WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2007; **7**:1–12
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/wcm.499

A brief survey of self-organization in wireless sensor networks

Kevin L. Mills*,†
National Institute of Standards and Technology, Gaithersburg, MD 20899-8920

Summary

Many natural and man-made systems exhibit self-organization, where interactions among components lead to system-wide patterns of behavior. This paper first introduces current, scientific understanding of self-organizing systems and then identifies the main models investigated by computer scientists seeking to apply self-organization to design large, distributed systems. Subsequently, the paper surveys research that uses models of self-organization in wireless sensor networks to provide a variety of functions: sharing processing and communication capacity; forming and maintaining structures; conserving power; synchronizing time; configuring software components; adapting behavior associated with routing, with disseminating and querying for information, and with allocating tasks; and providing resilience by repairing faults and resisting attacks. The paper closes with a summary of open issues that must be addressed before self-organization can be applied routinely during design and deployment of senor networks and other distributed, computer systems. Copyright © 2007 John Wiley & Sons, Ltd.

KEY WORDS: distributed systems; self-organization; sensor networks; wireless networks

1. Introduction

Scientists and engineers envision deploying wireless sensors that can form networks to make and convey measurements for many applications: measuring ocean temperatures and currents, analyzing moisture content in soils, gauging ground motions, assessing sunlight in forests, and monitoring stresses in structural supports of large buildings. What might such applications mean for the way we design, deploy, and manage wireless networks? The number of devices, communications channels, and data transmissions will become too large, varying, and uncertain to be deployed and managed with the costly techniques in use today. Instead, wireless networks must become adept at *self-organization*—allowing devices to reconnoiter their surroundings, cooperate to form topologies, and monitor and adapt to environmental changes, all without human intervention. Self-organization applied to wireless networks is not a new concept. Interested readers should consult a 1986 survey by Robertazzi and Sarachik [1]. While many problems identified in the earlier survey still exist, the nature of wireless networks has become more tangible and pervasive. The current survey focuses on self-organization in sensor networks, which did not exist in 1986.

The paper begins by considering self-organization from two views: natural phenomenon and design

*Correspondence to: Kevin L. Mills, National Institute of Standards and Technology, 100 Bureau Drive Stop 8920, Bldg. 222/B218, Gaithersburg, MD 20899-8920, U.S.A.
†E-mail: kmills@nist.gov

strategy. Self-organization is a natural phenomenon of distributed systems, where components interact on a microscopic level leading to global behaviors that emerge on a macroscopic level. Such emergent behaviors are unintended and thus may be undesirable. For example, unintended self-organizing phenomena have been observed in the Internet [2], cellular wireless networks [3], and computing grids [4]. As a design strategy, system components may be endowed with local rules intended to yield desired global behaviors. The paper identifies selected approaches to stimulate intentional self-organization for allocating spectrum, bandwidth, and processing capacity; for forming structures, disseminating information, and organizing tasks; for configuring software, synchronizing time, and conserving power; and for repairing faults and resisting attacks. The paper also presents open questions to stimulate further research into self-organization as a design strategy.

2. Self-Organization as Natural Phenomenon

A system with many simple components can exhibit behaviors of the whole that appear more organized than behaviors of the individual components [5]. These so-called emergent behaviors arise naturally through a process of self-organization, which appears in complex natural and man-made systems (e.g., biological organisms, ecosystems, food webs, geological systems, metabolic networks, transportation networks, and stock markets [6–12]). Complex systems encompass jumbles of positive and negative feedback loops that cascade the effects of changes in each component through an increasing number of interconnected components. Through such interactions, system state tends toward some coherent pattern. This is the essence of self-organization: patterns arise from many interactions spread over space and time. Such patterns are known as emergent properties because they have no meaning for individual components. For example, gas (a collection of molecules) exhibits both temperature and pressure, which measure strength of interactions among molecules.

What emergent properties might be observed? One possibility is equilibrium, where system state reaches some fixed point. Another possibility is oscillation, where system state cycles repeatedly through the same series of points. A third possibility is chaos, where system state wanders forever through a non-repeating set of points. Some scientists have noted a tendency for equilibrium states in certain systems to exhibit a delicate balance, referred to as self-organized criticality [13], where system state can be driven easily out of equilibrium. Some natural systems exhibit punctuated equilibria [14], where system state moves through occasional periods of turbulence with a frequency inversely related to magnitude. Movements among emergent patterns are known as phase transitions [15].

Investigations of natural and man-made dynamic systems reveal that phase transitions occur quickly after reaching some threshold. For example, Kuramoto [16] shows a system of coupled oscillators remains desynchronized until coupling strength reaches a critical threshold after which synchronization advances in stages. Floyd and Jacobson [2] observe network traffic that becomes synchronized only when the number of sources exceeds a transition threshold. Roli and Zambonelli [17] report that a dissipative cellular automaton exhibits macroscopic spatial structure as soon as external stimulation reaches a threshold value and exhibits a chaotic pattern once external stimulation passes a higher threshold. In a study of random graphs, Erdös and Rényi [18] identified a phase transition occurs when the number of randomly placed links reaches the number of nodes, after which a graph becomes fully connected.

Why do so many natural systems exhibit self-organizing properties? What benefits does self-organization convey? Adaptability is a key benefit in both short and long terms. Short-term flexibility allows maintenance of stable operating states under varying environmental conditions [19]. Long-term evolution enables development of new equilibrium states in response to shifting environmental patterns. Evolution also increases problem-solving range [20]. Evolution implies memory, which implies learning; thus, self-organizing systems can solve problems that are unsolvable using other techniques [5]. Even for problems with known solutions, self-organizing systems can devise innovative approaches that might otherwise go undiscovered [21]. Further, self-organizing systems exhibit the principle of least action, which tends to minimize distance to an optimal (stable) state [5], and thus prove efficient at solving difficult optimization problems. Many self-organizing systems also exhibit resilience: both robustness and survivability. By adapting to changing conditions, self-organizing systems can overcome failure of individual components [22]. Over the long run, a self-organizing system can continue to pursue system-wide goals even beyond the lifetime of all current, system components. Scalability is another benefit [23]. Self-organizing systems may grow without bound because complete

DOI: 10.1002/wcm

information need not be disseminated throughout the system and processed by all components.

Detecting or measuring presence or degree of self-organization remains subject of significant research. Systems may self-organize in space, in time, and in spatiotemporal combinations. Generally, self-organization leads to increased correlation along a dimension of measurement—implying self-similarity. For example, self-organizing systems often organize hierarchically, where statistical characterization of spatial organization at all layers appears quite similar [9]. Self-organizing systems can also show correlations in time, such that scaled time windows yield similar statistical characteristics [24]. Physicists often 'transform the autocorrelation function into the Fourier spectrum. A power-law decay for the correlations as a function of time translates into a power-law decay of the spectrum as a function of frequency... also called $1/f$ noise' [25]. Fourier transforms can reveal oscillations by identifying specific dominant frequencies [26]. Wavelet transforms may show correlations among spatial or temporal scales [27]. Other measures of self-organization have been proposed. For example, Oprisan defines [28] three measures, angular momentum, contrast, and correlation, to describe the level of aggregation within a spatial extent. Some researchers [29] leverage thermodynamics, using decrease in entropy to indicate increased order arising from self-organization. Other researchers [30] apply statistical complexity to measure changes in system order.

3. Self-Organization as Design Strategy

Noting the pervasive presence and potential benefits of self-organization in natural systems, numerous researchers investigate how models of self-organization can be applied to design large, distributed systems. This section introduces some representative models.

3.1. Biological Models

Scientists have uncovered evidence of self-organization in biological processes, inspiring computer-science researchers to investigate their application to system design. For example, during biological reproduction embryos form as a collection of homogeneous cells and then develop into a complex organism with specialized functions. This process of multi-cellular *embryogenesis* uses self-coordination to enable cells to differentiate function. Researchers at MIT [31] are investigating use of such techniques to enable substrates of homogeneous computers to self-organize into differentiated structure and function. NASA researchers [32] are also investigating embryogenesis as a means to adapt undifferentiated processors on deep-space probes in order to permit changes in spacecraft function during missions of long duration. Nagpal [33], a researcher at Harvard, has proposed a set of primitives, based on mechanisms from embryogenesis, which engineers could use to cause homogeneous processes to self-organize into desired functionality and structure. Other researchers aim to exploit the process that allows an undifferentiated collection of neurons throughout the brain to self-organize into specialized pattern recognition networks that can distinguish and classify sensory inputs. For example, Kohonen [34] has developed an algorithm for *self-organizing maps* (SOMs) that transform a multidimensional space of inputs into a lower dimensional lattice of neurons such that topological relationships among the input space are reflected into the constructed neural network. Researchers apply [35] SOMs to a range of system-engineering challenges. Other human biological functions also inspire design models. Hofmeyr and Forrest [36], for example, define an *artificial immune system* and describe its application to intrusion detection in computer networks. IBM [37] has founded an entire research program, autonomic computing, based on modeling self-managing systems after concepts inherent in the human nervous system. *Regulatory genetic systems* in living cells have been modeled as *NK* Boolean networks [6] (of *N* logic elements each with *K* inputs and one output) or probabilistic Boolean networks [38] that self-organize into attractors comprising cyclic sequences of states. *NK* networks have been applied to model structural dynamics in industrial networks [39]. Evolutionary processes have also inspired computer scientists to apply *natural selection* to evolve solutions [40] to a wide range of problems that are difficult to solve using other techniques.

3.2. Social Models

Recently, scientists have begun to study the organization and function of *swarms*, such as birds, insects, viruses, and molds, which exhibit self-organization, arising from the ability of swarm members to exchange information, either directly or indirectly. Direct information exchange (e.g., through visual or auditory channels) implies a synchrony in time. For example, birds can maneuver as flocks [41] if each bird follows three general rules: move toward the average heading of other birds, maneuver toward the average position of

DOI: 10.1002/wcm

other birds and avoid coming too close to other birds. Similarly, large groups of fireflies can synchronize their flashing, using visual cues and internal timing mechanisms [42]. Indirect information exchange, or *stigmergy* [43], implies that swarm members are mobile; thus, information can be deposited in space to be encountered by members arriving later. For example, ants deposit a chemical (pheromone) to attract other ants, which strengthen the scent attracting additional ants. This behavior helps ants to retrieve food and return it to the nest. As the food supply becomes exhausted, ant visits on a trail diminish, the scent decays, and the trail is eventually abandoned. Similar behavior has been observed in slime molds [44], which normally move through dirt as individual single-celled organisms until environment conditions deteriorate. A worsening environment leads cells to emit a chemical that guides collective movement so that large mold structures emerge, allowing cells to survive until the environment improves.

3.3. Economic Models

Economies are self-organizing systems where producers and consumers interact through markets to set prices under which to exchange goods and services. While most readers probably associate economics with capitalism, researchers are investigating how to design information systems based on numerous economic models [45–50], including self-interest, socialism, communism, altruism, game theory, and catallaxy.

3.4. Other Models

A number of self-organizing models from physics and chemistry have been applied [51–55] to design computer, communications, and information systems. Such models include electromagnetism (attraction–repulsion), thermodynamics (entropy reduction), molecular equilibrium (minimizing energy or repulsion force), diffusion (chemical gradients), and phase-transition resistance (stabilizing system state far from phase-transition regions).

4. Applying Self-Organization in Wireless Networks

Self-organizing mechanisms could pay dividends in almost any kind of wireless network. For example, self-organization might allow adaptation to changing user density and traffic patterns in fixed wireless networks, where only users move. Self-organization could help reconfigure topologies as nodes move in and out of range in mobile ad hoc networks, where all nodes may move. Self-organization could form an initial topology among large numbers of sensor nodes dropped across a geographic area, and then adjust the topology as sensors exhaust power and replacement sensors are injected.

This paper surveys the use of self-organization in wireless networks to accomplish specific functions: sharing resources (processing and communication capacity); forming and maintaining structures; adapting behavior associated with routing, with disseminating and querying for information and with assigning tasks and configuring software components; managing resources (synchronizing time and conserving power); and providing resilience by repairing faults and resisting attacks. These functions reflect increasing levels of abstraction: sharing physical resources, forming collectives, shaping collective behavior, managing collective resources, and ensuring collective survival under duress. Vast research literature exists on self-organization in wireless and sensor networks, with particular concentration on topics such as topology formation and maintenance. Few examples could be included in this brief survey. References were selected to achieve wide coverage of functions and broad representation across various models of self-organization.

4.1. Resource Sharing

Nodes in a wireless network must share a number of resources, such as electromagnetic spectrum, transmission bandwidth, and processing capacity. The task becomes difficult when the number of nodes and traffic demands are unknown or fluctuate. Self-organization can be used to discover participants and demands, to determine how best to allocate resources, to monitor changes, and to reallocate resources as needed.

4.1.1. Processing

Most sensor networks require nodes not only to act as data sources and sinks but also as relays that forward packets among neighboring nodes. Assuming nodes have finite power, tradeoffs arise between network throughput (which should be as high as possible) and lifetime (which should be as long as possible). Complete cooperation with forwarding minimizes a node's lifetime, while completely uncooperative behavior drives throughput to zero. Srinivasan *et al.*

DOI: 10.1002/wcm

[56] describe a *game-theoretic algorithm*, based on Generous Tit-For-Tat, designed to drive a system of nodes to Nash equilibrium where each node achieves the best possible tradeoff between throughput and lifetime. Assuming each node understands its maximum forwarding rate and maintains a history of experiences regarding the rate at which its forwarding requests are honored, a node will reject a forwarding request beyond its maximum rate (outside healthy operating bounds) or if the node is forwarding more packets than another node is forwarding for it. This latter condition allows a small amount of excess forwarding—representing the generous portion of the algorithm.

Typical energy-aware routing schemes maintain a list of possible routes and then forward packets with a uniform probability among them. Willig and colleagues [57] observe that sensor networks may contain nodes with a range of capabilities, including differences in available power, and argue that network lifetime could be increased if more-capable nodes handled more work. To enable asymmetric load assignment, Willig *et al*. define an *altruistic* (friendly neighbor) approach, where nodes periodically announce their capabilities, location, and address, along with a time for which a node is willing to forward packets. The assumption is that only nodes with rich power sources would announce. The cost of forwarding packets over self-declared altruistic nodes is then discounted, thus increasing the probability of relaying packets through those nodes. Simulation results show that this altruistic approach yields significant improvement in both network lifetime and response time when compared to a typical energy-aware routing scheme.

4.1.2. Communication channel

Kompella and Snoeren [58] present a distributed algorithm that allows individual sensors sharing a channel to independently adjust transmit power and rate to conserve energy without significantly degrading channel capacity or fairness on oversubscribed channels. The authors observe that when channel load is low then messages can be sent more slowly (i.e., at lower power) without building up an excessive queue, while high load requires messages to be sent more quickly to avoid excessive queuing. They define a self-organizing approach were nodes sharing a channel snoop on transmissions and use measured transmission rates to estimate the message load at each node. Once each node has sent at least one message, then all nodes can converge to a similar estimate of the channel load and each can then independently adjust transmission speed to ensure that all queued packets get an equal share of the channel.

Duque *et al*. [59] describe an approach, based on *self-organizing maps*, to allocate spectrum to connections in a dynamically changing cellular network. Given a set of network measurements (e.g., cell interference and channel compatibility), Kohonen's algorithm [34] is used to construct a mapping into equivalence classes where all radio relays in a partition have similar interference situations. Subsequently, an iterative algorithm searches for variations in channel assignments that optimize network performance for a given interference situation. The maps are then distributed to radio relays, where continuous monitoring allows relays to switch channel assignments to match changes in the interference situation.

Ho *et al*. [29] describe a self-organizing algorithm that allows radio relays in a cellular network to create and dynamically adjust cell sizes (i.e., transmission ranges) to maintain maximum coverage with minimum interference. Each relay will periodically listen for neighboring relays. Hearing a neighbor arrive or signoff stimulates a relay to conduct an expanding-ring search to calculate its distance from all reachable relays. Subsequently, the relay computes and distributes a new cell size, then waits for the next listening period. Ho uses an *entropy-based complexity metric* to reveal critical characteristics about the delay between listening periods. Below a threshold, the network never achieves full coverage. Above the threshold, the probability of achieving full coverage increases with delay. Beyond a second threshold, the network always achieves full coverage.

4.2. Structure Formation and Maintenance

Typically, sensor networks are deployed incrementally without central planning and must adapt to changes in node density, while simultaneously minimizing power consumption and meeting performance objectives. Designing and deploying static topologies cannot satisfy this challenging combination of requirements. For this reason, numerous researchers [e.g., 60–62] investigate approaches that allow nodes to self-organize into efficient, clustered topologies and to maintain essential cluster properties in response to changing node populations. In selected cases, networks contain mobile sensors, which researchers consider how best to position.

DOI: 10.1002/wcm

4.2.1. Sensor placement

Some sensors are mounted on mobile platforms, which permit the option to enhance sensor coverage after initial deployment. Wong and colleagues [51] propose a technique that allows mobile sensors to reposition themselves based on computing virtual *attraction and repulsion forces* exerted by other sensors and obstacles. To conserve energy, sensor movements are bounded within a limited range. The algorithm uses only local information to reposition sensors to improve coverage with minimum movement. Force between nodes is relative to distance; nodes that appear too close exert repulsion and nodes that appear too distant exert attraction. A node computes the relative influence from all surrounding forces in order to select a new position. To limit movement, nodes engage in an exponential back-off procedure to determine the order in which each node updates its position.

Low *et al*. [63] consider problems arising when mobile sensors with limited sensory range are deployed sparsely relative to territory and without certain knowledge regarding location of potential targets. Some means must be found to direct sensor movement in order to provide adequate coverage of targets while limiting interference from an excess of sensors within the same area. Low proposes an *ant-based, task-allocation scheme* that enables mobile sensors to organize into coalitions matched to the distribution of targets across areas. Each robot measures two average delays, one for encounters with other robots and one for encounters with targets, and computes their ratio, which represents task demand as observed by the robot. Robots within the same vicinity will periodically exchange ratios, along with the number of targets currently under observation. Using this information each robot conducts a probabilistic trial to determine its dominance over other robots. Winning such trials enhances a robot's tendency to remain in the area, while losing enhances tendency to leave. Periodically, robots conduct another probabilistic trial (which considers distances between areas) to determine whether to leave the current area.

4.2.2. Server placement

Parunak and Brueckner [64] consider server placement and selection in networks where power-constrained or mobile nodes cause continuous topology changes. They propose an approach, based on *stimergic learning*, that allows a server population to maintain the minimum necessary number of nodes at locations appropriate to serve a client population and that allows clients to learn where to direct service requests. Servers implement a *reinforcement-learning* algorithm where they extend their lifetime based on the number of client transactions arriving within a measurement interval. Clients share with direct neighbors a history of interactions with servers. Histories, reinforced based on positive and negative server interactions, decay over time in order to give more weight to recent interactions. Clients eliminate memory of any server that reaches a threshold of negative performance. Simulation results show that stimergic learning leads to significant power conservation without significantly reducing performance.

4.3. Behavior Shaping

Once deployed, sensor networks perform a range of functions: some generic (e.g., routing), some application-dependent (e.g., information dissemination and querying), and some situation-dependent (e.g., task assignment or software configuration). The dynamic nature of sensor networks prevents *a priori* design of optimal behaviors to implement such functions. For this reason, researchers investigate self-organizing techniques that could enable a network to shape its own behaviors based on environment and need.

4.3.1. Routing

Nodes within sensor networks appear with new deployments and disappear due to power exhaustion, periods of inactivity, and vulnerability to destruction. Such dynamics, coupled with desire to conserve power while limiting packet latency, present difficult challenges for routing algorithms. Servetto and Barrenechea [65] investigate how *interacting particle systems* (modeled as probabilistic walks on random graphs) might yield efficient multi-path routing in networks with many fixed sensors that power themselves off and on at random times in order to conserve power. Servetto defines a distributed algorithm where each node computes local parameters for a random walk such that the global network will exhibit two properties: short routes and evenly distributed packet-forwarding demands. In computing its parameters, each node uses local information augmented only by information from one-hop neighbors and from packets transiting the node.

Tang *et al*. [66] consider a unique problem associated with medical sensors implanted in human subjects.

DOI: 10.1002/wcm

Since radio frequency communication produces electromagnetic fields that can be absorbed by (and heat) human tissue, they propose a thermal-aware routing protocol that avoids hot spots. Temperature is estimated for points in a grid by using a continuous-time, differential, (Pennes) *bioheat equation*. Next routing hops for packets are selected based on temperature rather than shortest path. If a packet cannot advance (due to temperature constraints), then the packet is returned to the previous hop, which tries another path or returns the packet to its previous hop. Packets destined for a hot spot will be buffered until estimated temperature drops, and packets that cannot be delivered within a deadline are discarded. Simulation results show that thermal-aware routing yields a smaller maximum and average temperature increase and induces less traffic congestion than shortest-path routing—though shortest-path routing gives lower packet latencies.

4.3.2. Information dissemination

Information-dissemination protocols push data from sources (e.g., sensors) toward destinations for which information could be relevant. For example, sensors within various rooms in a building might push changing temperatures toward a fire-alarm controller. Such protocols should conserve energy, provide low latency, and tolerate node and link failures. Intanagonwiwat *et al*. [67] propose a *directed-diffusion* protocol where information, represented as attribute-value pairs, is drawn toward consumers that express an interest. A data consumer periodically sends to its neighbors a task consisting of a time-to-live, an event rate, and a list of attribute-value pairs. Nodes cache each received interest, along with one or more gradients, where each gradient defines a direction of flow and a desired event rate associated with one neighbor. Interests diffuse through a network as nodes forward received interests to neighbors. Typically, a consumer will disseminate a request for events to be received at a slow rate. Subsequently, the consumer evaluates the quality and timeliness of received events and then reinforces one particular neighbor by disseminating interest in a higher event rate. The reinforcement diffuses toward nodes providing desired data. Directed diffusion adapts automatically to failures in sensor nodes. Simulation results show that directed diffusion yields lower energy use and lower delay than a typical flooding algorithm.

Wischhof *et al*. [68] describe an ambitious project to develop a self-organizing system where information about traffic conditions propagates, using an *epidemic model*, among cars moving along a highway. Some cars are assumed to be equipped with special gear (e.g., global-positioning system, wireless radio hardware and computer connected to in-car sensors). Each equipped car conducts a repeated cycle of reception, analysis, and transmission. During reception, a car receives information from any cars within radio range. Based on received information, a car updates its own traffic picture during an analysis phase, and subsequently transmits its updated traffic picture to cars within range. Given that cars are moving relative to each other in various directions, traffic information propagates throughout the roadway.

4.3.3. Information query

Query protocols allow consumers to pull data from relevant sources, e.g., an intrusion-alarm controller within a building might periodically check readings maintained by motion sensors attached to various doors and windows. Wang *et al*. [52] consider a specific application where sensors are used to determine a target's location. Given an estimate of location, they wish to choose a sensor to query in order to increase estimate accuracy. They propose querying the sensor with information that would yield the largest reduction in uncertainty, represented as *entropy* associated with the probability distribution of the target's location. Simulation results show that entropy-based, sensor selection, with its lower computational demand, works nearly as effectively as more computationally demanding approaches.

Braginsky and Estrin [69] consider routing queries in sensor networks without a suitable geographical organization. For example, one might search for concentrations of a particular chemical or for acoustic events matching a specific signature, rather than seek information about a particular room or location. They propose *rumor routing*, which propagates queries using a random walk and allows network nodes to learn routes (through discovery agents) to various events in the network, and to optimize those paths over time. Once a (random-walk) query intersects with a path to an event of interest, the random walk ceases and the query follows the previously discovered path. The protocol is designed so that both discovery agents and queries have a limited time-to-live. The number of discovery agents is also a design parameter. The goal of rumor routing is to provide a tunable (energy cost vs. discovery probability) design alternative to flooding of events or queries.

DOI: 10.1002/wcm

4.3.4. Task assignment

Sensor networks may require a subset of nodes to host or provide particular services, such as translating between incompatible protocols or aggregating, caching or filtering data. Deciding which nodes should perform particular functions may require consideration of the capabilities or state of individual nodes, the network topology and variations in demand. These factors suggest the need to dynamically assign tasks, roles, or services to specific nodes and then to reassign them as conditions change. Itao and colleagues [70] investigate *biologically inspired models* for autonomous components to establish cooperative relationships to provide network services. Components discover other components and exchange sets of traits, such as identity, type, and capabilities. Each component maintains a relationship record for other discovered components to track the number and utility of interactions. When requested to provide a service, a component may enlist other components as needed based on their capabilities and on the strength of existing relationships. Users reward service providers based upon satisfaction received; the reward function is used to increase relationship strengths among components that cooperate to provide a service.

4.3.5. Software configuration

Wireless nodes may operate in a heterogeneous environment where channel conditions and protocols vary with place and time. This suggests need for nodes to sense the environment and reconfigure platform software as necessary. Such reconfiguration may involve dynamically loading and unloading appropriate software modules or tuning parameter settings to achieve desired performance. Suzuki and Yamamoto [71] describe an approach, modeled after the *immune system*, allowing system configuration policies to be determined dynamically and continuously based on measured system conditions. Pathological system conditions (e.g., server overload) are recognized as antigens that stimulate antibodies (e.g., policies for thread management, caching, and transport protocol) based on antigen concentrations. Positive and negative reinforcement signals drive evolution of antibody generation as system conditions vary. Simulation results show that dynamic reconfiguration provides superior throughput when compared against a default, static configuration selected to match nominal operating conditions.

4.4. Resource Management

Some critical resource management operations underlie many functions in sensor networks. For example, organizing a transmission schedule to limit interference requires that neighboring nodes have a synchronized notion of period and phase. Similarly, choosing sleep and wake periods for a node demands sufficient inter-node synchrony. Alternating sleep and wake periods provide one means of conserving power. Several other options may also be implemented to extend network lifetime.

4.4.1. Synchronization

Werner-Allen and colleagues [72] describe an algorithm for time synchronization based on a mathematical model representing the method used by *fireflies* to synchronize spontaneously. Further, these researchers provide an analysis, simulation, and implementation of the algorithm in the context of a multi-hop sensor network with asymmetric links and message losses. Results with a 24-node test bed achieve synchronization of about 130 ms (median) within less than 5 min.

Hong *et al*. [73] describe and characterize an algorithm, based on *pulse-coupled oscillators*, for reaching consensus regarding detection of a binary event in a distributed sensor network. The algorithm encodes a locally detected event as a linear function of a perturbation aimed to shift the pulse time of a local oscillator, which influences coupled oscillators to shift their own pulse times. The positive feedback loops that develop drive the entire system of coupled oscillators to pulse simultaneously, representing consensus that an event is detected. Failure to pulse represents consensus that no event is detected. Mathematical arguments, supported by numerical simulations, indicate the approach scales efficiently and reaches certain consensus as the number of sensors increases.

4.4.2. Power conservation

Most designs for wireless sensor networks consider techniques to reduce energy consumption. Two fundamental techniques include powering off radios and limiting transmission power. Chen and colleagues [46] observe that all nodes need not be powered on at all times in networks with sufficient density—in fact they argue that powering on too many nodes creates interference and diminishes network capacity. They define an algorithm, akin to *communism*, allowing

DOI: 10.1002/wcm

nodes to make local decisions about when to sleep and when to wake and begin forwarding. Whenever a node discovers two neighbors cannot communicate, the node delays before volunteering to forward packets. Nodes with more power delay for a shorter time, as do nodes that would connect more neighbors. This allows nodes with best ability and greatest utility to power on, allowing less capable and less beneficial nodes to remain dormant.

Conner *et al*. [74] investigate two complementary algorithms to increase the lifespan of sensor networks. One algorithm systematically adjusts a network topology to shift forwarding burden to energy-rich nodes, while the other algorithm enables non-forwarding nodes to sleep most of the time without missing packets. The topology-control algorithm, which adjusts based on periodic probing, favors selecting fewer forwarding nodes that are more richly connected, leading to a shallow network where most nodes can be reached within a hop or two. The node-scheduling algorithm allows a node at power up to discover (via snooping) the current schedule during which other nodes send short messages indicating any intention to send a data packet. The new node can then select an open spot in the schedule. To send a data message, a node first announces an intention to send at a particular time (avoiding known conflicts) to a particular destination, which will then know when to wake up to receive the transmission. This algorithm assumes that data transmissions are relatively rare and that power savings may be traded for higher latency.

Kubisch *et al*. [75] compare two node-local algorithms for adapting transmission power within fixed, wireless sensor networks. One algorithm requires nodes to periodically broadcast probe packets and to listen for acknowledgments from neighboring nodes. Failure to receive a sufficient number of acknowledgments stimulates a node to increase transmit power and retry. Receiving too many acknowledgments causes a node to decrease transmit power and retry. Receiving a target number of acknowledgments terminates a probe period and establishes a level for transmission power. The second algorithm includes in each acknowledgment the number of neighbors that can be reached by the respondent. The probe issuer computes a mean number of neighbors that it should be able to reach. If the mean is too small, then transmit power is increased and another probe is sent. If the mean is too large, then transmit power is decreased and another probe issued. Simulation results find that using these algorithms leads to network lifetimes within a lifetime or two of the global optimum that might be achieved using centralized computations.

4.5. Resilience

Given potential for sensor networks to be deployed in critical applications, issues arise regarding resiliency in the face of failures and attacks. Gupta and Younis [76] propose a method to recover sensors from a cluster with a failed cluster head. Their method does not require network-wide re-clustering. Fault detection depends upon cluster heads periodically exchanging vectors indicating perceived status of other cluster heads. Each cluster head uses these vectors to determine a consensus view of failed cluster heads. The interval between vector exchanges expands multiplicatively over time when all cluster heads appear operational and contracts linearly during periods when some cluster heads appear suspect. Variation in the vector-exchange cycle lowers overhead for stable topologies, yet improves responsiveness during periods of instability. Fault recovery depends upon the initial technique adopted for cluster formation, where the protocol has cluster heads identify all sensors within radio range and then partition that set into primary and backup cluster members. The partitioning places sensors into the primary set based on minimizing communication cost. During recovery, sensors in backup sets are reassigned to the primary set of the cluster head that offers the lowest communication cost.

Potential attacks against sensor networks come in a variety of forms, such as injecting false sensor reports and draining network power. Ye *et al*. [77] investigate a statistical mechanism to detect and drop false information within a large, dense, sensor network where elected nodes aggregate and forward readings collected by nearby sensors. The mechanism requires that each data sink possess an indexed collection of keys partitioned into disjoint sets and that each sensor is randomly assigned a subset of index-key pairs from one partition. Any sensor report is forwarded along with a message hash generated based on one of the keys (key index also forwarded) within the sensor. An aggregating node forwards a sensor report along with one hash and key index in each of some number of key partitions. While flowing through the network, probability increases that a report transits a node that shares one of the keys used to generate one of the hashes. In such a case, the transit node can verify the hash and could detect a forged report because a compromised node is unable to correctly forge all hashes for an aggregated report. Analysis

DOI: 10.1002/wcm

and simulation results suggest that the proposed mechanism could drop between 80% and 90% of injected false reports within 10 forwarding hops with an overhead of only 14 bytes per sensor report. Dropping false reports early would reduce energy consumption and extend the network lifetime by a factor of two.

Yu and Liu [78] propose a self-organizing scheme that encourages nodes to cooperate and simultaneously to resist attacks aimed to degrade performance and shorten network lifetime. Assuming that node identities may not be spoofed, the scheme requires that every sent packet be acknowledged and that acknowledgments for packets ripple back along the transmission route from destination to source. Forwarding packets and receiving acknowledgments cause updates to a balance sheet indicating the net difference between the utility a node contributes to each of its neighbors and the utility each neighbor contributes to the node. Nodes continue to forward packets for neighbors unless the net negative utility falls below some threshold. Route discovery is augmented to include information about the relative net utility between a node and all other nodes on particular paths. Packets will not be forwarded along routes without sufficient net utility to ensure delivery. Over time, cooperating nodes reinforce their net utilities and malicious nodes are shunned.

5. Open Issues

Researchers have yet to experiment with self-organizing designs that can simultaneously address multiple dimensions of performance, security, and robustness. One wonders how (or whether) a complete set of design objectives might be satisfied within a self-organizing framework? Do some underlying principles unify all approaches to self-organization? If so, what are those principles? Do selected mechanisms and models work best for specific problems? What are the implications of combining various mechanisms within the same system design? Will interaction effects arise? How could such effects be identified and mitigated?

Phase transitions pose another area of concern. Many natural systems tend to self-organize to critical equilibrium of a fragile nature. Could self-organizing networks exhibit similar propensity? Recall that Krishnamachari [55] reported phase transitions in wireless networks, identifying a critical threshold of node density that leads to global connectivity. Below the threshold a network will not connect; above the threshold a network generates interference and wastes energy. Krishnamachari suggests that phase-transition analysis could help to select design parameters that enable a self-organizing wireless network to reach a desirable operating point. But what about the possibility for changing conditions to disturb equilibrium and induce periods of instability, or to drive a system into oscillation or chaos? Can such conditions be forecast, analyzed and resisted?

Overall, the picture appears cloudy with regard to self-organization in wireless sensor networks. Further research is needed to develop techniques to measure, analyze, and visualize macroscopic behavior. Without an ability to understand global consequences of particular design decisions, deploying self-organizing networks could prove to be risky.

References

1. Robertazzi TG, Sarachik PE. Self-organizing communication networks. *IEEE Communications* 1986; **24**(1): 28–33.
2. Floyd S, Jacobson V. The synchronization of periodic routing messages. *IEEE/ACM Transactions on Networking* 1994; **2**(2): 122–136.
3. Antunes N, *et al*. Metastability of CDMA cellular systems. In *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, 2006, pp. 206–214.
4. Mills K, Dabrowski C. Investigating global behavior in computing grids. *Self-Organizing Systems* 2006; LNCS 4124: 120–136.
5. Hillis WD. Intelligence as an emergent behavior or, the songs of eden. *Daedalus, Journal of the American Academy of Arts and Sciences* 1998; (winter): 175–189.
6. Kauffman SA. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.
7. May RM. *Stability and Complexity in Model Ecosystems*. Princeton University Press, 1973.
8. Cohen JE, Briand E, Newman CM. *Community Food Webs: Data and Theory*. Springer Berlin, 1990.
9. Turcotte DL. *Fractals and Chaos in Geology and Geophysics* (2nd edn). Cambridge University Press, 1997.
10. Bhalla US, Iyengar R. Emergent properties of networks of biological signaling pathways. *Science* 1999; **283**(5400): 381–387.
11. Yerra BM, Levenson DM. The emergence of hierarchy in transportation networks. *The Annals of Regional Science* 2005; **39**(3): 541–553.
12. Plerou V, *et al*. Random matrix approach to cross correlations in financial data. *Physical Review* 2002; E 65 066126.
13. Bak P. *How Nature Works—the Science of Self-Organized Criticality*. Copernicus, 1996.
14. Ito K. Punctuated-equilibrium model of biological evolution is also a self-organized-criticality model of earthquakes. *Physical Review* 1995; **E 52**: 3232–3233.
15. Stanley HE. *Introduction to Phase Transitions and Critical Phenomena*. Oxford University Press, 1971.
16. Kuramoto Y. *Chemical Oscillations, Waves, and Turbulence* Springer-Verlag, 1984.
17. Roli A, Zambonelli F. Emergence of macro spatial structures in dissipative cellular automata. In *Proceedings of Cellular Automata: 5th International Conference on Cellular Automata for Research and Industry*, Springer Berlin, 2002.
18. Erdös P, Rényi A. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*. 1960; **5**: 17–61.

19. Kay JJ. *Self-organization and the Thermodynamics of Living Systems: A Paradigm* 1984; PhD Thesis, University of Waterloo.
20. Johnson NL, *et al*. Symbiotic Intelligence: Self-Organizing Knowledge on Distributed Networks Driven by Human Interaction. In *Proceedings of the Sixth International Conference on Artificial Life*, MIT Press, 1998, pp. 403–407.
21. Linden DS, Althshuler EA. Evolving wire antennas using genetic algorithms: a review. In *Proceedings of the First NASA/DOD Workshop on Evolvable Hardware* 1999; IEEE Computer Society: 225.
22. Bonabeau E, Meyer C. Swarm intelligence: a whole new way to think about business. *Harvard Business Review* 2001; **79**(5): 106–114.
23. Mostefaoui SK, *et al*. Self-organising applications: a survey. In *Proceedings of the International Workshop on Engineering Self-Organising Applications*, Springer-Verlag, 2003.
24. Karagiannis T, Molle M, Faloutsos M. Long-range dependence—Ten years of internet traffic modeling. *IEEE Internet Computing* 2004; **8**(5): 57–64.
25. Shalizi CR. Power law distributions, 1/f Noise, Long-memory time series. *Notebooks* 2006; http://www.cscs.umich.edu/~crshalizi/notebooks/power-laws.html
26. Parunak H, VanderBok RS. Managing emergent behavior in distributed control systems. In *Proceedings of the ISA Tech*, Instrument Society of America, 1997.
27. Abry P, Veitch D. Wavelet analysis of long-range dependent traffic. *IEEE Transactions on Information Theory* 1998; **44**(1): 2–15.
28. Oprisan SA. Theoretical approach on microscopic bases of stochastic functional self-organization: quantitative measures of the organizational degree of the environment. *Journal of Physics A, Mathematical and general* 2001; **34**(37): 10013–10028.
29. Ho L, Samuel LG, Pitts JM. Applying emergent self-organizing behavior for the coordination of 4G networks using complexity metrics. *Bell Labs Technical Journal* 2003; **8**(1): 5–25.
30. Bush SF. Genetically induced communication network fault tolerance. *Complexity Journal* 2003; **9**(2).
31. Abelson H, *et al*. Amorphous computing. *Communications of the ACM* 2000; **43**(5): 74–82.
32. Lodding KN. Hitchhikers guide to biomorphic software. *ACM Queue* 2004; **2**(4): 66–75.
33. Nagpal R. A catalog of biologically-inspired primitives for engineering self-organization. *Engineering Self-Organising Systems: Nature-Inspired Approaches to Software Engineering* (2003); **2977**: 53–62.
34. Kohonen T. *Self-Organization and Associative Memory* (3rd edn). Springer-Verlag, 1989.
35. Kohonen T, *et al*. Engineering applications of the self-organizing map. In *Proceedings of the IEEE*, 1996, **84**(10): pp. 1358–1384.
36. Hofmeyr S, Forrest S. Architecture for an artificial immune system. *Evolutionary Computation* 2000; **8**(4): 443–473.
37. Ritsko JJ, *et al*. *IBM Systems Journal* 2003, special issue on Autonomic Computing, **42**(1).
38. Shmulevich I, Dougherty ER, Zhang W. From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Bioinformatics* 2002; **18**(2): 261–274.
39. Wilkinson IF, Wiley JB. Modeling the structural dynamics of industrial networks. In *Proceedings of the 4th International Conference on Complex Systems* 2000.
40. Fogel DB (ed.). *Evolutionary Computation—The Fossil Record* IEEE Press, 1998.
41. Reynolds CW. Flocks Herds, and Schools: a distributed behavioral model. *Computer Graphics* 1987; **21**(4): 25–34.
42. Moiseff A, Copeland J. Mechanisms of synchrony in the North American firefly. Photinus carolinus (Coleoptera: Lampyridae). *Journal of Insect Behavior* 1994; **8**(3): 381–394.
43. Grassé PP. La reconstruction du nid et les coordinations inter-individuelles chez Bellicositermes natalensis et Cubitermes sp. La theorie de la stigmergie: Essai d'interpretation des termites constructeurs. *Insectes Sociaux* 1959.
44. Bonner JT, Lamont DS. Behavior of cellular slime molds in the soil. *Mycologia* 2005; **97**(1): 178–184.
45. Irwin D, Grit L, Chase J. Balancing risk and reward in a market-based task service. In *Proceedings of the 13th Symposium on High Performance Distributed Computing*, 2004, pp. 160–169.
46. Chen B, *et al*. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *ACM Wireless Networks Journal* 2002; **8**(5): 85–96.
47. Vahdat A, *et al*. Self-organizing subsets: from each according to his abilities, to each according to his needs. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems*, 2002, **2429**: pp. 76–84.
48. Hilaire V, *et al*. A formal approach to design and reuse agent and multiagent models. *Agent-Oriented Software Engineering V*. Springer, 2004.
49. Mahajan R, *et al*. Experiences applying game theory to system design. In *Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*, 2004, pp. 183–190.
50. Eymann T, Padovan B, Schoder D. The Catallaxy as a new Paradigm for the design of information systems. In *Proceedings of the 16th IFIP World Computer Congress, Conference on Intelligent Information Processing* 2000.
51. Wong T, Tsuchiya T, Kikuno T. A self-organizing technique for sensor placement in wireless micro-sensor networks. In *Proceedings of the 18th International Conference on Advanced Information Networking and Application*, 2004, pp. 78–83.
52. Wang H, *et al*. Entropy-based sensor selection heuristic for target selection. In *Proceedings of Information Processing in Sensor Networks*, 2004 pp. 36–45.
53. Heo N, Varshney P. An intelligent deployment and clustering algorithm for a distributed mobile sensor network. In *Proceedings of the IEEE International Conference On Systems Man And Cybernetics*, 2003, pp. 4576–4581.
54. Faruque J, Psounis K, Helmy A. Analysis of gradient-based routing protocols in sensor networks. *Distributed Computing in Sensor Systems: First IEEE International Conference*. Springer Berlin, 2005.
55. Krishnamachari B, *et al*. On the complexity of distributed self-configuration in wireless networks. *Telecommunication Systems* 2003; **22**(1–4): 33–59.
56. Srinivasan V, *et al*. Cooperation in wireless ad hoc networks. In *Proceedings of IEEE INFOCOM*, 2003.
57. Willig A, *et al*. Altruists in the picoradio sensor network. In *Proceedings of IEEE Workshop on Factory Communication Systems*, 2002.
58. Kompella R, Snoeren A. Practical lazy scheduling in sensor networks. In *Proceedings of SenSys'03*, 2003 pp. 280–291.
59. Duque-Anton M, Ruber B, Killat U. Extending Kohonen's self-organizing mapping for adaptive resource management in cellular radio networks. *IEEE Transactions on Vehicular Technology* 1997; **46**(3): 560–568.
60. Cerpa A, Estrin D. ASCENT: Adaptive self-configuring sensor networks topologies. In *Proceedings of IEEE INFOCOM*, 2002 pp. 1278–1287.
61. Chan H, Perrig A. ACE: An emergent algorithm for highly uniform cluster formation. In *Proceedings of the 1st European Workshop on Wireless Sensor Networks*, 2004, **2920**: pp. 154–171.
62. Hester L, *et al*. neuRFon Netform: A self-organizing wireless sensor network. In *Proceedings of the 11th IEEE ICCCN Conference*, 2002.
63. Low KH, *et al*. Task allocation via self-organizing swarm coalitions in distributed mobile sensor network. In *Proceedings of the 19th National Conference on Artificial Intelligence*, 2004; 28–33.

DOI: 10.1002/wcm

64. Parunak H, Brueckner S. Stigmergic learning for self-organizing mobile ad-hoc networks (MANET's). In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2004, pp. 1324–1325.
65. Servetto S, Barrenechea G. Constrained random walks on random graphs: routing algorithms for large scale wireless sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, 2002 pp. 12–21.
66. Tang Q, *et al*. TARA: Thermal-aware routing algorithm for implanted sensor networks. In *Proceedings of the International Conference on Distributed Computing in Sensor Systems*, 2005, **LNCS 3560**: pp. 206–217.
67. Intanagonwiwat C, *et al*. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, 2000, pp. 56–67.
68. Wischhof L, *et al*. SOTIS—A Self-Organizing traffic information system. In *Proceedings of the 57th IEEE Vehicular Technology Conference*, 2003, **4**: pp. 2442–2446.
69. Braginsky D, Estrin D. Rumor routing algorithm for sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks*, 2002, pp. 22–31.
70. Itao T, *et al*. Service emergence based on relationship among self-organizing entities. In *Proceedings of the Symposium on Applications and the Internet*, 2002.
71. Suzuki J, Yamamoto Y. Building an artificial immune network for decentralized policy negotiation. In *Proceedings of the 4th World Multiconferences on Systemics, Cybernetics, and Informatics*, 2000.
72. Werner-Allen G, *et al*. Firefly-inspired sensor network synchronicity with realistic radio effects. In *Proceedings of ACM SenSys'05*, 2005, pp. 142–153.
73. Hong Y-W, Cheow LF, Scaglione A. A simple method to reach detection consensus in massively distributed sensor networks. In *Proceedings of the International Symposium on Information Theory*, 2004, pp. 250–255.
74. Conner WS, *et al*. Experimental evaluation of synchronization and topology control for in-building sensor network applications. In *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, 2003, pp. 38–49.
75. Kubisch M, *et al*. Distributed algorithms for transmission power control in wireless sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, 2003.
76. Gupta G, Younis M. Fault-tolerant clustering of wireless sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, 2003, **3**: pp. 1579–1584.
77. Ye F, *et al*. Statistical en-route filtering of injected false data in sensor networks. *IEEE Journal on Selected Areas in Communications* 2004; **23**(4): 839–850.
78. Yu W, Liu K. Attack-resistant cooperation stimulation in autonomous ad hoc networks. *IEEE Journal on Selected Areas in Communications* 2005; **23**(12): 2260–2271.

Author's Biography

Kevin Mills (senior member of the IEEE) received his Ph.D. (1996) in information technology from the George Mason University in Fairfax, Virginia. He is a senior research scientist with the Information Technology Laboratory of the National Institute of Standards and Technology (USA), where he has served in various research and management positions since 1982, except for a term (1996 through 1999) as program manager with the Defense Advanced Research Projects Agency (DARPA). Dr. Mills was on the adjunct faculty (1996 to 2006) of the School for Information and Technology Engineering at George Mason University. His research interests include complex, distributed systems.

DOI: 10.1002/wcm