentage of devices that have located themselves in the correct position when this satisfactory solution is reached. The algorithm used in all cases remains constant. The algorithm allows every device in the network to locate themselves relative to the devices with external positional information in a certain number of iterations. In all cases due to the random nature of the deployment of devices, a small percentage may be unable to make a guess of location due to a lack of access to external positional information. In a small number of simulations, the algorithm fails to produce any stable, satisfactory solution within a sufficiently low number of iterations. From observations of how many iterations a simulation takes to find a satisfactory solution, (see figure 1) it seems that the probability of a solution being found decreases with length of time that the algorithm has been running. The same figure seems to suggest that, no matter how long the algorithm is run for, it is possible that no solution is found. It is thought that this is probably due to the initial guesses of location that the system makes which are made randomly, and so if the algorithm has failed to find a solution after a certain time, it may be beneficial to re-start it.

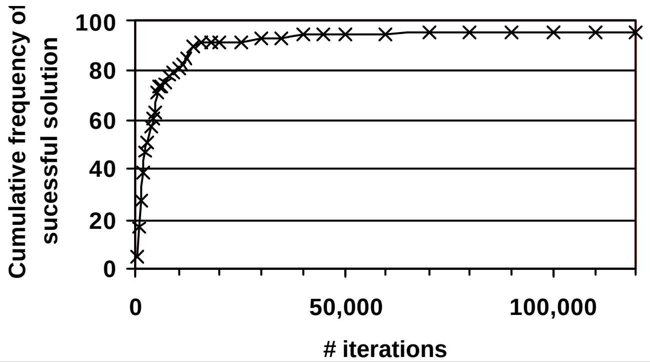

Figure 1 - Curves of percentage probability of a satisfactory solution being found within a certain number of iterations. Results were obtained by carrying out 68 simulations and noting how quickly a satisfactory solution was obtained.

4.1 Distribution of position-aware devices

Several different arrangements of position-aware devices were tried. All arrangements were of a number of devices placed in such a way so as no device in the network as a whole was within communication range of more than one device that had external information of its position. This led to two generic types of arrangements that were tried, namely square packing and the higher density of hexagonal packing. In both cases, several precise arrangements were trialed each providing a different number of position aware devices in the network. Some results from these simulations are shown in figure 2. The results suggest that, while there is an advantage in increasing the number of position-aware devices in the network, the exact distribution of those devices is less important.

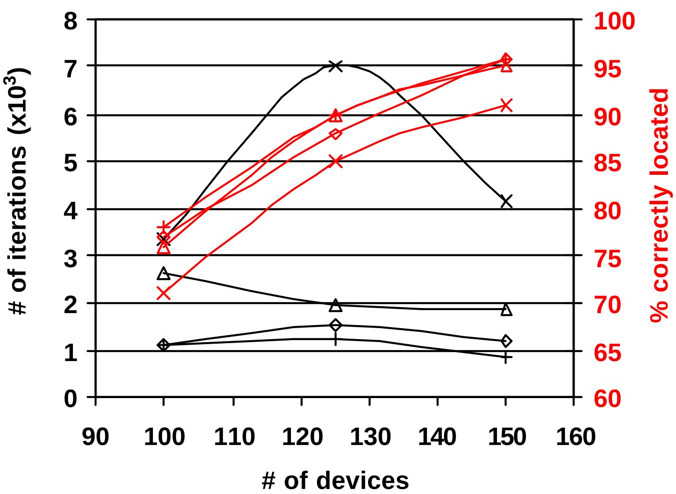

Figure 2 - Graph of average number of iterations taken for a network to find a satisfactory solution (left axis, black) and the average quality of the solution found (right axis, red) against number of devices in the grid that have no external positional information for grids with 16 (crosses) or 25 (plusses) square-packed, 22 (triangles) or 37 (diamonds) hexagonally packed position-aware devices. Results were obtained from 20 successful simulations at each data point. Simulations that had failed to find a suitable solution after 20,000 iterations were discarded.

4.2 Errors on Distance determination

It is likely that in any real world distribution of such devices, the devices will not be able to measure distances between each other with 100% accuracy. To simulate this, an artificial error was introduced to the ranging capability of devices producing an error in the distance measured of up to 5% and 10%. Results obtained shown in figure 3 indicate that although the exact determination of position for individual devices is less certain with higher ranging errors, the algorithm is largely resistant to errors in range determination.

5 Conclusions

The algorithm presented in this paper provides a new alternative method for determination of location of devices within a network. By iteratively improving on a solution the system as a whole is able to settle on a solution that satisfies all the data that is available to it. The network is able to do this without the need for central information to be stored about the system; each device needs only local knowledge and small processing capabilities. This has the

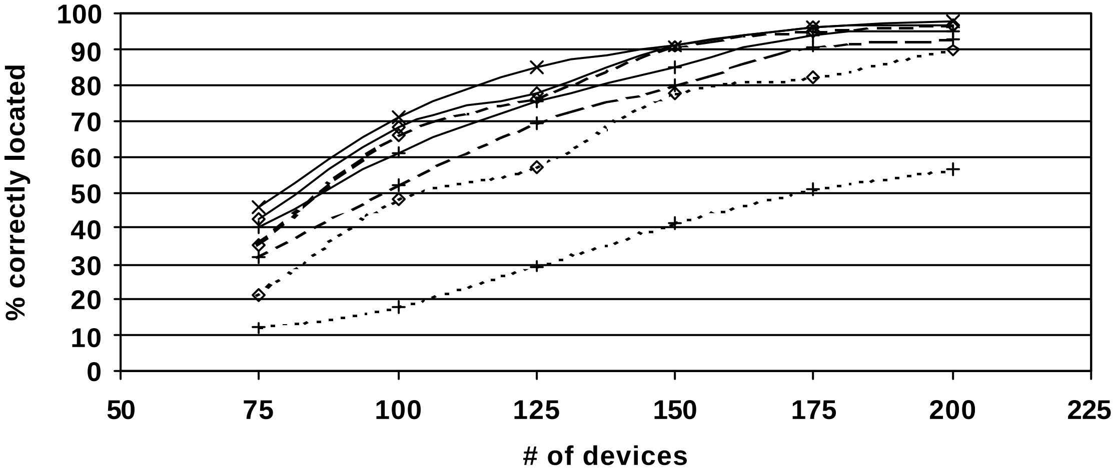

Figure 3 Graph percentage of devices located within 1 (dotted lines), 3 (dashed lines) and 10 (solid lines) of their real position for no (crosses), 5% (diamonds) and 10% (plusses) error in range determination against number of devices without external positional information in a standard grid with 16 square-packed, location aware devices.

significant advantage of meaning that the algorithm is potentially scalable to extremely large networks without an increase in the time taken to find a solution. The algorithm has already been shown to be resistant to noise errors in determination of the distances between devices and it is anticipated that it will be similarly resistant to small movements of the devices.

6 References

[1] A.B. McDonald and T. Znati, *A Path Availability Model for Wireless Ad-Hoc Networks*, Proc. of IEEE WCNC '99, New Orleans, LA.

[2] A. Ephremides, J.E. Wieselthier and D.J. Baker, *A Design Concept for Reliable Mobile Networks with Frequency Hopping Signaling*, Proc. of the IEEE, vol 75, no 1 1987, J. Sharony, *A Mobile Radio Network Architecture with Dynamically Changing Topology using Virtual Subnets*, MONET, vol 1 no 1, pp 75-86, B. Das and V. Bharghavan, *Routing in Ad-Hoc Networks using minimum connected Dominating sets*, IEEE Int. Conf. on Communications June 1997, & E. Royer and C-K Toh, *A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks*, IEEE Personal Communications Magazine, April 1999, pp. 46-55.

[3] J.-P. Hubaux, J.-Y. Le Boudec, S. Giodano, M. Hamdi, Lj Blazevic, L. Buttyan and M. Vojnovic, *Towards Mobile Ad-Hoc WANs: Terminodes*, IEEE WCNC, September 2000. & www.terminodes.org

[4] R. Want, A. Hopper, V. Falcao and J. Gibbons, *The Active Badge Location System*, ACM Transactions on Information Systems, Vol 10 No 1, pp 91-102, January 1992.

[5] R. Want and A. Hopper, *Active Badges and Personal Interactive Computing Objects*, IEEE Transaction on Consumer Electronics, Vol 38 No 1, pp 10-20, February 1992.

[6] P. Beadle, G. Maguire and M. Smith, *Using Location and Environment Awareness in Mobile Communications*, Proc. of the EEE/IEEE Int. Conf. on Information, Communications and Signal Processing, Singapore, pp. 1781-5, Sept. 1997.

[7] Z.J. Haas and B. Liang, *Ad Hoc Location Management Using Quorum Systems*, ACM/IEEE Transactions on Networking, April 1999

[8] J.J. Caffery and G.L. Stuber, *Overview of Radiolocation CDMA Cellular Systems*, IEEE Communications Magazine, April 1998.

[9] S. Capkun, M. Hamdi and J-P Hubaux, *GPS-free Positioning in mobile Ad-Hoc Networks*, 34th Hawaii International Conference on System Sciences, January 2001. & S. Capkun, M. Hamdi and J-P Hubaux, *GPS-free Positioning in mobile Ad-Hoc Networks*, Cluster Computing, April 2002.

[10] A. Ward, A. Jones and A. Hopper, *A New Location Technique for the Active Office*, IEEE Personal Communications, Vol. 4, No. 5, October 1997, pp 42-47 & www.uk.research.att.com/bat/