Flying Sinks: Heuristics for movement in sensor networks

Jeffrey V. Nickerson
Stevens Institute of Technology
jnickerson@stevens.edu

Abstract

Movement in wireless and sensor environments changes the degree to which we can communicate. Whereas sensor networks are generally seen as static, in many situations there is at least one component which moves, the data sink, which flies over a sensor field to integrate information. Also, it is possible to imagine sensors which, after they are deployed, move once into position. There are quality of service tradeoffs related to movement, for movement takes energy and time, but can increase integration, which we can measure in two ways. The utility of the sensor field is related to the number and size of its connected components. The pragmatic utility measures the communication back to human interpreters, and is a function of the periodicity of the transmission activity.

1. Introduction

Sensor networks are often seen as static, but usually there is at least one moving element. For example, in schemes involving small static sensors that are dropped in a location, a sink is often used to both synchronize and collect information from the network. A set of sensors may be deployed, and at that moment a sink may broadcast a signal that is used to synchronize clocks and provide a spatial reference for the small sensors [10, 14]. At some later point, the sink may return to integrate information the sensors have been gathering and relay this information to human decision makers. We can imagine the sink as flying over and performing the integration function.

We will refer to this event as synchronization; we mean by this the integration of information from the network, the relaying of this information to human interpreters, and the transmission of this integrated information back to the sensor network. Without the sink, the information from the sensors will not have a pragmatic effect.

We may think of the periodicity of the flyover as being a variable in a set of network tradeoffs. Research on tradeoffs related to networks is often described with the term *quality of service* (QoS); this term, which has a general applicability, focuses our attention on the many ways of measuring the utility of a network. A flying sink is one of many possible forms of movement that can be used to integrate an isolated sensor network. The relationship of movement to quality of service was shown in a paper which proved mobility can increase capacity in wireless networks [4]. Others have examined how different patterns of movement affect overall network characteristics [1].

Schemes for trading off mobility for bandwidth have been described in [3]. Recently, one researcher proposed mobility as a network control primitive [2]. Others have observed that mobility should be thought of as a potential tool in network design [12]. In previous papers, the author, in collaboration with others, has developed a concept of communication distance and applied it to mobile communications [7, 8, 11].

Most sensor literature focuses on trying to achieve total coverage with a set of sensors (e. g. [15, 17]). For many applications this makes sense. However, there may be some applications for which it is better to extend over a broad area with spotty coverage than over a narrower area with complete coverage. For example, if we want to know if people are crossing an area, it may not be essential to track every individual's every step. However, our ability to extend the range of sensing is limited by our need to let the sensors communicate with each other. There is another aspect to decisions on sensor placement; in adversarial situations, an irregular pattern may be better than a regular pattern for detecting an evasive opponent.

In this paper, we explore these ideas. First, we look at the dispersal of sensors, then at sensor movement. We use simulation to show the relationships between the radius of the flying sink and a number of dependent variables such as energy and time. This leads to a consideration of quality of service tradeoffs made by varying movement and thus affecting the utility of the network.

2. Sensor Placement

In mote-type sensor networks (for example, [13]), a number of small sensors may be dropped into a random pattern. If the sensors were to be placed carefully one at a time, they could be assigned positions on a lattice. However, if they are dropped, they will take on one of many possible patterns. If they fall densely and uniformly, then the sensors may form a connected component. It is not at all clear that they will fall in an evenly spread random distribution; if, for example, there are changing wind patterns as they drop, they will tend to clump. To guarantee a single connected component of sensors, we might have to compensate by dropping many more sensors than would be necessary if the sensors fell more evenly.

How, then, might we approach the problem of creating a connected component of sensors that minimizes the number of sensors while maximizing the area of coverage?

Some have suggested that we build sensor networks where the sensors are designed to reconfigure themselves once, and then freeze into position [15-17]. This kind of scheme acknowledges the difficulty of dropping sensors right the first time, yet seeks to avoid the energy problems involved in a continuously moving set of sensors.

It may be argued that such sensor movement is not plausible. Moving sensors will need to be capable of autonomy. They will need complex hardware and software installed to get a sense for location and terrain, making their deployment quite expensive.

We acknowledge this objection; however we do not think it impossible that, at some point in the future, insect-like propulsion could carry inexpensive sensors a number of yards to a more advantageous position. We do think that creating such sensors would be difficult, and it is worth asking before starting on such a complex undertaking what might be gained by sensor mobility. In this sense, we are performing a Gedanken experiment throughout the course of this paper. This experiment is far less costly than the program to actually build the moving sensors.

For now we consider sensor mobility possible. If sensors are dropped in a dispersed manner around the center of the area of interest, we could use a simple assembly heuristic [5], which we now describe briefly.

The sensors need to know the heading toward the center of the space, and the distance to be traveled. This might be accomplished through the sink issuing a strong signal at the drop time from the center of the field. The sensors then proceed toward the center in the following way. If they are at the center they stop. If they are in radio contact with someone who is stopped at the network center, they stop. Since the ad hoc network of the sensors is transitive, sensors on the edge of the field will only need to move until they overlap with sensors that have already been connected.

With such an arrangement, it is clear that a sink flying over any part of the aggregate can communicate to the new network.

In order to minimize the travel of the sensors, we might design the sink to have a much wider radio radius range (both transmission and reception) than the sensors.

In such a case, the sensors will only have to move into contact with the perimeter of the imagined circle, or connect with those that are already connected. If the circle is wide, many of the nodes may already be inside the circle. They may exist as components on their own. If the circle is extremely wide, the sensors never need to move.

The scheme has the advantage of tuning the amount of movement necessary, while guaranteeing that from one location, the information from the entire sensor network can be received.

To summarize, we initially configure a set of units around a virtual sink with a coverage area equal to that of the real sink. The movement heuristic for each unit calls on each unit to stop as soon as the unit is within a certain radius of the center, or is connected to some other unit which is. A sink then moves over the field; when the sink location overlays the virtual sink, the sensor data can be integrated. The time to integrate the data becomes the maximum of the time it will take to pull the sensors within a certain radius, and the time it takes for the sink to move over the central point. The latency of further communication is the periodicity of the sink's sweep over the sensor field.

We now proceed to demonstrate these ideas through the results of simulations.

3. Simulations

3.1 A convergence heuristic

Figure 1 shows a set of sensors as they might appear after a drop, presuming a random distribution. At a high enough density, in this case a density of 10%, with a coverage radius of 2, the sensors form into one connected component.

At a lesser density, the figure breaks up into several different connected components, as in figure 2. If sensor inputs can only be integrated within a particular component, the sensor is not as valuable as a detection device. For example, a component may identify an intruder, but have no way of notifying the rest of the network.

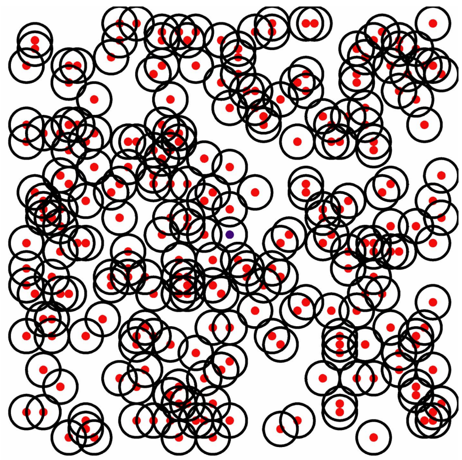

Figure 1. Density of .1.

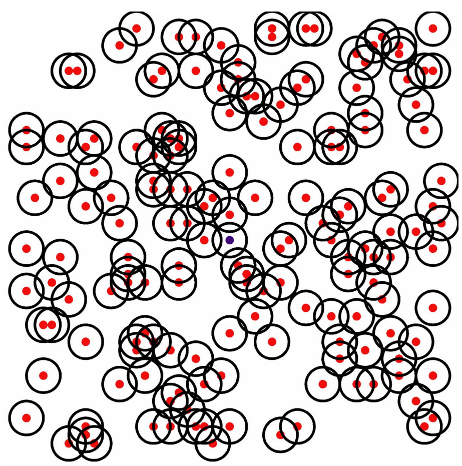

Figure 2. Density of .065.

The sensors can be pulled back into a single component, as is shown in figure 3. At each time step, the units move toward the center, and stop only if they either reach the center, or connect to a unit which has reached the center.

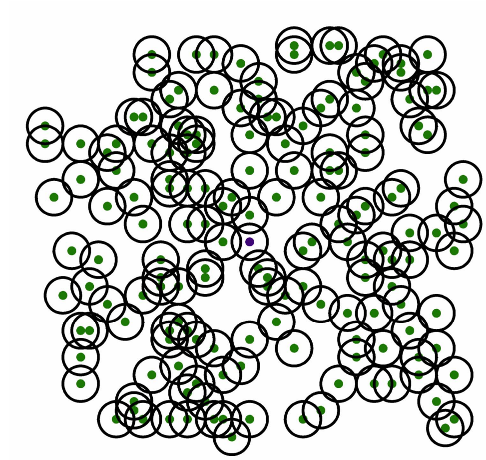

Figure 3. A convergence of the components of figure 2.

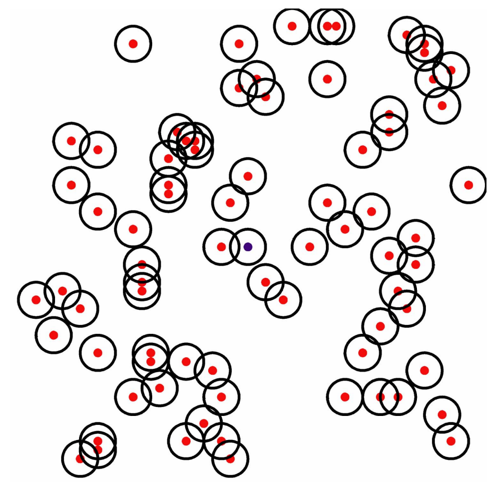

Figure 4. Density of .03.

At a lesser density still, the arrangement breaks up into many connected components, as in figure 4.

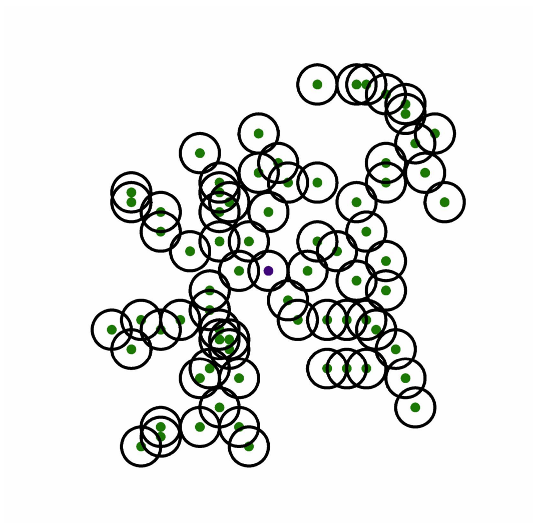

Figure 5. A convergence of figure 4.

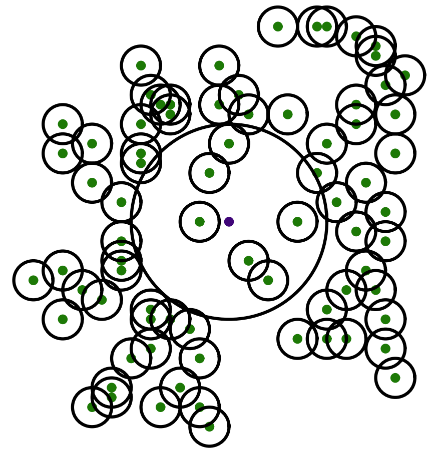

Figure 6. The large circle in the center is a virtual sink; all points converge on its perimeter.

Using the convergence heuristic, the components merge together. The patterns that form from the less dense arrangements increasingly look like axial tree structures; in related work, the author has explored the nature of these patterns [6].

Such patterns may be particularly useful in robotics applications, where robots need to coordinate their plans, and then disperse. In the case of sensor networks, there is an additional element, the moving data sink, which can be used to create another class of movement algorithms; we explain these next.

3.2 A flyover heuristic

Since the sink must fly over the field anyway, we might think of it as a missing element in the aggregate. Since the sink usually has a greater transmission and reception range, we may be able to take advantage of this range to create a network with a distinctive set of characteristics.

The units as before move toward the central point; but as we described before, we consider the central point to be virtual. The units stop when they are either touching the perimeter of the central circle, or are connected to someone who is, as in figure 6. Those inside the circle do not move; they are already considered connected.

When the sink flies over, at the instant it is over the central circle, the sink's perimeter matches the circle on the ground, and all information in the sensor network can be uploaded.

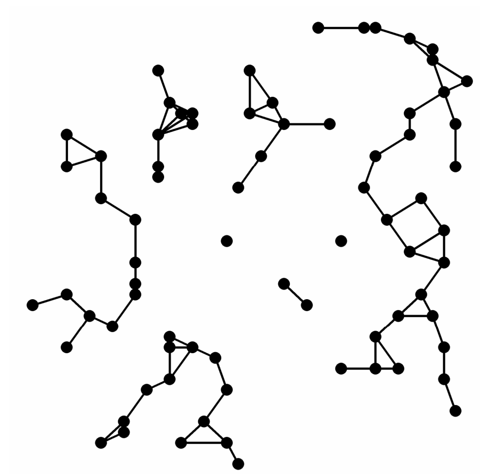

Figure 7. This shows the connected components of figure 6.

Figure 7 abstracts the diagram of figure 6, and shows the connected components. It is clear that the network will naturally partition into a set of connected components, depending on both the initial distribution and what happens in the movement toward the sink.

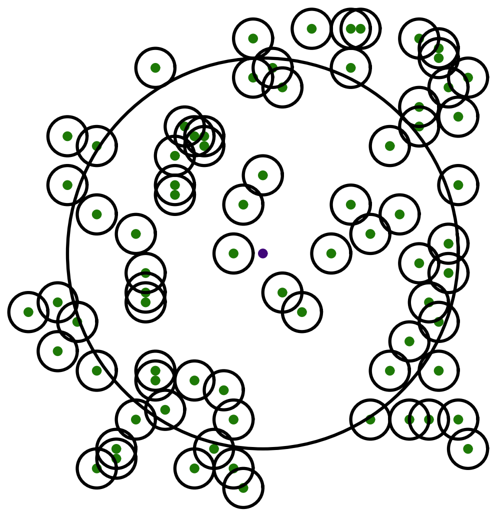

Figure 8. The same starting configuration of figure 6, but with a larger diameter sink.

Now we consider the effect of the size of the sink on the nature of the final aggregate. In figure 8, we can see that, as the sink area expands, a larger part of the initial distribution remains stationary in the center, and the outliers move in toward the periphery. At the extreme, if the sink is large enough, the initial sensor distribution doesn't change at all.

What happens if we systematically increase the sink diameter?

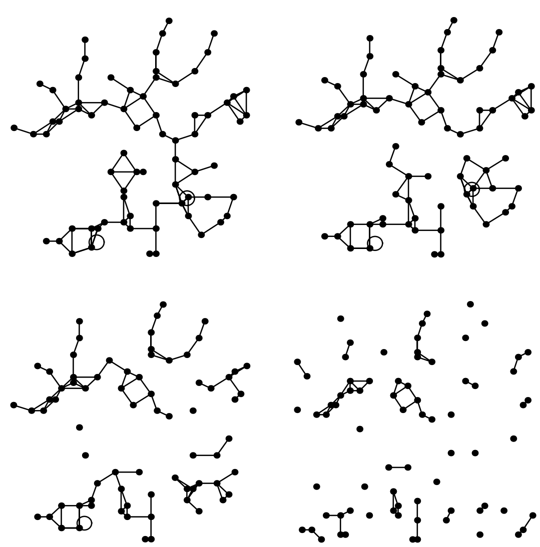

Figure 9. Graphs showing the final components for the same starting distribution as the sink diameter varies; the sink size increases left to right, top to bottom.

Figure 9 shows the result. As the sink size increases, so do the number of components.

Note that converging on a central sensor can be seen as a special case of converging on a virtual sink; if the size of the virtual sink is 0, then the integration of the sensor field will be complete.

3.3 Other possible integration mechanisms

While the idea of dropping sensors from airplanes and collecting information from flyovers permeates the sensor literature, other forms of integration are possible.

For the integration of the sensor field itself, let us reconsider sensor movement. Instead of building hundreds of moving sensors, we might build only a small set of moving sensors, and let them courier communication between the connected components.

This idea has its roots in research in mobile ad hoc networks (MANETs); for example, one paper explicitly analyzes the use of *runners* to communicate to disconnected entities in the network [1]. All entities in MANETs are presumed to move; however, it is easy to see the idea would work as a form of communication between static connected components. This is related to the idea of robotic couriers, who might, for example, bridge already connected sets of robots [5].

In the case of sensor networks, the number of connected components might not be known ahead of time, and so there might be an exploratory phase of determining where the components are, a planning phase of figuring out a pattern of movement, followed by a continuous operations cycle. If there was only one courier, the operations cycle might involve periodically traversing a Hamiltonian circuit of the connected components of the sensor field. At each step of the circuit, the information from all the other components is passed onto the visited component.

As with a sink, integration will take time, and there will be a deterioration of information quality within each connected component as the component waits for a new visit to update the aging information. Unlike with a sink, there may never be a moment in which all sensors know the current state of all other sensors. Every traversal between connected components will take time, so all information outside a particular component will be time-lagged. The lag time can be lessened by adding more couriers.

Now we turn to considering alternatives to the flying sink for the integration back to human monitors. We have postulated airplanes, but of course any vehicle might work. Automobiles, boats, blimps, and satellites all could perform the function if they can get within radio range of the sensor field.

A static device in the sensor field that can transmit a long distance might obviate the need for a vehicle at all. However, there are two difficulties with such devices.

First, such a device may be transmitting a signal with enough power that it might be detected by an adversary. Such a device will also to tend to be a good deal larger than a sensor, and thus more easily detected by eye. While over-flying vehicles are also targets, they are moving targets.

Second, such devices tend to consume a good deal of energy, and may rapidly become depleted after even a small number of transmissions.

If the first problem can be overcome, then the second problem might be solved by designing a device that harvests energy in some way. We might imagine a solar powered transmitter. It stores up energy from the sun, transmits, and then transmits again only when it has replenished its power. With such a scheme, we will have a pattern of periodic integration similar to that accomplished through flyovers.

4. The quality of service tradeoffs

4.1 Overview

As we vary the size of the flyover sink, we might expect there to be both advantages and disadvantages to the ensuing aggregates from a quality of service perspective. In this section we look at several tradeoffs. In each case, we simulated the tradeoffs on an exemplar random configuration.

In each case the *x* axis indicates the growing size of the virtual sink, and the *y* axis shows the dependent variable under discussion.

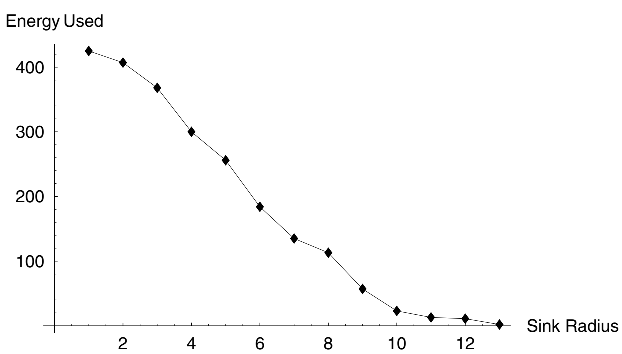

Figure 10. The energy expended is highest for the single integrated aggregate.

4.2 Energy

First, we look at the power consumed by the units in movement, as shown in Figure 10. We simply keep track of how far each unit has to move to get into the final configuration, and we sum these numbers. As the sink increases in diameter, the power used decreases, and eventually will go to zero when the diameter is greater than the starting diameter of the field, as in that condition all units are covered and none have to move.

4.3 Time

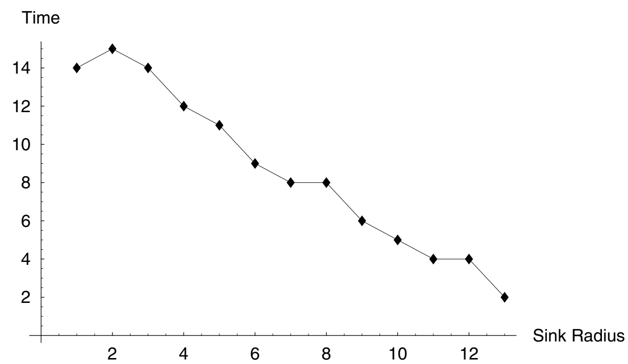

Figure 11. The amount of time to assemble into the final configuration.

Figure 11 shows that, just as the amount of energy reduces as the sink radius increases, so does the amount of time to form the final configuration. For not only are less units moving, the distance the farthest unit needs to move is proportional to its distance to the sink perimeter.

4.4 Extent

Now we look at a measure of the structure of the final aggregate after movement is complete. As the aggregates are irregular, we need some statistical measure of how far the aggregate extends; the radius of gyration provides such a measure, and is defined as:

$$R_g(m) = \sqrt{\frac{1}{m}\sum_{i=1}^{m} r_i^2}\ , \qquad (1)$$

where r_i is the distance to the center of gravity, in this case the central point.

A greater radius of gyration means the network will have a greater extent, which is usually a desirable property, as more territory can be covered.

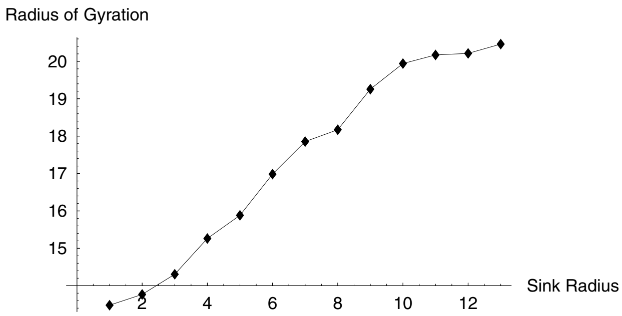

Figure 12. The radius of gyration extends as the sink radius grows

Figure 12 shows the growth of radius as the sink radius grows. When the sink radius is small, all sensors may move toward the center, decreasing the overall coverage of the sensor field. When the sink radius is large, no sensors move, and the extent is that of the original dispersion.

4.5 Sensor field utility

So far, increasing the sink size has resulted in two desirable properties; less use of energy and a greater extent of sensor coverage. Notice that the difference in energy is substantial, whereas the difference in radius is percentage-wise much less. However, for lower densities, the difference in the radius of gyration will be greater. From a sparse starting distribution, the movement heuristic will bring all units into a fairly narrow area if the sink size is small. If the sink size is large, then the highest density in the final aggregate will be around the perimeter of the sink. This again might be desirable for some applications; for example, in target tracking, a circular arrangement can be useful.

Yet there is value in aggregates that are continuously connected. In order to capture this connectivity, we might use a utility function based on Metcalfe's observation that the value of a network is proportional to the square of its connected members. In order to formalize the idea, we can think of the sensor network s as being partitioned into n connected components: $s = \{c_1, c_{2,\dots} c_n\}$. As we have a number of connected components, the *Metcalfe sensor field utility* is the sum of the squares of the cardinality of each component:

$$u = \sum_{i=1}^{n} |c_i|^2 . \qquad (2)$$

This leads to a strong penalty for disconnected components, as shown in figure 13.

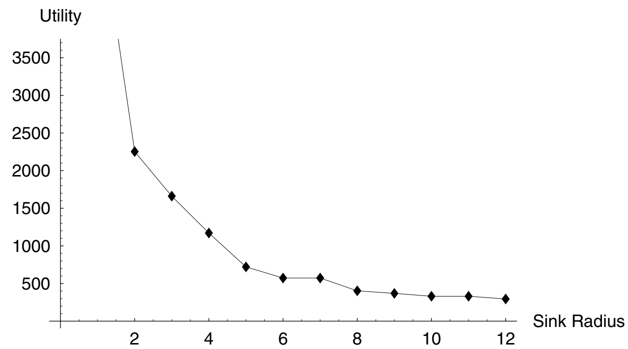

Figure 13. The utility of the sensor field using Metcalfe's law; the utility for a fully connected sensor field is off the chart at 5625.

However, Odlyzko and Tilly have argued that Metcalfe's law overstates the power of networks. They claim local connections are more valuable than other connections, and therefore the scaling of the value of the network, while greater than linear, is not quite as dramatic as Metcalfe's law suggests [9]. They proposed that the utility of a network is proportional to *m log m*, with *m* the number of nodes.

We can then define the *Odlyzko-Tilly sensor field utility* as:

$$u = \sum_{i=1}^{n} |c_i| \log |c_i| \qquad (3)$$

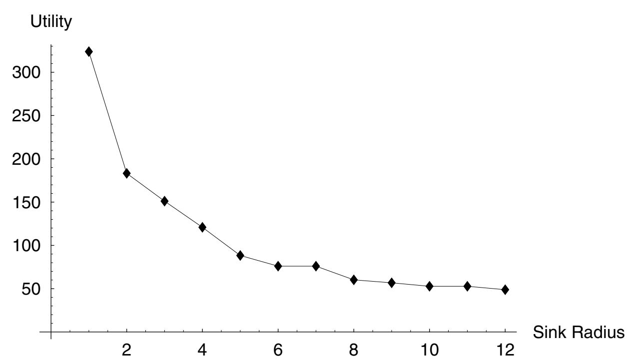

Figure 14. The utility of the sensor field using *m log m* instead of Metcalfe's law.

Figure 14 shows that an *m log m* utility function still gives a single component sensor field a much higher utility; whether this metric is more realistic or useful than Metcalfe's is a subject for future research.

4.6 Pragmatic Sensor Network Utility

We now introduce the concept of *pragmatic sensor network utility*. For if a sensor network is isolated, the pragmatic effect of it will come from the transfers of information to the sink, which serves as a relay to a remote location where human interpreters will make sense of the information. In other words, a static sensor network can't manifest pragmatic value until the information is transmitted to some interpreter capable of taking action.

We may approach the modeling in steps. Let us imagine that the entire sensor network is one component, continuously connected. A sink flies over at a time interval g (the gap), and simultaneously transmits the information back to the interpreter. In between time t and time $t + g$, the information gained becomes less timely, and therefore less valuable. If U_0 is the utility of the fully connected network at the time of integration, then we can model the deterioration of the utility as:

$$U_g = U_0 e^{-ag}, \tag{4}$$

where a is a constant controlling the rate of information decay.

The expected utility over the interval g is then

$$< U > = \frac{1}{g}\int_0^g U_0 e^{-ag}\, dg = U_0 \frac{1 - e^{-ag}}{ag}. \tag{5}$$

The expected pragmatic utility can be controlled by the frequency with which we fly over and gather information. There is a cost associated with these flyovers. From a quality of service perspective we can trade off the cost of the flyovers with the pragmatic utility.

For example, a vehicle might hover directly over the sink area; this will yield the highest level of integration. One cost associated with this activity is fuel consumption. In adversarial situations, there are additional costs. Flyovers might both provide information to an opponent and draw attack; the more frequent the flyovers, the more information revealed, and the greater the probability of a successful attack against the sink. Hovering might create an attractive target, and hence have a higher cost.

If the number of components is greater than 1, then the sink flyover performs two functions – not only does it transmit information back to a human network, but it performs the role of integrating the information in the components which have been isolated until that moment. At the instant of synchronization, the utility of the network jumps; however, the impact of the synchronization deteriorates over time. We might model this in the following way, using the Odlyzko-Tilly sensor field utility as a base:

$$u_{c_i} = |c_i|\log|c_i| + \frac{|c_i|}{|s|}\left(|s| - |c_i|\right)\log\left(|s| - |c_i|\right)e^{-ag}. \tag{6}$$

The utility of a connected component is augmented by a term that considers the residual power of the last integration activity across the rest of the sensor field, and amortizes the residual power according to the size of the connected components. In other words, a component knows the present state of all its sensors. It knows only the past state of all the other sensors in the network; as time goes on, this past information is less and less useful.

Improvements can be made to such a function (for example, in adding the utility of the components together we ideally should get back $|s|\log|s|$ if $g = 0$; for this we will need another term); the general point is that the more frequent the synchronization activities, the better the network will perform locally.

We have still not captured the full complexity of the interplay between sensor network utility and pragmatic utility. The pragmatic utility at time 0 when the sink flies over has two aspects. First, there is the aspect we have modeled: when the sink is directly over the sensor field, the entire power of the network is in effect at that instant. If the sensors maintain history, then the sink also at that time can integrate on the time series data.

The utility of integrating the time series may be a function of how subdivided the network is. For example, imagine a network whose functions include the tracking of moving objects. A set of entirely isolated sensors may not have been able to determine the location of something passing through the sensor field, whereas, through triangulation, a connected field will be able to. The more connected the field, the more likely the position information will be accurate, and the more likely that false positives might be screened out through Bayesian techniques.

Then, our pragmatic utility should contain a term that in some way penalizes the extent to which the network is disconnected. The penalty function might be modeled as a fraction of the difference between the utility of the fully connected network and the segmented network:

$$U_0 = |s|\log|s| - b\left(|s|\log|s| - \sum_{i=1}^{n} u_{c_i}\right). \tag{7}$$

If the sensor field is fully connected, then the second term is 0. The constant b controls for the degree to which the fragmentation of the sensor field is seen as detrimental.

5. Conclusions

Sensor network systems are made up not only of the sensors, but also of the sink and the interpreters of the data. In looking at situations with a moving sink, we have discussed ways sensors might move once to be within range of the sink when it flies over. If sensors move a great deal, they can more readily integrate with each other, but the movement creates energy costs and decreases the radius of the sensor field. We have discussed the tradeoffs that occur in the decision to create fully overlapping versus partitioned sensor fields. In particular, we have shown that a widening sink radius will decrease energy use, decrease time for sensors to position themselves, and increase the area of coverage, all desirable things. However, widening the sink radius also fragments the sensor field into multiple components, creating an integration problem.

We might choose to let sensors remain partitioned. We may have no choice; sensor mobility may be restricted or impossible. Then, an integration mechanism such as a flyover can be used to integrate the sensor field. Flyovers are not the only possible integration mechanism; for example, several local moving units might accomplish sensor field integration across connected components by serving as couriers.

We have distinguished between sensor field utility and pragmatic sensor network utility. As information value will deteriorate over time, the decision on the periodicity of the integration back to human interpreters also involves tradeoffs. More frequent integration will increase the utility of the information, but will also consume energy.

A more general point can be made. By regarding the movement of sensors and sinks as variables, we can design sensor networks with a variety of different characteristics which may provide qualities of service appropriate to particular applications.

Acknowledgements

This research was supported by the Office of Naval Research, Swampworks, grant # N00014-05-1-00632 and by the National Science Foundation under grant #ITR-032630. Also, the author would like to acknowledge the suggestions of Tom Barnes.

References

[1] I. Chatzigiannakis, S. Nikoletseas, N. Paspallis, P. Spirakis, and C. Zaroliagis, "An Experimental Study of Basic Communication Protocols in Ad-hoc Mobile Networks," *Lecture Notes in Computer Science*, vol. 2141, pp. 159-169, 2001.

[2] D. K. Goldenberg, J. Lin, A. S. Morse, B. E. Rosen, and Y. R. Yang, "Towards Mobility as a Network Control Primitive," *MobiHoc*, 2004.

[3] D. J. Goodman, J. Borras, N. B. Mandayam, and R. D. Yates, "INFOSTATIONS: a new system model for data and messaging services," *Vehicular Technology Conference*, 1997.

[4] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, pp. 477-486, 2002.

[5] J. V. Nickerson, "Robots and Humans Reconvening," *IEEE International Conference on Systems, Man and Cybernetics*, 2004.

[6] J. V. Nickerson, "Rally Point Simulations," http://www.stevens.edu/jnickerson/rallypoints 2004.

[7] J. V. Nickerson, "A Concept of Communication Distance and its Application to Six Situations in Mobile Environments," *IEEE Transactions on Mobile Computing.*, vol. 4, no. 3, 2005.

[8] J. V. Nickerson and S. Olariu, "Protecting with Sensor Networks: Attention and Response," *HICSS*, 2006.

[9] A. Odlyzko and B. Tilly, "A refutation of Metcalfe's Law and a better estimate for the value of networks and network interconnections," http://www.dtc.umn.edu/~odlyzko/doc/metcalfe.pdf 2005.

[10] S. Olariu, A. Wadaa, L. Wilson, and M. Eltoweissy, "Wireless sensor networks: leveraging the virtual infrastructure," *IEEE Network*, vol. 18, no. 4, pp. 51-56, 2004.

[11] S. Olariu and J. V. Nickerson, "Protecting with Sensor Networks: Perimeters and Axes," *MILCOM*, 2005.

[12] R. Rao and G. Kesidis, "Relaying and Surveillance in Mobile Ad Hoc Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 225-232, 2004.

[13] P. Saffo, "Sensors, the next wave of innovation," *Communications of the ACM*, vol. 40, no. 2, pp. 93-97, 1997.

[14] A. Wadaa, S. Olariu, L. Wilson, M. Eltoweissy, and K. Jones, "Training a wireless sensor network," *Mobile Networks and Applications*, vol. 10, pp. 151-167, 2005.

[15] G. Wang, G. Cao, and T. L. Porta, "Movement-Assisted Sensor Deployment," *Infocom*, 2004.

[16] Y. Zou and K. Chakrabarty, "Sensor Deployment and Target Localization Based on Virtual Forces," *Infocom*, 2003.

[17] Y. Zou and K. Chakrabarty, "Sensor Deployment and Target Localization in Distributed Sensor Networks," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 1, pp. 61-91, 2004.