Communal Cooperation in Sensor Networks for Situation Management

Kennie H. Jones, Kenneth N. Lodding
National Aeronautics and Space Administration
Hampton, VA
{k.h.jones,kenneth.n.lodding}@nasa.gov

Stephan Olariu, Larry Wilson
Computer Science Department
Old Dominion University
Norfolk, VA
{olariu,wilson}@cs.odu.edu

Chunsheng Xin
Computer Science Department
Norfolk State University
Norfolk, VA
cxin@nsu.edu

Abstract* – Situation management is a rapidly evolving science where managed sources are processed as real-time streams of events and fused in a way that maximizes comprehension, thus enabling better decisions for action. Sensor networks provide a new technology that promises ubiquitous input and action throughout an environment, which can substantially improve information available to the process. Here we describe a NASA program that requires improvements in sensor networks and situation management. We present an approach for massively deployed sensor networks that does not rely on centralized control but is founded in lessons learned from the way biological ecosystems are organized. In this approach, fully distributed data aggregation and integration can be performed in a scalable fashion where individual motes operate based on local information, making local decisions that achieve globally-meaningful results. This exemplifies the robust, fault-tolerant infrastructure required for successful situation management systems.

Keywords: Sensor networks, massive deployment, ubiquitous input, local control, decentralized action.

1 Introduction

Situation management is a rapidly evolving science supported largely by military requirements. Military conflicts are increasing in mobility, velocity, complexity, and the dynamic nature of their situations. More effective methods are needed for situation monitoring, awareness, and control to provide command options, predict probable situation outcomes, and analyze potential threats and vulnerabilities. However, such requirements are not limited to military applications. Indeed, any activity that requires correct, decisive action in a timely manner including homeland security, emergency/crisis management, manufacturing processes, financial management, medical, among many others, can benefit from the research in situation management [1].

The National Aeronautics and Space Administration (NASA) is researching how the techniques under development in situation management can contribute to future missions in Earth, planetary and space science, mission control, and vehicle/equipment health management.

Jakobson et al. [1] describe the critical aspects of situation management as "managing and controlling sources of information, processing real-time or near real-time streams of events, representing and integrating lower level events and higher level concepts, multi-source information fusion, information presentation that maximizes human comprehension, and reasoning about what is happening and what is important." They define a situation as a "large number of dynamic objects that change state in time and space and engage each other into complex spatio-temporal relationships". The state of an object is expressed as a set of parameter values. The situation is a state of one or more objects at an assigned time. Therefore, two situations are distinguished by both the state of the objects and the time. Situations change as states and/or time changes. [2]. This presents major problems for managing situations:

- *Information freshness*: If information is collected and time passes before a decision can be made using that information, the decision may be incorrect.
- *Information quality*: Information may be redundant, incomplete, or irrelevant.
- *Information overload or starvation*: Information may come in high volumes at times; little to none at other times.
- *Information fidelity*: Analysis of many situations requires massive inputs that can only be acquired in situ.

No new technology shows more promise for meeting these demands than sensor networks. Indeed, the small size and low cost of individual sensors make these networks ideally suited for massive, non-intrusive, and non-obtrusive deployment which can be critical for many situation management applications. However, the implementation of sensor networks simply as sources of input or as effectors will not solve, but actually contribute to, the problems listed above. Additionally, Jakobson *et al.* [1], list new problems posed by massive deployment of sensors:

- Configuration of the network changes frequently.
- Each node has limited computation, memory, power, and communication capabilities.
- There can be a high rate of node and link failure.
- Communication is limited by internode communication through multihop broadcasting.
- Functionality is enhanced by or requires an optimal spatial distribution.

- Nodes are limited in situation recognition; the final operational situation must be compiled from fusion of input from a large numbers of nodes.

This paper reports on our continuing work [3,4,5] in which we look at sensor networks in a novel way, motivated by our belief that in order to scale to massive deployment, sensor networks can benefit from lessons learned from the way biological ecosystems are organized. Indeed, in the presence of a massive deployment, sensor networks must behave as a community of organisms, where individual sensor nodes, or *motes*, operate asynchronously and autonomously in parallel. We focus on fundamental characteristics of future sensor networks that are not demonstrated by current implementations, yet are imperative for optimal use in situation management. More specifically, we demonstrate that in such a model, fully distributed data aggregation and integration can be performed in a scalable fashion in massively deployed sensor networks, where individual motes operate based on local information, making local decisions that are aggregated across the network to achieve globally-meaningful effects.

This approach mitigates many of the problems identified for massively deployed networks. Furthermore, this approach, with the benefit of rapidly increasing computational, power, and communication capabilities of motes, will mitigate many of the problems identified for managing situations by allowing much of the decision making process to occur closer to the source of information, improving the freshness of information and decisions.

It is not our claim that the examples presented here directly apply to any problem of situation management, but rather that the application of sensor networks to situation management requirements may benefit from our alternate approach. Particularly in adversarial scenarios, centralized control of *in situ* networks, with its complex infrastructure and single points of failure, may be quickly defeated. Our decentralized approach is much more robust and fault tolerant and would, therefore, be much more difficult to disable.

In Section 2, we describe a NASA program to develop technologies for aircraft health management. This provides an atypical example of a complex situation to be managed, yet has much in common with more conventional applications. In Section 3, we describe current sensor network research and implementations, identifying concerns about their direction for support of situation management. In Section 4, we describe our ecological model for sensor networks and how this approach leads to autonomous communities of motes that operate locally to support global goals. In Section 5 we discuss the general problem of data aggregation and the problems with a centralized model. In Section 6, we present our model for distributed averaging. In Section 7, we present our conclusions.

2 Aircraft Health Management

NASA's Aviation Safety Program within its Aeronautics Research Mission Directorate is initiating a sub-program, Integrated Vehicle Health Management (IVHM) with the goals of improving safety, reducing costs, and improving performance in every aircraft class.

Concerning safety, an examination of recent aircraft incidents and accidents reveals the need for such an effort. The Commercial Aviation Safety Team (CAST) [6,7] identified system/component failure or malfunction as the third leading cause of crashes in examining commercial jet airplane accidents worldwide from 1987 through 2004. They found other leading causes to be fire, ice, fuel, wind shear, and lightning. These classes of accidents were identified as causing 24% of the total of all accidents. Loss of control caused 26%, but hardware and software failures are often contributing factors in loss-of-control accidents.

Examining data from US-registered transport aircraft accidents from 1980 to 2001, the National Transportation Safety Board (NTSB) [8] found 52% of hardware-induced accidents were aircraft systems related, 36% were caused by propulsion system components, and 10% were caused by failures in the airframe. Of these, approximately one third each was due to problems with landing gear, turbine/turboprop engines, and flight control systems.

The Federal Aviation Administration (FAA) [9] examined 40,964 incidents involving US airplanes from 1998 through 2003. They found that about 67% of incidents were caused by a combination of system and component failure and malfunction, fire/smoke, and power loss.

CAST concluded that not all incidents and accidents were directly due to system failure but resulted from the failure of flight crews to:

- correctly interpret, process, and cross-check available, relevant data, assess failure modes and analyze effects;
- maintain aircraft system status awareness;
- understand the impact of inoperative or degraded systems.

Furthermore, sensors and other equipment failed to:

- accumulate and present adequate trend information;
- indicate impact and other damage;
- indicate and prevent icing;
- provide warning of unsafe flight critical systems.

In providing solutions for these problems, more information is not the only requirement. CAST was concerned that warnings and equipment failure announcements presented to the crew should not cause a nuisance that would contribute to crew complacency. They also identified that part of the solution is to provide real time information to ground crews.

In summary, the evidence reveals that a high percentage of aircraft accidents and incidents are caused directly by equipment failure or indirectly by the inability of crew to properly and timely manage situations of equipment failure. IVHM is tasked with developing technology that will mitigate both of these problems. We recognize that advances in both sensor network technology and situation management technologies are necessary to meet the goals of IVHM.

Future aircraft will be designed to sense, control, communicate, and navigate with increasing levels of autonomy. Automatic health monitoring combined with self-healing systems in aircraft will not only improve safety but also performance, reliability, and predictability of service while reducing costs. IVHM is tasked with developing many of the technologies required for such future aircraft. An essential component is improving diagnostic capabilities that form the basis for prognostics in airframe, propulsion, and other aircraft systems. IVMH is investigating fundamental failure physics and associated effects of damage and degradation caused by environmental hazards. The application of situation management techniques using this knowledge steered by information collected real-time on operating characteristics will provide diagnostics that will determine the prognosis that is used for failure mitigation. From a systems perspective, IVHM also addresses challenges in communication and effective architectures to facilitate the integration of IVHM components with each other and with other vehicle systems.

IVHM represents a constrained microcosm of the larger problem domains more generally associated with the traditional study of situation management. The problems associated with providing a successful IVHM environment mirror the more general problems associated with situation management.

At its most fundamental level of implementation, IVHM is the concept of instrumenting an aerospace vehicle with a web of large numbers of sensors to report on the health of its constituent parts. The simplest architecture for such a web would have the multitude of sensors sending their data to a central computer for processing. The single central processor must perform all situation management functions: data reduction, conversion, fusion, and presentation to the pilot or flight engineer. This approach has a number of shortcomings:

- Single point of failure is at the central computer.
- Potentially large amounts of uninteresting data (e.g. current jet exhaust temperature) will unnecessarily burden the central computer.
- Data that is most meaningful when viewed in conjunction with other data, rather than in isolation must be segregated and fused with other information for presentation. This consumes processor cycles and memory that could be used for other tasks.
- Data that is not currently needed, or not important to the current mission, but needs to be made available to a different user at a later time will still flow to the central computer, further absorbing available processor capabilities: cycles and memory.

We provide an alternative model, which is inspired by biological ecosystems. As we will demonstrate, we consider sensor motes as organisms interacting with their environment. They function autonomously, yet cooperate with local neighboring sensor motes. This architecture provides many advantages over the conventional, centralized approach:

- Multiple, independent processors reduce the chance for catastrophic failure.
- Multiple sensors may be logically joined to provide virtual sensors, which offer the potential for inferring data that would be otherwise unavailable.
- Local sensors can hold data for alternate users, such as maintenance workers, without burdening the central processor, or flight crew with unnecessary, yet important detail.
- Local sensor groups can rapidly make decisions in highly dynamic situations, such as flight anomalies, to respond to anomalies with suggested recovery or mitigation actions.

We recognize that IVHM is more tightly constrained than larger, situation management environments, such as in battle management. However, the benefits of the ecological approach, which are clear for IVHM, seem to us to be appropriate for large-scale deployments of sensors and processors in an isomorphic manner. Distributing decision making to the proper level in the network will be the real challenge.

Sensor networks may fulfill many of the requirements for IVHM and other situation management applications, but current research in sensor networks is not moving in a direction that will facilitate practical implementations. In the next section, we describe these limitations.

3 Examples of sensor networks

The recent flurry of research in sensor networks may be credited to the DARPA-sponsored SmartDust program whose goal was to make machines with self-powered sensing, computing and communication capabilities so small and inexpensive that they could be released into the environment in massive numbers [10]. These devices are called *motes* and serve as nodes in a sensor network [11,12]. As the motes are severely energy-constrained, they cannot transmit over long distances, restricting interaction to their immediate neighborhood.

An examination of current implementations reveals both successes and limitations to the promises of sensor networks. In 2002, a sensor network was implemented on Great Duck Island, Maine [13]. The initial application was to monitor the microclimates of nesting burrows and, by disseminating the data worldwide, to enable researchers anywhere to non-intrusively monitor sensitive wildlife habitats. The sensor motes were placed in the habitat and formed a multi-hop network to pass messages back to a base station. The data was eventually passed by satellite to servers in Berkeley, CA, where it was distributed via the Internet to any interested viewer. The sensors periodically measured environmental factors and relayed the measurements to the base station. The largest deployment had 190 nodes up to 1000 feet from the nearest base station.

The interested reader can examine other implementations [14,15,16]. What was novel about these approaches is the small size of the sensors and their wireless networking allowing inexpensive and unobtrusive installation directly into the environment. However, these demonstrations used between 6 and 800 motes, thus, they do not approach the high fidelity information architecture advertised by proponents of sensor networks. Will the techniques used scale to massive numbers? In these designs, behavior is predetermined, its results collected, and otherwise managed by a central authority.

As current sensor networks are for the most part modeled after conventional networks under centralized control and involve a small number of motes, it is not clear that they provide a credible approximation of the massive deployment envisioned by the proponents of sensor networks [14,17]. 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 are likely to provide realistic models for sensor network design.

4 An ecological model

We think of motes as organisms within a community. At birth (i.e., at deployment time) the motes are endowed with genetic material, containing, among others, an initial state and rules by which they interact with the environment. The state and the rules may change as the motes interact with the environment, reflecting their dynamic adaptation to conditions in their neighborhood. Additionally, the motes may remember and record their interaction with the environment by storing information in their limited on-board memory. Memory and its use to change state or rules are considered learning. Changing state conditions based on learning demonstrates some level of cognition.

One of the goals of this work is to investigate how local decisions based on strictly local information can effect global results. Limiting decisions to localities is important for reasons of scalability and autonomy. Local decisions allow distributed control. In turn, distributed control through local decisions provides a natural redundancy affording fault tolerance – as some motes exhaust their energy budget and expire others will continue to make decisions.

Although genetic algorithms are a popular algorithmic paradigm, they rely in a crucial way on extremely fast computational speed to evaluate many random mutations of some genetic specification. While most of these new combinations will prove useless, or worse, harmful to the objective, the search is for the small percentage of mutations that prove beneficial. Indeed, the limited computational power of the motes would make the use of genetic algorithms prohibitively expensive. Furthermore, we view a mote as an individual organism. Just as with living organisms, successful changes in behavior or other capabilities must be based on experience and learning. Random changes would be highly likely to result in death (i.e., failure) of the mote and catastrophe for the network. Afterwards, there would be no chance to try another mutation.

We use cellular automata as a viable model for massively deployed 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, eight neighbor cells surround each internal cell. Border cells have three or five neighbors. Neighbor cells represent those motes that can receive a transmission from a cell. Thus, the regularity of the grid represents a logical indication of physical proximity. Throughout this work we assume that each sensor has exactly eight neighbors. Visibly, the set of neighbors need not be limited to the eight adjacent cells. Specifying a neighbor radius greater than 1 increases the number of cells that can receive from the transmitting cell. A radius of two would include in addition to the eight adjacent cells, the 16 cells adjacent to these neighbors. One apparent limitation of this model is that the number of neighbors is fixed for a given radius; however, disabling some of the neighbors can change this.

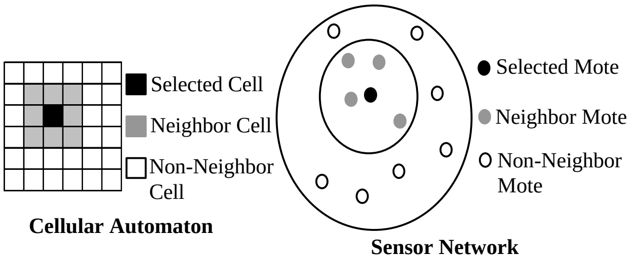

Figure 1: Illustrating the neighborhood of a cell

5 Aggregating sensed data

There are, essentially, two ways in which data sensed by the motes can be aggregated. In a centrally-controlled network, data aggregation and integration is a two-stage process: in the first stage the motes forward the data collected to the sink. Some of the data may be fused en route but the final responsibility for aggregation rests within the sink. In the second stage, the aggregated result is broadcast back to the network. Though straightforward, this method does not scale well [17,18]. By contrast, in a truly distributed system, as is the case in an autonomous sensor network, 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 eventually obtain and maintain the global average by using local data only. As the values change, the process must be repeated. While a number of solutions to the aggregation problem have been proposed in the literature [17,18,19,20,21,22], 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 massively deployed sensor networks.

In the following, we first examine the issues of using a centralized approach for data aggregation in average calculation, and then propose our autonomous and distributed approach. As our cellular automaton is defined such that each cell can only transmit to its immediate neighbor, then in the centralized approach, data collected by the sink from each cell must pass through half the span of the grid on average. It is possible to minimize the number of transactions by aggregating values as they are routed to the sink, but this requires substantial infrastructure and coordination. In the second stage (broadcasting aggregated result), to conserve mote energy, a suggested approach 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 and the transmission from the sink must be directly receivable by all motes (i.e., there can be no blockage). We assume a cost of flooding the computed average to all cells, which increases with the size of the grid. Besides transmission costs, there are additional problems with the centralized approach. Routing tables to reach the sink must be discovered and maintained. Disruption in these routes must be handled to assure messages arrive at the sink. Regardless of how this is done, this method is open to single points of failure.

Recently, it was noticed by Wadaa et al. [23] as well as by other workers that centrally-controlled sensor networks are prone to uneven energy depletion leading to the creation of energy holes in the vicinity of sinks. Specifically, [23] showed that by the time the motes close to the sink have expended all their energy, other motes in the sensor network still have about 80-90% of their original energy budget. This uneven energy depletion creates an energy hole around the sink, severely curtailing network longevity. Figure 2 illustrates the uneven energy depletion problem, where all black cells must pass messages to the sink through a single cell in the top row. In our cellular automaton model, the cells closest to the sink must relay messages from every cell in the grid and their energy budget will decrease rapidly relative to cells further away.

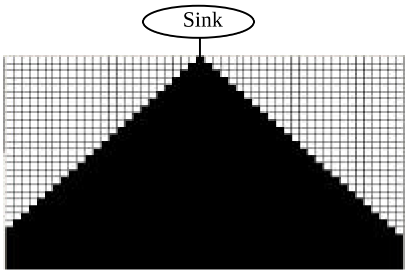

Figure 2: Illustrating the energy-hole problem

Our objective is for the sensor network to function as a community that will come to a consensus on some value across the network without any mote or any central authority having global knowledge of all mote values. The problem is for each mote to obtain and maintain the global value by iteratively using only data that is available locally.

We now give an informal description of our model. Assume a data value on which consensus is to be formed is known by each mote and obtained by some interaction with the environment. Upon deployment, each mote is assigned a *time period* and a *selection* time for action within that time period as genetic material. The time period is divided into one or more slots. The selection time assigned to a mote is one of these slots within the time period (e.g., one mote may be selected at slot 3, another at slot 18, but all motes will be selected at some time during a time period). Because the selection time for each mote is determined randomly, there is no guarantee that two or more motes will never be selected at the same time. Upon deployment, each mote starts its own clock. If two motes are at a neighbor radius greater than 3, they may execute the algorithm simultaneously (i.e., be selected), 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 Media Access Control (MAC) layer protocol is assumed to decide cellular execution order. To simulate this control, all motes (i.e. cells) selected at the same time slot are executed in random order. The order is randomized anew when they are selected next time. Thus, the selection time is fixed at deployment (being part of the genetic material), but the execution order in each time slot is a "function of the environment" and may change. 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 the consensus value, selection time assignment, and execution order in each time slot, we run multiple executions varying all three parameters and average the results.

When a cell reaches its selection time within the time period, it executes the algorithm as follows. If its status is inactive, it does nothing. Otherwise, if its status is active, it begins a series of transactions. A transaction is either a request for information from a neighbor or a specification given to a neighbor. Transactions are significant because they require radio transmissions, typically the most costly activity of a mote. If a mote determines no action is required, it sets its status to inactive and will not participate again until it is reactivated by one of its neighbors. If a mote determines action is required, it will perform the necessary work resulting in some change to itself and/or its neighbors. During this process, if any neighbor is inactive, it may be reactivated as determined by the selected mote. The simulation continues until all cells are set to inactive. Thus, the simulation ends using only local information; no global control is required. As these local neighbor cells act autonomously and asynchronously yet cooperate with each other, act only on local environmental information, and remember information from one action that will affect a future action, we argue that this system demonstrates simple cognition.

6 Computing the global average

In the following, we use our CA to simulate the function of averaging a value across the network. Figure 3 depicts an initial distribution in a cellular automaton with a 30 x 30 grid of cells showing a random distribution of values (for example, assume a color) in the range of [0,255]. The objective is to calculate an average color and set all motes to that value using only local information to determine local actions. The basic operation occurs when a cell is *selected* for action. When each cell is selected, it begins a series of *transactions*:

- It requests (requires transmissions) the color value of each of its neighbors.
- If all are equal, it inactivates itself.

- If not, it calculates an average for the neighborhood and sets itself and all neighbors (requires transmissions) to that value.

As transmissions are the most expensive operations of a mote and the number of transmissions required when a cell is selected are fixed, a cell selection is a measure of workload: the more times a cell is selected, the higher its workload and the more energy it expends. Parameters for the simulation are set such that all cells will be selected once during each time *period*. Figure 4 illustrates the color change after 10 periods. Soon thereafter, the color differences are indistinguishable to the naked eye.

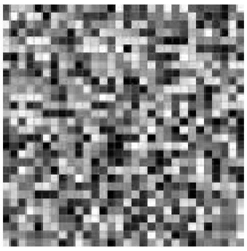

Figure 3: Initial distribution of colors for color averaging

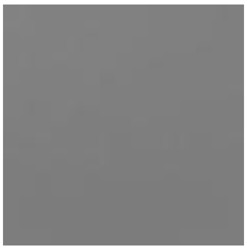

Figure 4: Distribution of colors after 10 time periods

The converged average for the community, the average after all cells are inactivated, 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 following each cell selection is also 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.

A great 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 the energy of cells much faster when their distance to the sink is shorter. In our decentralized approach, the cell selections required is not only evenly distributed, but the cell selections required of an individual mote actually decreases as the grid size increases, as shown in Figure 5. Figure 5 shows that the number of selections per cell increases steadily until the grid size exceeds 400X400 cells but, thereafter, is stable. Thus, for large networks, the cell selections per mote does not increase.

As stated earlier, some applications require a close agreement of common value, while others may tolerate a much larger divergence. We call the latter case "good enough computing" and show that, in such cases, a distributed consensus on a small range of values can occur quickly with relatively few transactions. As an example, a 200 x 200 grid begins with a color distribution depicted in Figure 6. All 256 colors are represented in a fairly even distribution with a standard deviation of 73.74, 49.6% of colors above the average plus a tolerance of 0.5, and 49.95% of colors below the average minus this tolerance.

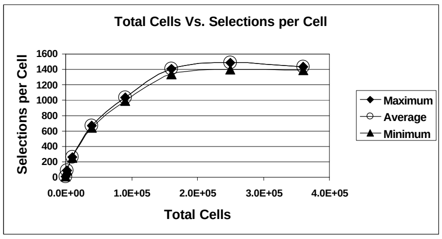

Figure 5: Illustrating the minimum, average, and maximum number of cell selections

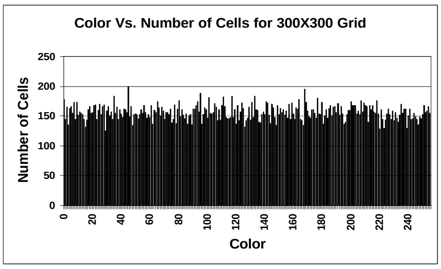

Figure 6: Initial distributions of colors

However, as shown by the visual display of the simulation in Figures 3 and 4, 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. By the 86th time step, there are only 3 colors represented. The standard deviation is within the tolerance of 0.5. No color is more than 0.56 above the average plus the tolerance and no color is more than 0.42 below the average minus the tolerance. Thus the range of colors is substantially reduced long before all colors are within the specified tolerance.

As the grid size grows, the total number of transactions increases with a linear slope as shown by the line with the steepest slope in Figure 7. This is not a problem for scalability as the total energy available also grows linearly with more motes and recall that Figure 5 shows that the number of transactions per mote does not grow with large grid sizes. But the other lines in Figure 7 show that solutions with a large percentage of cells within the tolerance are achieved with far fewer transactions. Note that slopes for these solutions asymptotically slope towards zero as the grid size increases, a desirable property for scalability. Solutions up to 95% within the tolerated average reach that asymptote with relatively few transactions compared with the 100% solution regardless of grid size.

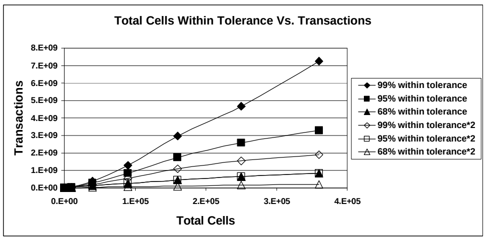

Figure 7: Percentage of cells within tolerance

7 Concluding remarks

The full potential of sensor networks can 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 form a functional and sustainable network interacting with the environment. This is how living systems have evolved so successfully. Individual organisms operate, make decisions, and take actions based on 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.

In this work we have demonstrated a function completed by a sensor network working as a community: autonomous motes functioning asynchronously cooperate 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 community. We have also shown that the goal can be closely approached with few costs in time and resources compared with the much more costly final answer

The ever-increasing technology curve is going to continue to escalate the capabilities available for situation management. These tools cannot be timely managed and analyzed under centralized control. Methods such as we have described here will be necessary for the large systems to operate as a community where many functions can be facilitated by autonomous, asynchronous sub systems.

The IVHM program at NASA exemplifies the complexity encountered when designing the management of many diverse situations in an environment where timely decisions must be made correctly to avert catastrophe. In such an environment, situation management through a centralized controller is impractical. Our communal approach, where decisions are made locally when possible or through cooperation among subsystems when necessary, provides the architecture required of such complex systems.

References

[1] G. Jakobson, L. Lewis, C. Matheus, M. Kokar, and J. Buford, “Overview of Situation Management at SIMA 2005”, retrieved May 9, 2006 from http://vistology.com/papers/SIMA2005.pdf

[2] G. Jakobson, J. Buford, and L. Lewis, “Towards an Architecture for Reasoning about Complex Event-Based Dynamic Situations”, International Workshop on Distributed Event-Based Systems (DEBS '04) at 26th International Conference on Software Engineering (ICSE 2004), Edinburgh, Scotland, UK, May 24-25, 2004.

[3] Jones, K., Lodding, K., Olariu, S., Wilson, L., and Xin, C., "Biology Inspired Approach for Communal Behavior in Sensor Networks", presented at Minitrack on Wireless Sensor Networks and Applications, Proceedings of 39th Hawaii International Conference on System Sciences (HICSS-39), Kauai, Hawaii, January 4-7, 2006.

[4] Jones, K., Lodding, K., Olariu, S., Wilson, L., and Xin, C., "An Ecosystem Model for Massively-Deployed Sensor Networks", Proceedings of 2nd International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Melbourne (ISSNIP 2005), Australia, December 5-8, 2005.

[5] Jones, K., Lodding, K., Olariu, S., Wilson, L., and Xin, C., "Energy Usage in Biomimetic Models for Massively-Deployed Sensor Networks", presented at 1st International Workshop on Mobile Ad-hoc and Ubiquitous Sensor Networks (MAUSN 2005), Proceedings of 3rd International Symposium on Parallel and Distributed Processing and Applications (ISPA'2005), Nanjing, China, November 2-5, 2005.

[6] Commercial Aviation Safety Team (CAST) Safety Plan, October 7, 2004.

[7] Commercial Aviation Safety Team (CAST) Update, February 2006.

[8] NTSB accident database, Retrieved may 15, 2006 from http://www.ntsb.gov/ntsb/query.asp

[9] FAA AIDS database, Retrieved May 15, 2006 from http://www.nasdac.faa.gov/portal/page?_pageid=33,36440&_dad=portal&_schema=PORTAL

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

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

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

[13] A. Mainwaring, J. Polastre, R. Szewczyk, and D. Culler, “Wireless Sensor Networks for Habitat Monitoring”, Intel Research, IRB-TR-02-006, 2002 ACM International Workshop on Wireless Sensor Networks and Applications, June 10, 2002, Retrieved April 5, 2004, from http://www.greatduckisland.net

[14] D. Lammers, “Embedded projects take a share of Intel's research dollars”, EE Times, August 28, 2001. Retrieved April 5, 2004, from http://today.cs.berkeley.edu/800demo/eetimes.html

[15] UCB/MLB 29 Palms UAV-Dropped Sensor Network Demo, 2001, University of California, Berkeley. Retrieved April 5, 2004, from http://robotics.eecs.berkeley.edu/~pister/29Palms0103

[16] M. Chen, C. Majidi, D. Doolin, S. Glaser, and N. Sitar, “Design and construction of a wildfire instrumentation system using networked sensors”, Network Embedded Systems Technology (NEST) Retreat, Oakland California. Retrieved April 5, 2004, from http://firebug.sourceforge.net

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

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

[19] 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.

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

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

[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), pp. 317-331, 2004.

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

[24] S. Ganeriwal, R. Kumar, and M. B. Srivastava, “Timing-sync protocol for sensor networks,” ACM Conference on Embedded Networked Sensor Systems (SENSYS 2003).

[25] MPR/MIB User’s Manual, Crossbow Technology, Inc., Retrieved May 5, 2005, from http://www.xbow.com/Support/Support_pdf_files/MPR-MIB_Series_Users_Manual.pdf

[26] P. Levis, S. Madden, D. Gay, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, “TinyOS: An Operating System for Sensor Networks”, in Ambient Intelligence, Jan Rabaey Ed, 2003.