Biology Inspired Approach for Communal Behavior in Sensor Networks

K. H. Jones
NASA Langley Research Center
Hampton, VA 23681
k.h.jones@nasa.gov

K. N. Lodding
NASA Langley Research Center
Hampton, VA 23681
kenneth.n.lodding@nasa.gov

S. Olariu
Old Dominion University
Norfolk, VA 23529
olariu@cs.odu.edu

L. Wilson
Old Dominion University
Norfolk, VA 23529
wilson@cs.odu.edu

C. Xin
Norfolk State University
Norfolk, VA 23504
cxin@nsu.edu

Abstract

Research in wireless sensor network technology has exploded in the last decade. Promises of complex and ubiquitous control of the physical environment by these networks open avenues for new kinds of science and business. Due to the small size and low cost of sensor devices, visionaries promise systems enabled by deployment of massive numbers of sensors working in concert. Although the reduction in size has been phenomenal it results in severe limitations on the computing, communicating, and power capabilities of these devices. Under these constraints, research efforts have concentrated on developing techniques for performing relatively simple tasks with minimal energy expense assuming some form of centralized control. Unfortunately, centralized control does not scale to massive size networks and execution of simple tasks in sparsely populated networks will not lead to the sophisticated applications predicted. These must be enabled by new techniques dependent on local and autonomous cooperation between sensors to effect global functions. As a step in that direction, in this work we detail a technique whereby a large population of sensors can attain a global goal using only local information and by making only local decisions without any form of centralized control.

1. Introduction

The dream of ubiquitous machinery was born of the Industrial Revolution: that machinery could eventually be developed to service man's needs, particularly replacing man’s direct participation in physical work for provision of food, clothing, housing, and other necessities. The realization of this dream has progressed for several centuries but has taken a revolutionary leap through the invention of electronic computers, which facilitate more automation of machines. Futurists have long predicted that machines would eventually communicate and cooperate with each other to accomplish extraordinarily complex tasks without human intervention [20].

While futurists long dreamed of machines working with other machines, a giant step towards realization of this dream may be credited to a DARPA sponsored program, SmartDust, originated in the late 1990’s [11,26]. The title of the program creatively described its goal: to make machines with self-contained sensing, computing, transmitting, and powering capabilities so small and inexpensive that they could be released into the environment in massive numbers. Following their work, we define a sensor network as follows: *A sensor network is a deployment of massive numbers of small, inexpensive, self-powered devices that can sense, compute, and communicate with other devices for the purpose of gathering local information to make global decisions about a physical environment.*

The National Research Council's (NRC) Committee on Networked Systems of Embedded Computers published a report [14] defining the concept of the embedded network, EmNet, as a network of heterogeneous computing devices pervasively embedded in the environment of interest. Since that time, the term *wireless sensor network* (sensor network, for short) has evolved to describe such systems and research abounds in academic and commercial environments. Improvements in Micro

Electro-Mechanical Systems (MEMS) technology is producing, at rapidly decreasing cost, smaller and smaller self-powered devices that can sense, compute, calculate, and communicate [26,28]. These devices have come to be called sensor *motes* [27] and serve as nodes in a sensor network. As these new motes are severely energy constrained, they cannot transmit long distances. Much current research is in how to form and execute multi-hop networks to send information to and from the network motes where no single mote need transmit any farther than to nearby motes.

As technology improves, computing and communicating capabilities are expected to improve at an accelerating rate and new techniques for supplying energy will significantly reduce the low power constraint [1,20]. Increased capabilities will be possible and futurists predict that societies of machines will evolve to be autonomous, cooperative, fault-tolerant, self-regulating, and self-healing [20]. Rather than adapting conventional techniques of centralized computer control, new techniques dependent on local cooperation among network nodes will lead to self-sustaining communities of machines with emergent behavior that autonomously operate and adapt to changes in the environment. This evolution so parallels the development of life on Earth that living systems provide models for system design.

Due to the limited resources of available motes, most current implementations of sensor networks are composed of a small number of motes that collect simple sensory values and report these values to a central computer [5,13]. In Section 2, we describe several current implementations and compare their similarities. These conventional techniques are insufficient to fulfill the full potential of sensor networks. Centralized control of the sensor network, by definition, will not scale to large numbers of motes. As the size of the network increases, more and more demand is placed on the central controller and, in a multi-hop network, motes nearest the central controller will be bottlenecked passing messages to and from other motes further away. If the sensor network is to interact with the environment in ways predicted, it must effect the environment as well as sense information about it. We believe to reach its full potential, the sensor network must behave as a community of organisms, where individual motes function based on local information, making local decisions that combine across the sensor network to impact global functions. In Section 3, we describe how living systems may serve as models for communities of motes in a sensor network and describe how cellular automata may be used to simulate a sensor network. In Section 4, we describe how a simple global function, the calculation of an average value of randomly distributed values among motes in a sensor network, can be accomplished by the communal cooperation of motes where they are limited to local information and act only on their local environment. In Section 5, we conclude with a comparison of current implementations and our methods.

2. Current Sensor Network Implementations

In the past few years we have witnessed the deployment of sensor networks in support of a growing array of applications ranging from smart kindergarten [17,19,23], to smart learning environments [8], to habitat monitoring [18,24], to environment monitoring [6,7,13], to greenhouse and vineyard experiments [5,10], to helping the elderly and the disabled [8], among others. These prototypes provide solid evidence of the usefulness of sensor networks and suggest that the future will be populated by pervasive sensor networks that will redefine the way we live and work [1,4]. It is, thus, to be expected that in the near future, in addition to the examples above, a myriad of other applications including battlefield command and control, disaster management and emergency response, will involve sensor networks as a key mission-critical component.

A closer examination of current implementations of sensor networks reveals both successes and limitations. In the spring of 2002, researchers at University of California, Berkeley (UCB), collaborated with the College of the Atlantic in Bar Harbor to install a sensor network on Great Duck Island, Maine. The initial application was to monitor the microclimates of nesting burrows of the Leach's Storm Petrel, and, by disseminating the data worldwide, to enable researchers anywhere to non-intrusively monitor this sensitive wildlife habitat [18,24]. The sensor motes were placed in the habitat and formed a multi-hop network to pass messages in an energy efficient manner back to a laptop acting as a sink. The data was intermediately stored at the sink and eventually passed by satellite to servers in Berkeley, CA, where it was distributed via the Internet to any interested viewer. The sensors measured temperature, humidity, barometric pressure, and mid-range infrared by periodically sensing and relaying the sensed data to the sink. The largest deployment had 190 nodes with the most distant placement approximately 1000 feet from the nearest sink.

Other implementations demonstrate similar capabilities. In the SmartDust demonstration at the Marine Corps Air/Ground Combat Center, Twentynine Palms, CA [29], six sensor motes were dropped from a Unmanned Aerial Vehicle (UAV) along a road. These motes self organized by synchronizing their clocks and forming a multi-hop network. They magnetically detected vehicles passing and reported the time of the passing. Using information collected from all motes, velocity of passing vehicles was computed.

In August 2001, researchers from UCB and the Intel Berkeley Research Lab demonstrated a self-organizing sensor network to those attending the kickoff keynote of the Intel Developers Forum. Several students each brought on stage a wireless sensor mote, activating it independently. As the motes were initiated, their icons appeared on a display with lines connecting all motes that could “hear” each other and highlighted lines depicting a multi-hop routing structure over which sensor data could be transmitted to a central collector, a PC. In a second, large-scale demonstration of the networking capability, the quarter-sized motes were hidden under 800 chairs in the presentation hall and simultaneously initiated forming what was described as, “… the biggest ad hoc network ever to be demonstrated” [12].

The FireBug system [5,6] is a network of GPS-enabled, wireless thermal sensors motes, communicating through a control layer for processing sensor data, and a command center for interactive monitoring and control of the sensor network. The FireBug network self-organizes into clusters where cluster leader motes act as sinks, receiving sample data from cluster members and brokering commands to these members. The controller is a personal computer running the Apache web server interfaced with MySQL using PHP. The FireBug Command Center allows user interaction for controlling the FireBug network and displays real time changes in the network.

These current sensor network implementations represent pioneering work that demonstrates the enormous promise of the sensor network concept. They share a number of similarities. Most of the motes used are similar in capability to those used in the UCB-Intel large-scale demonstration [12]: a 4 MHz microprocessor with 16 KB of flash instruction memory, 512 bytes of SRAM, a 256 KB EEPROM for secondary storage, and a 10 Kbps radio transmitter. Each network has homogeneous motes although the motes may be multifunctional having multiple sensing capabilities and serving other functions such as routers, cluster leaders, etc. PCs serve as sinks for interfacing with the world outside the sensor network. However, all are passive, sensing systems (i.e., no effectors) observing the environment and reporting these observations to a central authority where decisions will be made often by human observers.

However, sensors for measuring the parameters in these implementations and the ability to feed their measurements into a computer are not new. What was novel about these approaches is the small size of the sensors and their wireless networking connection. This allowed easy and inexpensive installation of these motes directly into the environment. Using wireless communication, motes do not require an expensive communication infrastructure and expense does not increase exponentially with the number of motes as in wired networks (i.e., more motes would need more wire). Most importantly, both the small size of the motes and the wireless network enabled an installation that was unobtrusive compared with what was previously possible. An obstructive installation required by larger equipment may interfere with the ecosystem in a biological application. In other applications, the introduction of wiring infrastructure can delay projects due to intrusions into other infrastructure (i.e. may require additional approval or may interfere with design), and therefore be unacceptable.

Current sensor networks are modeled after conventional computing networks under centralized control and involve a small number of motes (between 6 and 800). It is, therefore, not clear that they provide a credible approximation of the massive deployment envisioned by the proponents of sensor networks [1,4,11,26]. Moreover, in these designs, the mote’s behavior is predetermined and a central authority otherwise manages its collected results. Motes appear to be working together, though only in simple ways: multi-hop networks are formed and clocks are synchronized.

An examination of the method used for clock synchronization in the Twentynine Palms demonstration [29] reveals that it closely models the behavior of biological singletons such as bacteria. Each mote periodically broadcasts its current time. Each mote that receives the broadcast updates its time if the broadcast value is greater than its own. Though effective, are the motes really working together? Or are they singletons following simple rules: they periodically broadcast their time and they listen for the broadcast of packets containing time values. Do they know or care that other motes exist? From the individual application of these simple rules, a simple behavior for the group *emerges*: their times are synchronized. However, neither the individual or group behavior can be described as cognitive.

We believe that in order to scale to massive deployment, sensor networks benefit from lessons learned from the way biological ecosystems are organized, with motes acting as organisms interacting locally with the environment in a way that produces global effects that facilitate a functional and sustainable sensor network. Along this line of thought, our work centers on three fundamental characteristics of future sensor networks that are not demonstrated by the described implementations:

1. We are developing techniques that do not bottleneck by dramatically increasing the number of motes in a sensor network.
2. We demonstrate motes that do more than sense a value and report that value. Our motes are designed to be asynchronously autonomous, learning from their environment to make better decisions.
3. We design sensor networks that do not require centralized control.

3. A Bio-mimetic Model for a Sensor Network

Sensor motes can be considered *cells* of tissue (for homogeneous motes) or organs (for heterogeneous motes) in a multi-cellular organism or they may be considered *organisms* within a community. In either case, they are endowed with genetic material: an initial *state* and *rules* by which they interact with the environment. Both state conditions and the rules may or may not change as the mote interacts with the environment. Additionally, the mote may "*remember*" its interaction with the environment by storing information in memory. Memory and the use of memory to change state or rules are considered learning. Changing state conditions or rules based on learning demonstrate some level of cognition.

Our goal is to demonstrate how learning and cognition can benefit the functionality of a sensor network. In particular, we are interested in using these attributes to enable *local decisions* from *local information* that effect a *global* result. Limiting decisions to localities is important for two reasons in sensor networks:

- ***Scalability***: If all motes must be contacted and/or instructed for a function, then the communication and data management overhead grow substantially with the number of motes in the deployment. Thus, larger deployments become increasingly costly and problematic.

- ***Autonomy and intelligence***: local decisions allow distributed control. Distributed control through local decisions provides a natural redundancy affording fault tolerance (i.e., if some motes fail, others will continue to make decisions). As each mote is operating autonomously, they may learn differently and respond differently just as organisms do. Some may be more successful than others allowing evolution to operate.

Current implementations of sensor networks represent singleton organisms (e.g. bacteria) operating with no knowledge of others in the community: they sense and report. They do not know why they are doing this, they do not know what others are doing, and they do not know the effect their actions have on the community. Our research is in demonstrating progressively "higher" life forms for sensor networks. Initially, motes in our designs go beyond singleton behavior: they have memory, are aware of their neighbors, and make local decisions. Our goal is to design increasingly capable motes that will cooperate with one another as a society with each mote operating in its own interest, yet contributing positively to the community.

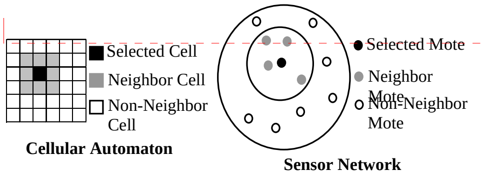

Figure 1: Illustrating the neighborhood of a cell

Comment [CSD1]: The text "Neighbor Mote" and "Non-Neighbor Mote" on the right part of the figure are overlapped

We use cellular automata to simulate sensor networks operating as organisms in an ecosystem. A cellular automaton represents, in most ways, a distribution of sensor motes throughout a geographic region. As illustrated in Figure 1, each internal cell is surrounded by 8 neighbor cells. Border cells have 3 or 5 neighbors. Neighbor cells represent those sensor motes that can receive a transmission from a cell.

Thus, the regularity of the grid represents a logical indication of physical proximity. Selectively, neighbors may not be limited to the eight adjacent cells. Increasing the number of cells that can receive transmissions from the selected cell is specified by the neighbor *radius*. A radius of 2 would include the 8 adjacent cells plus the 16 cells adjacent to these neighbors. The only limitation of this model is that the number of neighbors is fixed for a given radius, but this can be changed by disabling some of the neighbors.

4. Aggregating Sensed Data: Average Calculation

There are, essentially, two ways in which data sensed by the motes can be aggregated. In a centrally-controlled network, data aggregation is a two-stage process: in the first stage the motes forward the data collected to the sink. Some of the data is fused en route but the final responsibility for aggregation rests with the sink. In the second stage, the aggregated result is broadcast back to the network. Though straightforward, this method does not scale well. By contrast, in a truly distributed system, as is the case in an autonomous sensor network, the motes cannot rely on the sink to perform the aggregation. Instead, the aggregation must be performed *in-situ* by the sensors themselves. To illustrate, suppose a distributed system has capabilities to both sample the temperature of its immediate environment and to set that temperature. The goal is for each mote to maintain the average temperature throughout the domain. As the temperatures in localities drift, this average value must be recalculated and localities reset to the average.

More generally, it is often the case that a community must reach a *consensus*, that is, an agreement on a common state when individuals are in different states from some domain of states. An example of a consensus problem is clock synchronization. Synchronization is a straightforward task in a centralized-control system: here the central entity dictates the common time. The situation is vastly different in a distributed system. Even if we assume [15,25] that at initialization, all clocks in a network of distributed systems may be set to a common value, over time, due to clock drift, the motes in the network will no longer have a common value. The system must have a procedure to periodically return to a consensus on a common clock setting for each mote. While a number of solutions to the aggregation and synchronization problem have been proposed in the literature [1,2,3,8,15,16,21,22] they were either designed for sensor networks of small size or have a centralized flavor.

One of the key contributions of this work is to show that fully distributed data aggregation can be performed in a scalable fashion in massively deployed sensor networks. A cautionary note is in order, however. For real applications such as the above, the goal is not to have the common value equal across the system. Especially because the values are constantly changing, it is acceptable for the value at each node to be within a given tolerance of the desired common value. For such applications as clock synchronization, this tolerance must be quite low, but for applications such as the temperature alignment, a larger tolerance may be acceptable [1,3].

As our cellular automaton is defined such that each cell can only transmit to its immediate neighbor, on average, data collected by each cell must pass approximately half the span of the grid. A suggested approach to conserve mote energy is for the sink to broadcast a return message to all motes. But this method does not scale well as the power of the sink's transmission would have to grow with the size of the sensor network distribution. Thus, we assume return messages must be sent back through the multi-hop network and the cost in hops on average increases with the span of the grid. Figure 2 depicts this expansion for grids up to 1000 x 1000.

Comment [CSD2]: Since, it's not exact half of the span, I use an "approximately " here

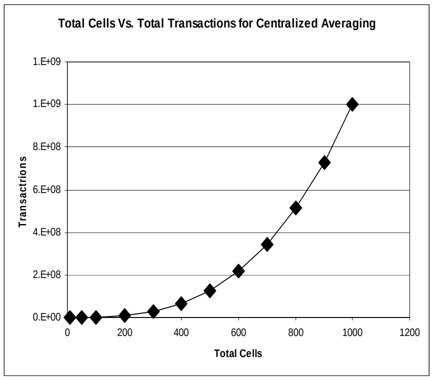

Figure 2. *Illustrating the complexity growth for centralized averaging*

However, this is not the worst problem with the centralized approach. It was recently noticed by Wadaa *et al*. [25] as well as by other workers that in a centrally controlled sensor network, the cells closer to the sink must serve as routers for many mote-to-sink paths, depleting their energy budget much faster than other motes in the network. Indeed, Wadaa *et al.* [25] have noted that by the time the motes close to the sink have expended all their energy some other motes in the sensor network still have about 80-90% of their original energy budget. This uneven energy depletion effectively creates an *energy hole* around the sink, severely curtailing network longevity. Figure 3 illustrates the uneven energy depletion problem. In our cellular automation, the cell closest to the sink must relay messages to and from every cell in the grid and their energy budget will decrease rapidly relative to cells further away. The longevity of the network is then dependent on the rapidly depleting budget of the cells closest to the sink. Indeed, in Figure 3, all black cells must pass messages through a single cell in the top row.

Our objective is for the sensor network to calculate an average value across the network without any mote, or any central authority, ever knowing all the data values. Suppose motes are randomly distributed and that each collects a value relevant to some characteristic of its locale. An average value must be calculated for the entire network and this average value must be assigned to each mote. The challenge is to attain this goal with all cells requesting/dictating information only from/to their neighbors.

We now give an informal description of our algorithm. Assume that the data value collected by each mote is converted to a real number - a *color* - in the range [0, 255]. Upon deployment, the motes are endowed with genetic material (see [15,25] for details): a *status* (*active* or *inactive*) and a *transmission time* within a specified *time period*.

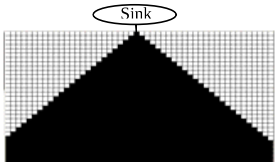

Figure 3. *Illustrating the energy-hole problem*

When the algorithm begins, each mote starts its own clock. The transmission time assigned is some time step within the time period (e.g., one mote may transmit at step 3, another at step 18, but all motes will transmit some time during the time period. Because the transmission time for each mote is a random number, there is no guarantee that two or more motes will ever transmit at the same time. If two motes are at a neighbor radius greater than 3, they may execute the algorithm simultaneously, as the results of their calculations are independent of each other. If the radius is less than 3, the result of calculations is *order-dependent*. In this case, simultaneous transmissions will cause collisions, thus a MAC layer protocol is assumed to decide cellular execution order. To simulate this control, all motes (i.e. cells) *selected* at the same transmission time are executed in random order. The order is randomized anew with each time step. The transmission time is fixed at deployment (being part of the genetic material), but the execution order in each time step is a "function of the environment" and may change each time step.

Setting the time period to a value larger than the size of the total number of cells increases the probability that each cell will transmit independently. This provides for the most repeatable simulation as a given color distribution with a given transmission time assignment will execute exactly the same each time, provided none transmits simultaneously. This does increase the execution time of the simulation, as only one cell is selected each transmission time. Reducing the time period speeds the simulation but introduces more randomness for each time step. Setting the time period to 1 guarantees that all cells will be selected in that time step, though in new random order for each period. The results are very similar regardless of the length of the time period; specifying a longer time period just requires more time during which nothing happens. Even with MAC layer negotiations, by this method each cell is acting *asynchronously* and *autonomously*. Because the time and cost of reaching a global decision is a function of *color* distribution, transmission time assignment, and execution order in each time step, we ran multiple executions varying all three parameters and averaged the results.

When a cell reaches its transmission time within the time period, it is selected for action. If its status is inactive, it does nothing. Otherwise, if its activity status is active, it begins a series of *transactions*. A transaction is either a request for information from a neighbor (e.g, what is your color?) or a specification given to a neighbor (e.g., set your color to 134). Transactions are significant because they require radio transmissions, typically the most costly activity of a mote. If each neighbor's color is equal to the color of the selected cell, it sets its status to inactive and will not participate again when selected until it is reactivated. An inactive status is significant in that no transactions occur when a selected cell is inactive. If any neighbor's color is not equal to the selected cell's color, the *selected* cell computes the average for the neighborhood and sets itself and all neighbors to this value. During this process, if any neighbor is inactive, it is reactivated as it now has a color different from some of its neighbors. The simulation continues until all cells are set to inactive. Thus, the simulation ends using only local information; no global control is required. The algorithm can be proven to converge in a finite number of time periods (details are omitted here), and as can be seen from simulation results, it actually converges quickly.

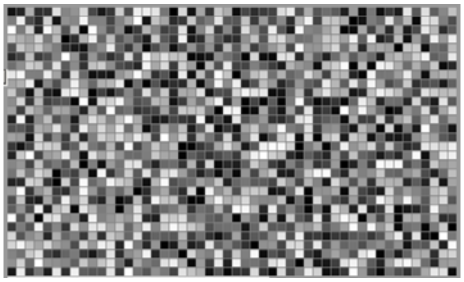

Figure 4: Initial distribution of colors for color averaging

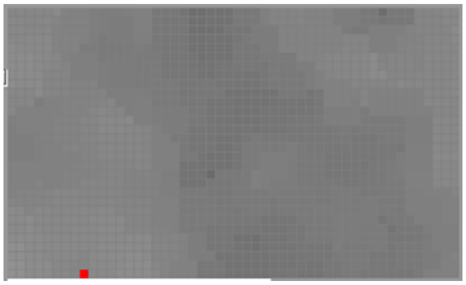

Figure 5: Distribution of colors after the first time period

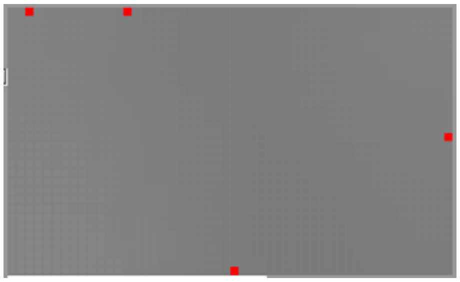

Figure 6: Distribution of colors after 10 time periods

Figure 4 depicts an initial distribution in a cellular automaton with a 30 x 50 grid of cells showing a random distribution of colors. Figure 5 depicts the color change after the first time period (i.e. all cells have been selected once). Notice that after a few iterations the cells are much closer to the same color. Figure 6 illustrates the color change after 10 periods. Soon thereafter, the color differences are indistinguishable to the naked eye.

Inactive cells are identified by the color red. Interesting color patterns are displayed as the simulation nears completion: the number of red cells varies but increases until all are red.

The converged average is always equal to the average of the original distribution calculated prior to starting the algorithm An interesting and valuable attribute of this algorithm is that the average color at each time step is always equal to the initial calculated average. Re-examining the process reveals the reasons for this state of affairs. A group of nine cells of different colors contributes to the average of the total grid. When these cells are averaged and set to the same average value for the group, they contribute exactly the same to the average of the entire grid as they did with differing values.

Figure 7 compares the increase in transactions for both the centralized and decentralized approaches. As shown above, the centralized approach increases exponentially while the decentralized approach, though initially more costly than the centralized approach, shows a slope asymptotically approaching zero.

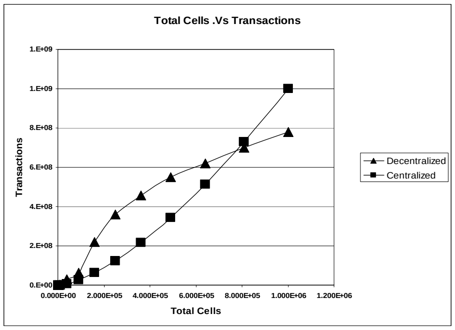

Figure 7: Number of transactions required for centralized and decentralized methods

A greater advantage of our decentralized approach is in the distribution of energy expenditure. The "funnel effect" of multi-hop routing required for the centralized approach described above will deplete cells required for routing much faster as their distance to the sink decreases. In our decentralized approach, the workload is not only evenly distributed, but the workload required of an individual mote actually decreases as the grid size increases, as shown in Figure 8.

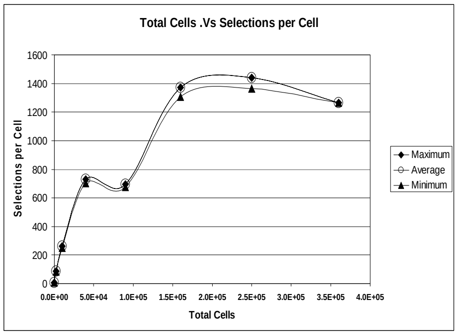

Figure 8: *Illustrating the minimum, average, and maximum number of cell selections in decentralized approach*

As stated, some applications will require a close agreement of common value, while others may tolerate a much larger divergence. We call the latter case one of "good enough computing" and show that, in such cases, a distributed consensus on a common value range can occur quickly with relatively few transactions. As an example, a 200 x 200 grid begins with a color distribution depicted in Figure 9. All 256 colors are represented in a fairly even distribution with a standard deviation of 73.74, 49.6% of colors above the tolerance, and 49.95% of colors below the tolerance.

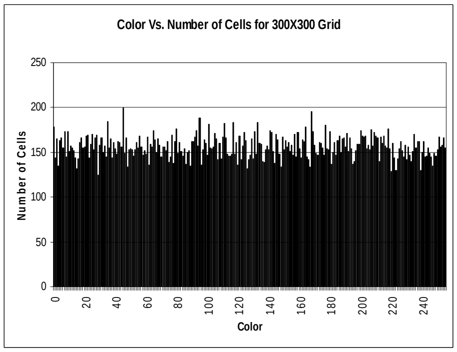

Figure 9. Initial even distribution of colors

However, as shown by the visual display of the simulation, the algorithm comes close to the solution very quickly. The simulation takes 271 time steps to come to a solution where all cells are within the specified tolerance of 0.5. Figure 10 shows that the distribution after only 1 time step (i.e. each cell has been *selected* once). The number of colors represented has been reduced to 55. The standard deviation has reduced to 6.26. While the majority of colors are still outside of the tolerance, the average offset below the tolerance is only 4.9 and the average offset above the tolerance is only 4.7

Finally, Figure 11 shows that, like the total transactions required for a solution, solutions with a large percentage of cells within the tolerance also have a slope that asymptotically approaches zero as the grid size increases.

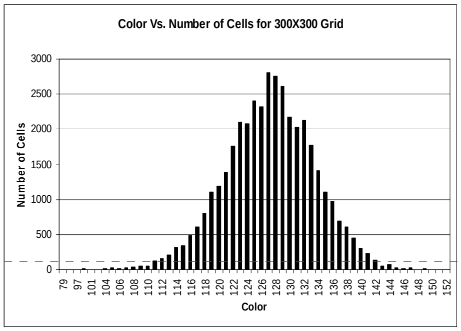

Figure 10. Color Distribution after one time step

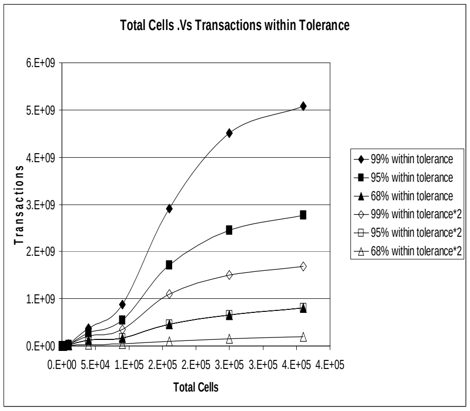

Figure 11. Transactions required for partial solutions

5. Concluding Remarks

The full potential of sensor networks for new kinds of science and business is not being met by current

Comment [CSD4]: Missing a "," here

Comment [CSD3]: Missing a "t" here

practical implementations. Most implementations use relatively small sensor motes in a multi-hop, wireless network. The small size and relatively inexpensive cost of the motes together with wireless networking allow networks to be installed quickly at low cost with little intrusion in the infrastructure of existing structures or the ecosystem of living organisms. However, these implementations use a small number (< 100) of motes operating under centralized control. All do little more than collect a few data and transmit this data back to the central controller. Few if any, decisions are made by the motes.

Most of the techniques being developed for sensor networks work for this environment, but may not scale to massive number of cooperating motes. Transactions required for centralized control, by definition, scale at an exponential rate. If every mote must by queried and/or instructed by a central authority, then, as the size of the network scales exponentially, so must the number of required transactions.

We see the full potential of sensor networks only reached when there are massive numbers of heterogeneous motes acting asynchronously and autonomously, yet cooperating in a way that their local actions, based on local information, combine to effect a functional and sustainable network interacting with the environment. This is how living systems have evolved so successfully. Individual organisms operate by a combination of innate rules (i.e. genetics) and learned behavior in a local niche. The combination of the actions of the individual organisms results in a multifunctional, sustainable ecosystem.

We also believe such a sustainable, functional "ecosystem" of sensor networks does not require enough computation to be 100% correct all of the time. Living systems do not require this for functionality or sustainability. As an example, it would be desirable for human societies to maintain a zero murder rate. Yet, few human societies have ever maintained for long periods of time the total absence of murder. Is it possible for a human society to have zero murders indefinitely? Yes, it is possible, but the resources and the interruption to normal functions required to achieve such a goal are unacceptable. So, societies operate with a "tolerable" level of murder among them. When the murder rate increases to an "intolerable" level, public outcry results in changes that reduce that level. But it rarely reaches zero for long periods of time. Yet the society functions and sustains.

We have demonstrated a function completed by a communal sensor network: autonomous motes functioning asynchronously cooperative to achieve a common goal. The function is carried out without centralized control and without any mote needing to know all information known within the society. We have also shown that the goal can be closely approached with few costs in time and resources, while a final answer may be too costly. We argue that future systems should be designed, wherever possible for such "good enough" computing.

Comment [CSD5]: Missing a "s" here

We will continue our efforts to evolve more sophisticated functionality in large sensor networks with the common goals of:

- No requirement for centralized control.
- Scalability to massive sized sensor networks.
- Design for "good enough" computing.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, Wireless sensor networks: A survey, *Computer Networks*, 38(4), 2002, 393-422.

[2] I. Chatzigiannakis and S. Nikoletseas, A sleep-awake protocol for information propagation in smart dust networks, *Proc. IEEE International Parallel and Distributed Processing Symposium*, (IPDPS'03), Nice, France, April 2003.

[3] D. Culler, D. Estrin and M. Srivastava, Overview of sensor networks, *IEEE Computer*, 37(8), 2004, 41-49.

[4] K. A. Delin and S. P. Jackson, The sensor web: a new instrument concept, *Proc. SPIE Symposium on Integrated Optics*, San Jose, California, January 2001.

[5] D. M. Doolin, S. D. Glaser and N. Sitar. Software Architecture for GPS-enabled Wildfire Sensorboard. *TinyOS Technology Exchange*, February 26, 2004, University of California, Berkeley CA.

[6] D. M. Doolin and N. Sitar, Wireless sensors for wild re monitoring *Proc. SPIE Symposium on Smart Structures & Materials* (NDE 2005), San Diego, California, March 6-10, 2005

[7] D. Estrin, D. Culler, K. Pister and G. Sukhatme, Instrumenting the physical world with pervasive networks, *Pervasive Computing,* 1(1), 2002, 59-69.

[8] D. Estrin, R. Govindan, J. Heidemann and S. Kumar, Next century challenges: Scalable coordination in sensor networks, *Proc. MOBICOM*, Seattle, WA, August 1999.

[9] B. Hemingway, W. Brunette, T. Anderl and G. Boriello, The flock: Mote sensors sing in undergraduate curriculum, *IEEE Computer*, 37(8), 2004, 72-78.

[10] Intel http://www.intel.com/technology/techresearch/research/rs01031.htm, accessed April 17, 2005.

[11] J. M. Kahn, R. H. Katz, and K. S. J. Pister, Next century challenges: Mobile support for Smart Dust, *Proc. ACM MOBICOM*, Seattle, WA, August 1999, 271--278.

[12] Lammers

[13] K. Martinez, J. K. Hart and R. Ong, Environmental sensor networks, *IEEE Computer*, 37(8), 2004, 50-56.

[14] National Research Council, *Embedded Everywhere*, National Academy Press, 2001.

[15] S. Olariu, A. Wadaa, L. Wilson and M. Eltoweissy, Wireless sensor networks: leveraging the virtual infrastructure, *IEEE Network*, 18(4), 204, 51-56.

[16] S. Olariu and Q. Xu, A simple self-organization protocol for massively deployed sensor networks, *Computer Communications*, to appear, 2005.

[17] S. Park, I. Locher, A. Savvides, M. B. Srivastava, A. Chen, R. Muntz and S. Yue, Design of a wearable sensor badge for smart kindergarten, *Proc. 6th International Symposium on Wearable Computers*, Seattle, WA, October, 2002.

[18] J. Polastre, R. Szewcyk, A. Mainwaring, D. Culler and J. Anderson, Analysis of wireless sensor networks for habitat monitoring, in *Wireless Sensor Networks*, Raghavendra, Sivalingam, and Znati, Eds., Kluwer Academic, 2004, 399-423.

[19] K. Ryokai and J. Cassell, StoryMat: A play space for collaborative storytelling, *Proc. CHI'99*, October 1999.

[20] P. Saffo, Sensors, the next wave of innovation, *Communications of the ACM*, 40(2), 1997, 93-97.

[21] K. Sohrabi, J. Gao, V. Ailawadhi and G. Pottie, Protocols for self-organization of a wireless sensor network, *IEEE Personal Communications*, 7(5), 2000, 16-27.

[22] K. Sohrabi, W. Merrill, J. Elson, L. Girod, F. Newberg and W. Kaiser, Methods for scalable self-assembly of ad hoc wireless sensor networks, *IEEE Transactions on Mobile Computing*, 3(4), 2004, 317-331.

[23] M. Srivastava, R. Muntz and M. Potkonjak, Smart Kindergarten: Sensor-based wireless networks for smart developmental problem-solving environments, *Proc. ACM MOBICOM*, Rome, Italy, July 2001.

[24] R. Szewczyk, J. Polastre, A. Mainwaring, J. Anderson and D. Culler, An analysis of a large scale habitat monitoring application, *Proc. 2nd ACM Conference on Embedded Networked Sensor Systems, Nov. 2004*

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

[26] B. Warneke, M. Last, B. Leibowitz and K. Pister, SmartDust: communicating with a cubic-millimeter computer *IEEE Computer*, 34(1), 2001, 44-55.

[27] Xbow Technologies, http://www.xbow.com/, accessed April 17, 2005.

[28] V. V. Zhirnov and D. J. C. Herr, New frontiers: self-assembly and nano-electronics, *IEEE Computer*, 34(1), 2001, 34-43.

[29] http://robotics.eecs.berkeley.edu/~pister/29Palms0103/ , accessed April 17, 2005.