Real-Time Communication and Coordination in Embedded Sensor Networks

JOHN A. STANKOVIC, FELLOW, IEEE, TAREK F. ABDELZAHER, MEMBER, IEEE, CHENYANG LU, LUI SHA, FELLOW, IEEE, AND JENNIFER C. HOU, SENIOR MEMBER, IEEE

Invited Paper

Sensor networks can be considered distributed computing platforms with many severe constraints, including limited CPU speed, memory size, power, and bandwidth. Individual nodes in sensor networks are typically unreliable and the network topology dynamically changes, possibly frequently. Sensor networks also differ because of their tight interaction with the physical environment via sensors and actuators. Because of this interaction, we find that sensor networks are very data-centric. Due to all of these differences, many solutions developed for general distributed computing platforms and for ad-hoc networks cannot be applied to sensor networks. After discussing several motivating applications, this paper first discusses the state of the art with respect to general research challenges, then focuses on more specific research challenges that appear in the networking, operating system, and middleware layers. For some of the research challenges, initial solutions or approaches are identified.

Keywords*—*Embedded systems, middleware, networking, operating systems, real time, sensor networks.

I. INTRODUCTION

Many future applications will rely on an embedded sensor network. A sensor network is a general term that covers many variations in composition and deployment. A typical sensor network consists of a large number of nodes deployed in the environment being sensed and controlled. In many cases, each node of the sensor network consists of sensors and wireless communication. Memory, power, and computational capacities are typically limited. In other sensor networks, nodes may also contain actuators. Often sensor nodes are densely deployed, are prone to failures, and the topology of the network can dynamically change. Sensor networks may consist of all homogeneous nodes or exhibit a heterogeneous structure where some nodes are much more powerful than others or contain different sets of resources. These networks are very data-centric, with data queries being issued from base stations and time-dependent sensor data being routed and aggregated throughout the network.

Regardless of the variant of a sensor network, it is necessary to support real-time communication and coordination. For many reasons to be discussed in this paper, this is an exciting, but very challenging problem. Fundamentally, new paradigms and solutions are required.

Applications for this technology are numerous. One class of application is the monitoring and control of safety-critical military, environmental, or domestic infrastructure systems. This includes battlefield applications, biological, chemical or radiological detection and protection systems, or aiding areas hit by disasters. Another class of application is the so-called smart space. This may include smart factories, buildings, cities, or universities. A third class of application is in entertainment. This may include amusement parks or museums. Many of the challenges to be discussed apply to all applications, although the degree to which certain issues apply is application dependent.

To better understand the rest of the paper, we describe one type of application in more depth. Sensor networks can be used for homeland security at airports, bridges, and public buildings. As it is rather difficult for security guards to continuously watch a set of video monitors when most of the time nothing occurs, the overall security effectiveness will improve when the security video system is coupled with motion detectors and/or acoustic monitoring and alerts based on unusual sounds. For this type of sensor network, a large number of low-cost lightweight wireless devices is scattered over a geographic region and forms a surveillance and communication network whose major function is to locate and track unusual sounds in the region. These wireless

Manuscript received August 14, 2002; revised December 17, 2002. This work was supported in part by the Multidisciplinary University Research Initiative (MURI) under Award N00014-01-1-0576.

J. A. Stankovic and T. F. Abdelzaher are with the Department of Computer Science, University of Virginia, Charlottesville, VA 22904-4740 USA (e-mail: stankovic@cs.virginia.edu).

C. Lu is with the Department of Computer Science and Engineering, Washington University, St. Louis, MO 63130-4899 USA.

L. Sha and J. C. Hou are with the Department of Computer Science, University of Illinois, Urbana, IL 61801 USA.

Digital Object Identifier 10.1109/JPROC.2003.814620

devices are equipped with acoustic sensors and can locate a sound wave (by determining the magnitude of the sound and the angle of arrival, and performing primitive frequency analysis). The nodes organize themselves dynamically and convey the location information periodically or on-demand to controller nodes, which then take appropriate actions under real-time constraints. More than one sensor may observe the same phenomenon and, hence, the information collected by various sensors may be correlated, redundant, and/or of different qualities. It is expected that most of these devices have limited battery life and transmission/computational capability, but a few of them may be equipped with better processing capability, stronger transmission power, and longer battery life. These energy-rich nodes can act as controllers or cluster heads and perform processing and communication operations.

The main purposes of this paper are to overview the state of the art and to present key research challenges in real-time communication and coordination in embedded sensor networks. To meet these objectives, the paper first presents general research challenges (Section II). These are the global overarching challenges that this area faces. We discuss these challenges for six topics: paradigm shifts, resource constraints, unpredictability, high density and scale, real-time, and security. We then address more detailed and specific challenges in the network layers (Section III), and the operating system and middleware layers (Section IV). When appropriate we indicate current solutions and approaches to meeting the challenges. We conclude with a brief summary in Section V.

II. General Research Challenges

The general research challenges for real-time communication and coordination in sensor networks arise primarily due to the large number of constraints, many of them new, that must be simultaneously satisfied. For example, large distributed computer systems (the Internet) have existed for a long time. However, solutions for communication and coordination in those systems did not have to address small capacities in memory, limited CPU execution speeds, and scarce communication bandwidth. Further, many classical solutions did not address minimizing power, interacting with real world events through sensors and actuators, or meeting real-time constraints. On the other hand, there have been distributed embedded systems such as those that exist on submarines or in factories. These systems do deal with sensors and actuators, real-time constraints, cost, and other issues, but they do not have solutions for many of the key issues such as those dealing with wireless communication, large scale, power management, and unreliable devices.

Unlike traditional wired or wireless networks, sensor networks possess certain characteristics which warrant their treatment as a special class of ad-hoc network.

Data-centric: Sensor networks are largely data-centric, with the objective of delivering time sensitive data, in a timely fashion, to the required destination.

Application-oriented: While traditional wired and wireless networks are expected to cater to a variety of user applications, a sensor network is usually deployed to perform specific tasks. This property makes it possible to enable nodes to respond in an application-aware fashion. Data can be collected, appropriately aggregated with consideration of the requirement of the applications, and then acted on locally and/or forwarded to a higher level controller node (rather than simple end-to-end data transfer).

Note that many of the research challenges and solutions presented in this paper overlap. However, to organize the presentation of the general challenges we structure the discussion as follows:

- paradigm shift;
- resource constraints;
- unpredictability;
- high density/scale;
- real time;
- security.

A. Paradigm Shift

Fundamentally, a wireless sensor network is deployed to support an integrated set of functions/applications. The system must sense and act to produce the desirable outcomes. As mentioned above, the severe constraints give rise to the need for a new paradigm. In particular, it is critical to produce aggregate behavior of the system where any single node is not important. In fact, nodes should not have any permanent ID. Messages should not be sent to individual nodes, but instead to locations or areas based on data content. For example, a user might want to know the average temperature in the basement of a building; he does not care which nodes respond. Or, he may want to know what area has a temperature above a certain threshold. These examples illustrate that these sensor networks are very data-centric. The fact that the sensor network interacts with the physical environment also implies differences from many classical distributed systems solutions. This is largely due to real-time requirements, high degree of faults, noise, and nondeterminism caused by the uncontrolled aspects of the environment.

New paradigms are being developed based on biological metaphors in projects such as the amorphous computing project at the Massachusetts Institute of Technology (MIT), Cambridge [5], [36]. Other paradigms are exploiting the data-centric aspects of the system, and still others are creating solutions that depend on high density. These and other ideas may lead to effective paradigms in the future.

B. Resource Constraints

Many new solutions are needed because of the severe resource limitations. The main resources in short supply include power, CPU execution speed, memory, and communication bandwidth. Since the sensor network is likely to contain a very large number of nodes, cost is also a significant problem. Not only are novel solutions needed to solve specific problems, but also to deal with tradeoffs. For example, better power management for a node is required. This may involve putting a node or various components on that node to sleep. In addition, it is necessary to decide when to transmit with greater power so that fewer hops are required to reach the destination or when is it better to transmit at low power and traverse more hops. If a node is having trouble getting its message received properly, it may be able to physically move, send at higher power, or send at a different frequency. Many of the new resource allocation and management problems that are exhibited in sensor networks have this flavor of a large number of potential actions to take. How to make this decision and how to understand the overall quality of the resource decisions for the entire sensor network are key challenges.

C. Unpredictability

A sensor network is subject to a great deal of uncertainty from many quarters. First, the sensor network is deployed in an environment with uncontrollable aspects (e.g., when and where will a fire break out in a city hit by an earthquake). Second, the wireless communication is subject to many physical errors and missing messages due to radio interference of many types. Third, individual nodes are not reliable. Fourth, sensors may not all be calibrated properly. Fifth, the connectivity and routing structures are changing dynamically. There may even be network partitions. Sixth, new nodes may be added or old nodes removed from the sensor network. This implies that the sum total of resource capacity is not fixed. Seventh, power availability at each node can vary significantly even when initially deployed. Eighth, nodes may be physically moved or be controlled to do so under their own power, thereby restructuring the topology. And so on.

One challenge is how to create a view from the application layer that the sensor network is a reliable large-scale entity with known operating performance that can be relied upon? Since sensor networks are deployed to operate with little direct management, they must exhibit self-organizing, self-optimization, and self-healing properties [122]. These are relatively easy to state as challenges, but very difficult to attain.

D. High Density/Scale

A number of solutions for sensor networks depend on an assumption of a minimum density of nodes in the system. Challenges include: computing that density for various situations, ensuring that the sensor network actually achieves that density, and developing solutions that require a minimum density and power in order to minimize cost and maximize lifetime of the system.

If the density is high and the sensor network is deployed in a wide area, then we also have a large-scale system. This large-scale system is subject to many faults, noise, and other uncertainties (as discussed above) and is highly decentralized. Further, when a sensor network is deployed, it must then be largely self-operating and self-maintaining. All of these things can give rise to parts of the system working at conflicting purposes. One of these nefarious interactions is a form of race condition where the system never settles down to some conclusion. Research is needed for protocols and algorithms to be self-stabilizing. In spite of the fact that algorithms must be simple and inexpensive, they must aggregate properly when used in large numbers.

E. Real Time

Sensor networks operate in the real world, hence, timing constraints are important. These systems have implicit time requirements, e.g., when a user enters a room, he should be recognized within a very short time. The faster such a task is accomplished the better we consider the system. However, many sensor networks will also have explicit real-time requirements related to the environment. For example, an accelerometer might have to be read every 10 ms, or else there will be a bad estimate of speed and consequently a high probability of a vehicular crash. There may also be deadlines associated with end-to-end routing, e.g., a sensitive pressure reading might have to periodically arrive at a monitor and actuation station on time, each time. Because of the large scale, nondeterminism, noise, etc., it is extremely difficult to guarantee real-time properties. New research that employs feedback control [2], [91], [90] seems to have promise. However, many challenges in the real-time design and analysis of solutions for sensor networks exist. These challenges are exacerbated due to the large scale and unreliable aspects of these systems.

F. Security

Since many sensor networks will be deployed in critical applications, security is essential. Unfortunately, security may be the most difficult problem to solve. In particular, it is easy to eavesdrop or cause a denial of service attack on the sensor network. Further, most real-time communication and coordination solutions do not address security, so it is easy for an adversary to exploit those implemented solutions on a given sensor network.

A fundamental dilemma is that sensor networks have limited capacity and security solutions are resource hungry. For example, many sensor networks will deploy a single-frequency communication scheme because of cost and the simplicity of a node. This makes it trivial to eavesdrop. Due to the wireless nature of sensor networks, an adversary can deploy his own node that can take many actions to create a denial of service attack. Some of these are simply broadcasting at high energy, advertising that it is the fastest path to everywhere and simply throwing away packets that arrive, or sending wake-up calls to neighbors to exhaust their power. Note that when an adversary deploys a node to cause denial of service, it is the self-organizing and positive characteristic of sensor networks that opens the system to various security breaches.

Protocol solutions for media access control, routing, congestion control, and others all attempt to operate with minimum overhead and cost. This also subjects them to security problems. For example, a good solution for large-scale sensor networks is to give routing priority to packets passing through a node rather than admitting new packets. This helps prevent long delays for packets that have to traverse a large part of the sensor network. However, this protocol makes flooding attacks more effective, i.e., an intruder performing flooding is actually given preference! Basically, the research challenges for security in sensor networks are vast and difficult. Lightweight schemes are required. Solutions must exploit the nature of the sensor network, possibly related to issues such as: 1) most data is only valid for a short time, so perhaps lightweight security will be effective; 2) individual nodes may possess little knowledge by themselves, so protecting the data aggregation function may be possible; and 3) new ideas on the fundamental limits for security in these systems are needed. For more details, see [133].

III. Networking Research Challenges

Many of the research challenges facing sensor networks reside in the communication layers. We begin this section with a discussion of the requirements facing networking and highlight the key challenges. Many of these challenges cross cut the communication protocol stack. We then discuss, in detail, the state of the art in the medium access control (MAC) (Section III-A), network (Section III-B), and transport layers (Section III-C). We conclude this with a detailed discussion (Section III-D) of three key issues that cross cut the communication stack: power management, topology control, and real-time. In total, this provides a comprehensive view of the real-time and coordination issues in the network layers of a sensor network.

Novel communication protocols must be developed to support higher level services in sensor networks. In most envisioned sensor network applications, a large number of sensors are deployed in an area and a small number of more powerful nodes, called base stations (e.g., gateways to the Internet or command and control centers), form possibly mobile interfaces to users. In this system, a user may query the physical environment through base stations. Alternatively, he may register for an event. The occurrence of the event automatically triggers a specified query. For example, a user can register for a virus-found event in an area and specify a query on the event to report the density of the detected virus. Communication in sensor networks involves both in-network aggregation and sensor-base communication. Before sending information to a base station, sensors within the local area aggregate raw sensor data and generate reliable information. For example, acoustic sensors may perform triangulation among multiple nodes to decide the location of a tank. Sensor-based communication is responsible for reporting (aggregated) data to the base station, which often spans many hops.

A major requirement for sensor networks is to reliably aggregate and disseminate information within a time frame that allows the controllers to take necessary actions, even in the case of poor spatial distribution of sensor devices, wireless/acoustic interference, and malicious destruction. Out-of-date information is of no use; for example, an object that was being tracked may no longer be in the vicinity when the information is received. This presents a key technical challenge in cooperative engagement—how to effectively coordinate and control sensors in real-time over an unreliable wireless ad-hoc network. In particular, due to the unique characteristics of data-centric sensor networks, many new design issues arise and protocols originally designed for wireline and/or generic ad-hoc networks have to be adapted or entirely redesigned.

We now highlight the key challenges that cross cut all layers of the communication stack.

- *Data-centric:* Traditional networks (e.g., the Internet and mobile ad-hoc networks) are address-centric. In such networks, data are communicated through a route between two or more addressed nodes. In contrast, sensor networks are intrinsically data-centric [80]. Data from multiple sources related to the same physical phenomenon need to be aggregated and sent to a base station. The mismatch between address-centric protocols and sensor networks motivated new data-centric protocols [68] that achieve significantly better energy efficiency in sensor networks.
- *Location-based:* Since sensor networks deal with physical environments, data usually correspond to physical locations rather than logical IDs. Hence, data-centric communication can be supported by location-based communication stacks. Instead of querying a sensor with an ID 1002, users often query a physical location or region. The identities of sensors that happen to be located in that region are not necessarily important. Any sensors in that region that receive the query may initiate local coordination to aggregate the requested data. A leader may be elected to send the query result back to the base station. New data-centric and location-based protocols (e.g., directed diffusion [68], greedy perimeter stateless routing (GPSR) [77], and real-time architecture and protocols (RAP) [89]) were developed to improve scalability and efficiency in sensor networks.
- *Large scale:* The large scale of sensor networks requires communication protocols to be highly scalable, maintain minimum global state inside the network, and incur as little control overhead as possible.
- *Unpredictable workloads:* While a sensor network may remain silent for a long time, a communication “hot region” can emerge quickly due to simultaneous events. For example, a fire may cause all active sensors in a region to generate data flows. Highly adaptive protocols are needed to deal with such unpredictable traffic patterns and achieve real-time guarantees.
- *Nonuniform node distribution:* When sensors are placed in open fields for environmental applications, they may not be evenly distributed over a region. It is necessary either to use mobile “router” sensors to fill the “holes” and maintain network connectivity, or to exercise topology and power control in a hierarchical, clustering fashion. The fact that nodes are not uniformly distributed also implies that conventional, flat ad-hoc routing protocols [23], [43], [71], [69], [70],

[78], [92], [103]–[105], [119], [126] may not render the best performance.

- *High fault rates:* Sensor networks are subject to higher fault rates than traditional networks. As in other wireless networks, connectivity between nodes can be lost due to environmental noise and obstacles. Nodes may die due to power depletion, environmental changes, or malicious destruction (e.g., crushed by vehicles). However, the practical utility of sensor networks is usually demonstrated in the presence of faults. In the above example of homeland security, communication protocols that not only are efficient and robust against the failure of individual components, but also self-stabilize in the face of high fault rates must be devised.
- *Energy constraint:* Because sensor networks run on small batteries and often need to operate for a long time, power conservation is a key issue in sensor networks. Recent studies have shown that radio communication is the dominant consumer of energy in sensor networks [65]. Power conservation is an especially important challenge at the communication layers. In the future, solar cells may be attached to sensor network nodes, but energy conservation will remain a key research challenge.

A. MAC Layer

In wireless sensor networks, the MAC performance has been predominantly measured in terms of bandwidth requirement, power consumption, contention mitigation, and support to maintain network connectivity. The latency incurred in message delivery has not been a metric to be optimized, but is likely to become increasingly important as sensor networks are deployed in critical applications. Timeliness is perhaps the most difficult requirement to meet since it brings to the fore the tradeoffs between power consumption, interference mitigation, and scheduling and routing efficiency. Existing MAC protocols for multihop wireless networks can be classified into four categories: 1) scheduling based; 2) collision free; 3) contention based; and 4) hybrid schemes. In what follows, we summarize the state of the art and discuss the advantages and drawbacks of existing approaches with respect to the key challenges of sensor networks. We also specifically identify the special requirements of a MAC layer in sensor networks and evaluate extant technologies in that context.

1) Scheduling-Based MAC Protocols: In scheduling-based MAC protocols, the time at which a node can transmit is determined by a scheduling algorithm, so that multiple nodes can transmit simultaneously without interference on the wireless channel. The time is usually divided into slots, and slots are further organized into frames. Within each frame, a node is assigned at least one slot to transmit. A scheduling algorithm usually finds the shortest possible frame so as to achieve high spatial reuse (and, thus, high network utilization) and low packet latency.

A large amount of early work has been focused on time division multiple access (TDMA) scheduling [9], [10], [30]–[33], [35], [57], [58], [85], [102], [121], [127]. Most of the studies concentrated on devising fair conflict-free algorithms that maximize the system throughput by using graph theory. Most of them are centralized and require global connectivity information. As a result, they cannot adapt adequately and keep the optimality property in highly dynamic environments (such as topology change).

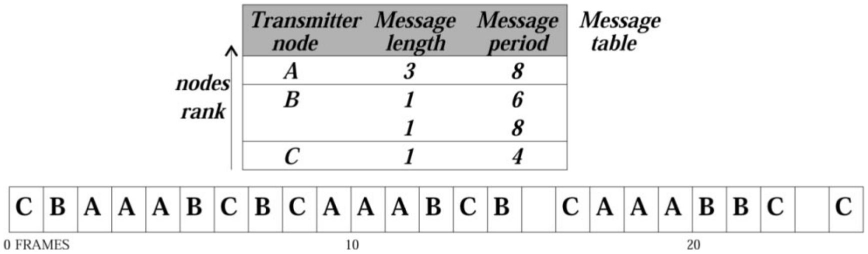
To resolve the above problem, Chlamtac *et al.* [34] first proposed a topology-independent algorithm that depends only on global network parameters, i.e., the number of nodes and the maximum nodal degree. With the use of certain mathematical properties of finite (Galois) fields, the algorithm ensures that for every node and for each of its neighbors, there is at least one slot assigned in each frame. Similar algorithms were proposed in [73] and [74] that use different slot assignment functions to maximize the minimum throughput a node can achieve.

2) Collision-Free Real-Time MAC: The above MAC protocols focus on maximizing spatial reuse and system throughput. An important performance criterion in data-centric sensor networks is timeliness. By exploiting the periodic nature of sensor network traffic, Caccamo *et al.* realize collision-free real-time scheduling as follows [24]: frequency division multiplexing (FDM) is used among adjacent cells to allow for concurrent communications in different cells. Implicit earliest deadline first (EDF) scheduling is used inside each cell. There is a router located in the center area of each cell. Router nodes are equipped with two transceivers so they can transmit and receive at the same time using two different frequency channels.

Intracell communication: The key idea for conflict free real-time scheduling is to replicate the EDF schedule at each node for packet transmission. If the schedules are kept identical, each node will know which one has the message with the shortest deadline and has the right to transmit next. For instance, suppose each node is given a message table as shown in Fig. 1, the same schedule is derived by every node in the cell according to EDF (deadline ties are broken in favor of the node with the highest address ID). Due to the identical ordering of the schedule at each node, a node knows which node should transmit next. In addition, when a node is listening to the channel, it is also able to know the completion of a node's transmission and, thus, update its scheduling queue for the next round of communication.

Take Fig. 1 as an example: the scheduling table reserves the worst case message transmission time for each periodic message stream. Suppose that node A in its first round uses only one of three reserved frames. Since all nodes are listening, they know that Node A has finished early and Node B is the next one to transmit. Instead of transmitting its reserved periodic message early, Node B may use the two frames left by Node A to send best effort aperiodic messages. This is the observation that prompted the development of the FRAme SHaring (FRASH) technique [24] designed to systematically and reliably exploit reserved, but unused, frames.

Intercell communication: Each router node transmits intercell messages using the channel of the cell it belongs to, and receives intercell messages using the channel of the cell it expects to receive from. Intercell messages are ordered by earliest deadline by each router, and each of them is able to



Transmitter node	Message length	Message period
A	3	8
B	1	6
	1	8
C	1	4

C	B	A	A	A	B	C	B	C	A	A	A	B	C	B		C	A	A	A	B	B	C		C

Fig. 1. Example of implicit contention using EDF.

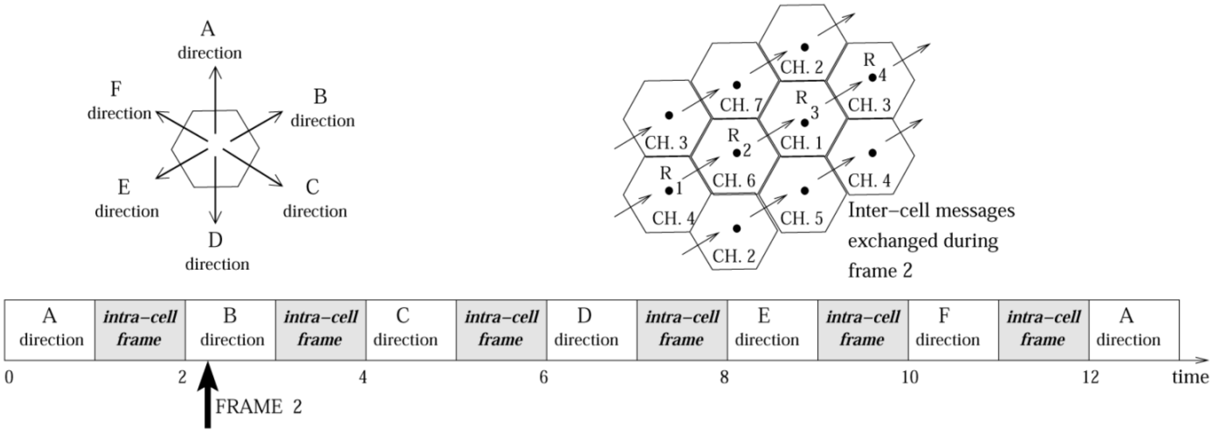


Fig. 2. Example of intercell communication mechanism using TDM.

reach only its six neighboring cells within one hop. Whenever an intercell frame occurs synchronously in all cells, each router transmits and receives intercell messages according to a predetermined direction, which is the same for all cells. Note that there are six possible directions that are assigned statically to the intercell frames following a periodic scheme as Fig. 2 shows.

Take frame 2 as an example: notice that router R_2 is receiving a message from router R_1 using channel four and is transmitting a message to router R_3 using channel six. During the same frame, R_3 is receiving from R_2 on channel six and is transmitting to R_4 on channel one; in short, each router is transmitting and receiving in the same direction at the same time. After the routing path is set, the end-to-end delay is simply the sum of cell delays along the message path.

The interference due to the intercell frames can be taken into account in the cell schedulability analysis as blocking terms. In fact, let m_i be the message transmission time, T_i be message period, and T_{block} $(T_{\text{block}} \geq 2)$ be the period of the intercell frames, the schedulability of intracell messages can be determined by using the approach proposed in [24]: all the messages are sorted by increasing relative deadlines, so that $D_i \leq D_j$ only if $i < j$. It is worth noting that the blocking time of each message is equal to the maximum number of intercell frames that can occur during the message period

$$B_i = \left\lceil \frac{T_i}{T_{\text{block}}} \right\rceil .$$

Due to the contention free nature of this method, implicit EDF not only provides guaranteed schedulability, but also delivers higher throughput, especially during heavy workload as compared with commonly used ad-hoc network protocol such as CSMA/CA, enhanced DCF, and Black-Burst [24].

3) Contention-Based MAC Protocols: Most of the distributed MAC protocols are based on carrier sensing and/or collision avoidance mechanisms and may employ additional signaling control messages to deal with hidden and exposed node problems. Such signaling messages may be delivered in two ways: in-band handshaking or out-of-band signaling. Busy-tone multiple access (BTMA) [125] is a representative of the out-of-band signaling protocol. In BTMA, a node that hears an ongoing transmission transmits a busy tone, and any node that hears a busy tone does not initiate transmission. This eliminates the hidden nodes, but increases the number of exposed nodes.

Another class of MAC protocols uses in-band control packets such as *request to send* (RTS) and *clear to send* (CTS) to exchange the local view of channel status, so as to avoid potential collisions. There have been quite a number of protocols being proposed in this category, representative ones of which are [76], [16], [88]. Multiple access with collision avoidance (MACA) [76] uses three-way handshaking to solve the hidden node problem. A node that has data to send transmits a short RTS packet. All nodes within one hop of the sending node hear the RTS and defer their transmission. The destination responds with a CTS packet. All nodes within one hop of the destination node hear the CTS packet and also defer their transmission. On receiving the CTS, the transmitting node assumes that the channel is acquired and initiates the data transmission. The hidden node problem is not completely solved by this scheme, but is avoided to a large extent. Several schemes have been proposed to enhance the RTS/CTS handshaking mechanism, the details of which can be found in [48] and [49]. Some other variations are MACAW [16], MACA/PR [88], and MACA-BI [124], just to name a few. In sensor networks, we find asymmetric communication, high message loss, short messages (e.g., sending a temperature value), and that the interference range is greater than the effective communication radius. These features make solutions that use control packets costly for sensor networks and, therefore, may not be used.

The *distributed coordination function* (DCF) in the IEEE 802.11 wireless local area network (LAN) standard is the basic access method for 802.11. DCF is based on CSMA/CA and uses optional RTS/CTS handshaking to reduce packet collision. The DCF functions as follows: Before initiating a transmission, a station senses the channel to determine whether or not another station is transmitting. If the medium is sensed idle for a specified time interval, called the *distributed interframe space* (DIFS), the station is allow to transmit. If the medium is sensed busy, the transmission is deferred until the ongoing transmission terminates. A slotted binary exponential backoff technique is used to arbitrate the access: a random backoff interval is uniformly chosen in $[0, \widehat{\mathrm{CW}} - 1]$ and used to initialize the backoff timer, where $\widehat{\mathrm{CW}}$ is the maximum contention window. The backoff timer is decreased as long as the channel is sensed idle, stopped when a transmission is in progress, and reactivated when the channel is sensed idle again for more than DIFS. When the backoff timer expires, the station attempts transmission at the beginning of the next *slot time*. Finally, if the data frame is successfully received, the receiver initiates the transmission of an acknowledgment frame after a specified interval, called the *short interframe space* (SIFS), which is less than DIFS. If an acknowledgment is not received, the data frame is presumed to be lost and a retransmission is scheduled. The value of $\widehat{\mathrm{CW}}$ is set to $\mathrm{CW}_{\min}$ $(=32)$ in the first transmission attempt and is doubled at each retransmission up to a predetermined value $\mathrm{CW}_{\max}(=256)$. In addition to physical channel sensing, virtual carrier sensing is achieved by using the network allocation vector (NAV) fields included in the packets. NAV indicates the duration of the current transmission. All nodes that hear the RTS or CTS message back off an amount of time indicated in NAV before sensing the channel again.

While the IEEE 802.11 standard and other related schemes were designed mostly for LANs, they are not directly applicable to sensor networks. In particular, Woo and Culler [40] observed that IEEE 802.11 does not achieve sufficient multihop fairness, energy efficiency, and bandwidth utilization in *motes*—a sensor network prototype that is being developed at the University of California at Berkeley. A number of solutions were proposed to deal with implementation issues. First, for the sake of energy savings, the authors argue that listening on the channel throughout the backoff period as performed by 802.11 is not energy efficient. Alternatively, they propose that the backoff timer should not be paused if the channel is sensed busy during the backoff period. In this way, the radio module of the sensor can be turned off during the backoff time to save energy. Second, to reduce energy consumption as well as to improve bandwidth utilization, the authors advocate to omit the acknowledgment phase and implicitly induce whether a data packet has been received by the receiver through overhearing whether the receiver forwards that packet or not. However, the overhearing approach may not always give accurate acknowledgment information because: a) the receiver may not necessarily forward the packet and b) the sender may not tell whether the overheard packet corresponds to the packet it just sent (as the receiver may have altered the packet or aggregated packets). Third, the authors choose to drop the RTS/CTS handshake mechanism and only use a simple CSMA $+$ random backoff scheme. The rationale behind such a choice is the observation that the typical data packet size is of the same order of the RTS/CTS packet size and, hence, removing the RTS/CTS overhead fully offsets the potential throughput penalty by data packet corruption caused by the hidden terminal problem. Finally, an adaptive rate control was used to provide fairness to multihop flows in term of end-to-end throughput.

All the above contention-based MAC protocols are subject to the open challenge of providing a statistical bound on the real-time requirement. Due to the distributed and random backoff nature, contention-based MAC does not strictly guarantee the priority order of packets from different nodes. For example, two high-priority packets may collide and cause each node to back off, while a third node may send out a low-priority packet when the other two nodes are in the backoff phase. It is necessary to bound the probability of priority inversion in order to establish statistical end-to-end delay guarantees.

4) Hybrid MAC Protocols: Several MAC protocols, such as power controlled multiple access protocol (PCMA) [97] and dual busy tone multiple access protocol (DBTMA) [43], take advantage of the busy tone and the RTS/CTS mechanism and can be viewed as hybrid schemes. In PCMA [97], the power control information is piggybacked on the request-power-to-send (RPTS) and acceptable-power-to-send (APTS) packets. The RPTS/APTS handshake operation occurs in the data channel and precedes the data transmission. After the successful reception of the data, the receiver sends back an ACK packet confirming its reception. A noise tolerance advertisement or busy tone is periodically pulsed by each receiver in the busy tone channel, where the signal strength of the pulse indicates the tolerance to additional noise. A potential transmitter first “senses the carrier” by listening to the busy tone for a minimum time period to detect the upper bound of its transmit power for all control (RPTS, APTS, ACK) and data packets. The major advantages of the RPTS/APTS handshake mechanism are: a) it has the same semantics of the RTS/CTS handshake mechanism and b) it can also be used to determine the minimum transmission power required for successful packet reception at the receiver.

5) Challenges for MAC Technology in Sensor Networks: Sensor networks provide a different computation and communication infrastructure from those for traditional wireless networks. Those differences originate not only from their physical characteristics, but also from their typical applications. For example, physical characteristics include the large scale of deployment, limited computing capability, and constraints on power consumption. Typical applications include tracking objects or detecting events, which are seldom emphasized in mainstream traditional wireless networks.

As a result, the requirements for the MAC layer of a sensor network are noticeably different from those for traditional networks. The major requirements for the MAC layer in a sensor network are as follows.

- *Real-time or Quality-of-Service (QoS) requirements:* Sensor networks are often deployed in a physical environment and are expected to interact with the environment. Therefore, the timely detection, processing, and delivery of information are often indispensable requirements in a sensor network application. As the base of the communication stack, the MAC layer should support real-time guarantees or QoS features.
- *Decentralized:* Most algorithms running in sensor networks need to be decentralized. This is due to both the large scale of the network and the intrinsic unreliability of any single node in the network. Consequently, the MAC layer needs to run decentralized algorithms.
- *Power aware:* In the design of a MAC protocol for sensor networks, the power limitations need to be taken into consideration. This has two direct implications. One is that the MAC protocol needs to be mindful that the power may not always be available. This could be because the power management service has put the node to *sleep* to save power, or the node has actually run out of power. The other is that the MAC protocol needs to save power consumption as much as possible. For example, the MAC protocol may want to avoid excessive collisions, continuous listening, and long-range communication.
- *Flexibility:* Sensor networks are often application specific. While there are typical applications for sensor networks, different applications still exhibit peculiarities on their usage pattern of the network. As a result, the MAC layer of a sensor network needs to be flexible enough to accommodate a variety of network traffic patterns—rate-based or bursty, reliable or best effort, and so on.
- *Balance among multiple metrics:* The MAC design for sensor networks needs to accomplish a balance among a number of metrics. This balance might be more important than the performance on any individual metric. In an unbalanced design, a protocol that performs excellent on one performance metric in lab experiments could observe surprising performance degradation in real environments. For example, a protocol can use a smart scheme to save power. However, if this scheme does not consider other metrics, such as the real-time guarantee or reliability of the packet delivery, it could not only hinder the performance on other metrics, but also degrade on the performance of power saving. For example, if the node turns off the radio component too often, some packets may be lost and more retransmissions could happen, which result in an even greater power consumption.

With these requirements in mind, we can evaluate current MAC layer technologies and consider whether they are suitable for sensor networks.

TDMA is a promising technology because it provides fair usage of the channel and, if equipped with an adequate scheduling algorithm, could also avoid collisions. But many TDMA protocols use global information to do scheduling, which render those protocols to be impractical in general sensor networks. Besides, some of the protocols still have collisions, and it is quite difficult to control the collisions to the degree that does not hurt the guarantee of timeliness. These issues make it difficult for existing TDMA protocols to be broadly used in sensor networks.

Collision-free protocols are surely noteworthy because they save power by eliminating collisions. A good collision-free protocol can also potentially increase the throughput, reduce the delay, and provide real-time guarantee. A problem in a large class of current collision-free protocols is the use of multiple channels [24]. This imposes a nontrivial requirement on the hardware of the nodes in a sensor network. Further study is needed to tell whether the performance gain would overcome the increased cost of hardware. Another concern is the complexity of the protocol. Normally, simple protocols are preferred because of the limited computing capability of nodes in the network.

Contention-based protocols often have difficulty in providing real-time guarantees. As mentioned above, collisions also waste energy. However, there have been some advances in this area which can largely mitigate chances of collisions and reduce power consumption [137]. This could be useful in some applications where predictability is less critical and power consumption is the main concern. On the other hand, for the collision-based protocols to be successfully used in sensor networks, a well-defined statistical bound is still needed.

In summary, existing wireless MAC protocols focus more on optimizing system throughput and do not adequately consider the requirements of sensor networks. The key challenge remains to provide predictable delay and/or prioritization guarantees while minimizing overhead packets and energy consumption.

B. Network Layer

1) Ad-Hoc Routing Protocols: The literature in ad-hoc routing is vast and rich, and we will only summarize existing work most relevant to wireless sensor networks. We roughly classify routing protocols using the following taxonomy: 1) flat routing and 2) hierarchical routing. In flat routing, every node has the equal responsibility of maintaining routing information and relaying packets. Routing algorithms in this category can be further classified into proactive, reactive, and geographic routing.

a) *Proactive routing:* These algorithms maintain routes continuously for all reachable nodes. They require periodic dissemination of route updates. The destination sequenced distance vector routing (DSDV) protocol [104], the adaptive distance vector routing protocol [22], the path-finding algorithms (PFA) [99], and the wireless routing protocol (WRP) [98] fall in this (sub)category.

b) *Reactive routing:* These algorithms establish and maintain routes only if they are needed for communication. New routes are acquired when a connection is to be established and maintained through the lifetime of the connection, even in the presence of topology changes. Representatives in this (sub)category are Gafni and Bertsekas's algorithm [50], the dynamic source routing (DSR) protocol [71], the temporally ordered routing algorithm (TORA) [103], the associativity-based routing (ABR) protocol [126], the signal stability-based routing (SBR) protocol [45], the location aided routing (LAR) algorithm [78], the power aware routing protocol [117], and the ad-hoc on-demand distance-vector (AODV) protocol [112]. In particular, ABR [126] and SBR [45] attempt to build routes that traverse links with high signal strength stability and/or location stability, and the power aware routing protocol [117] explores the issue of increasing network lifetime by using power-aware metrics for routing. LAR [78], on the other hand, uses location information (obtained through GPS) to generate *request zones* in which there is a high probability of finding the destination node.

c) *Geographic routing:* As the name suggests, geographic routing protocols such as GPSR [77] utilize location in routing decisions. Specifically, GPSR forwards a packet to a neighbor node if: 1) it has the shortest geographic distance to the packet's destination among all immediate neighbors and 2) it is closer to the destination than the forwarding node. When such nodes do not exist, packets can be routed around the perimeter of the void region. The only state on each node maintained by GPSR are the locations of immediate neighbors, which is proportional to the density instead of the size of the network. As a result, GPSR is especially suitable for sensor networks that support location-addressed communication. Location-addressed communication means that GPSR can work without a location directory service, which could introduce extra management and communication overhead. The high density in sensor networks leads to a high success probability for GPSR to find a "straight" path from source to destination resulting in efficient communication.

In the category of cluster-based routing, the k-cluster-based routing scheme [129], the zone routing protocol (ZRP) [56], the spine routing framework [41], the adaptive clustering scheme [128], [87], and the min ID/max degree scheme may have received the most attention. In k-cluster-based routing, the network is dynamically organized into k clusters, where all the nodes in a cluster can be reached from any other node within the cluster in k hops. Then, Dijkstra's shortest path algorithm is used to build the routing table. Similar to k-cluster-based routing, ZRP [56] enables nodes to maintain their own *routing zones*—clusters of nodes that can be reached along paths that are at most n hops away. As far as routing is concerned, ZRP uses a hybrid routing strategy (proactive intrazone routing + demand-based interzone routing) to balance the tradeoff between proactive and reactive routing.

The spine routing framework [41] was built upon the notion of *spines* (virtual backbones)—a set of relatively stable and connected nodes such that every node is either part of the spine or one hop away from a node in the spine. While the framework does not completely specify its routing algorithm, it presents two approaches: clustered spine routing and partial spine routing. The framework presents a dedicated backbone for information dissemination, but spine maintenance is costly and introduces significant control traffic when updates are made.

The adaptive clustering scheme proposed in [128], [87] uses lowest node IDs to divide the network deterministically into clusters, with the intention to limit reorganization required in the case of node mobility. No effort was made to utilize the hierarchy and to improve routing efficiency. In the min ID/max degree scheme, min ID and max degree are used to group nodes into clusters within which two nodes are at most two hops away. In the min ID algorithm, each node has a globally unique ID. Neighboring nodes exchange node ID information, and a node with minimum ID among all its neighbors will declare itself as the cluster head and assign its ID as the cluster ID. A subsequent node will declare itself as a cluster head if and only if its neighbors with a lower ID belong to other cliques. The algorithm ensures that by the end of the clustering process, each cluster head has the lowest ID in the cluster and is one hop away from any other node. The max degree algorithm exploits a similar idea. Each node broadcasts the list of nodes it can hear. A node is elected as a cluster head if it has the maximal node degree among all the "uncovered" neighbor nodes, where a uncovered node is one which does not yet have an elected cluster head. The max degree algorithm has the advantage of using topological information to obtain a smaller number of clusters, but as compared to the min ID algorithm, is relatively sensitive to topology change.

2) *Multicast and Anycast:*

Group coordination in sensor networks requires reliable and real-time multicast and anycast communication. Such services may be based on geographic areas.

- Area multicast delivers a message to every node in a specified area. Area multicast can be used to register for an event or send a query to an area, or for coordination among nodes in a local group.
- Area anycast delivers a message to at least one node in a specified area. Area anycast can also be used for sending a query to a node in an area. The node can initiate group formation and coordination in that area.

The dynamics and wireless nature of sensor networks make multicast and anycast particularly challenging problems. While area multicast has been investigated for mobile ad-hoc networks (e.g., geocast [79]), there has been relatively little research on real-time multicast and anycast in wireless sensor networks.

3) Challenges for Routing Technology in Sensor Networks: While many of the ideas found in the above routing algorithms may be modified for sensor network routing, there are enough significant differences that preclude their direct use. In particular, sensor networks are very dynamic, nodes join and leave the network regularly, there is high message loss, and there are real-time constraints. These facts mean that solutions which rely on routing tables that contain global states are very costly and may not work at all. In addition, some of the above solutions require too much state to be retained at each node. The limited memory of a device in a sensor network precludes solutions that require large routing tables.

The large scale and high density of sensor networks typically prohibit solutions that rely on flooding. Also, due to the large scale and data-centric characteristics of sensor networks ID-less routing is typically used. Here, geographic location is more important than a specific node's ID. For example, in tracking an object the application only cares where the object is, not which nodes are reporting the data.

Providing end-to-end real-time guarantees is a challenging problem in sensor networks. Due to the amount of needed state information and the signaling overhead, reservation schemes are unlikely to scale well in sensor networks. Instead, timing guarantees should be achieved with minimum state information and end-to-end signaling. The routing protocol should be adaptive to avoid unpredictable congestion and holes in the network.

SPEED [60] is an adaptive, location-based real-time routing protocol that aims to reduce the end-to-end deadline miss ratio in sensor networks. SPEED is based on a notion of communication speed. For each node A, the communication speed of its neighbor B for a packet is defined as the difference between A and B's distances to the packet's destination divided by the communication delay between A and B. SPEED bounds the end-to-end communication delay by enforcing a lower bound on the communication speed on every hop. Given a speed bound S, the end-to-end delay of a packet is bounded by dis/S, where dis is the distance from the source to the destination. Similar to geographic routing, each node only maintains the states of one-hop neighbors. For each one-hop away neighbor, a node records its location and delay. At the core of SPEED are a set feedback-based adaptation algorithms that enforce a per-hop speed in the face of unpredictable traffic. The first adaptation mechanism is a neighborhood feedback loop on each node that periodically computes the probability of forwarding a packet to a neighbor based on its measured speed in the last sampling period. The feedback loop ensures that only the neighbors whose speed is higher than S are eligible for receiving packets, and (busier) neighbors with lower speeds get lower probabilities of receiving packets. When none of its neighbors satisfies the required speed bound, a node informs upstream neighbors to redirect packets away from it, a process called back pressure rerouting. The back pressure can propagate upstream until it reaches outside the congestion region or the source. The combination of neighborhood feedback loops and back pressure rerouting enforces the per-hop deadline in steady states and reduces the end-to-end deadline miss ratio. Simulation experiments given in [60] show that SPEED can achieve significantly lower deadline miss ratio than geographic routing, DSR, and AODV in face of sudden congestion. Meanwhile, SPEED's number of overhead packets is comparable to geographic routing and significantly smaller than DSR and AODV. SPEED demonstrates that localized feedback control is a promising approach for real-time communication in sensor networks. Remaining challenges in this direction include establishing stability analysis, providing statistical guarantees on end-to-end delays, and supporting QoS by enforcing different required speeds for different data flows.

C. Transport Layer

Not until recently has the research community started to address the problem of maintaining reliable end-to-end communication in wireless ad-hoc networks [27], [52], [53], [66], [96]. In particular, several studies [27], [52], [53], [66] have shown that TCP performance in terms of attainable throughput and fairness deteriorates significantly in ad-hoc networks. This is attributed to the following.

a) *Fairness of the underlying MAC protocol:* It has been shown in [123] that as the widely adopted IEEE 802.11 MAC protocol cannot arbitrate bandwidth among competing connections at the link level, it cannot achieve either long-term or short-term fairness for TCP connections.

b) *Link failure due to mobility:* Due to frequent and dynamic link failures (arising from a mobile host moving out of/into the transmission range of its neighbor), it is very difficult, if not impossible, for routing protocols to keep their routing cache updated. Consequently, packets may be routed based on stale route information and dropped at some intermediate node (which does not know how to route these packets). As reported in [67], the majority of packet loss in ad-hoc networks are due to outdated entries in the routing cache. However, a TCP sender cannot recognize the cause of packet loss (congestion or link failure), so considers packet loss as an indication of congestion, and invokes the additive increase multiplicative decrease (AIMD) algorithm. This results in throughput degradation. While nodes in sensor networks are generally static, a similar link failure problem arises due to power saving strategies that frequently turn nodes off to conserve energy consumption. Consequently, connectivity becomes intermittent. TCP may conclude that congestion occurred when, in fact, the problem is due to a change in available routes produced by power management.

Holland *et al.* [66] and Chandran *et al.* [27] propose a solution to alleviate this problem using explicit feedback, called *explicit link failure notification (ELFN)*. When a router detects a link failure, it notifies the sender with an ELFN message.[1] When a TCP sender receives an ELFN message, it freezes the retransmission timer, stops sending (new or retransmitted) packets, and enters *standby* mode. In standby mode, the TCP sender neither reduces its window size upon receipt of duplicate ACKs nor incurs timeouts, and is hence "immune" to the AIMD effect. To recover from standby mode, the TCP sender may periodically send packets to probe the network until it receives a new ACK (at which point it knows a new route has been located).

c) *Coupling effects of the forward and reverse paths:* Mobile ad-hoc networks exhibit network asymmetry, due to the following reasons. i) The noise that results from the interference is usually location dependent and, hence, two entities that are engaged in communication may observe different signal-to-noise ratios. ii) A mobile host that uses IEEE 802.11 as the MAC protocol is half-duplex (i.e., cannot send and receive at the same time). As a result, connections destined for different directions may contend with each other for the resources. iii) If DSR or some other multipath routing algorithm is used as the underlying routing protocol, the forward and reverse paths of a TCP connection may be different and may exhibit different congestion and connectivity characteristics. As a TCP sender relies on the timely return of ACKs to advance its transmission window, ACK losses (as a result of congestion and link failure) on the reverse path have an adverse impact on the TCP performance. Although AODV enforces bidirectional routes, the same problem exists if multiple paths of the same number of hops are present. Two paths with the same hop count may be used as the forward and reverse paths, respectively.

Zheng *et al.* [138] explore the impact of the reverse path characteristics on data transport on the forward path in ad-hoc networks. They propose an end host approach, called *TCP with retransmitted ACK* (*TCP RACK*), to eliminate the effect of path asymmetry on the TCP performance. The key idea is to selectively retransmit "important" ACKs when the receiver has sent all eligible ACKs, but has not received any new data packets, to determine (at the sender side) whether a received ACK is a normal (nonduplicate or duplicate) ACK or a retransmitted ACK (generated by TCP RACK) and to take appropriate congestion control actions.

1) Challenges for Transport Layer Technology in Sensor Networks: To the best of our knowledge, there exists little work on transport layer issues in sensor networks. In sensor networks, usually there are a few sinks to which packets are directed, and data are often redundant and correlated to the same physical phenomenon. As a result, the major objective of data transport is no longer to maximize the raw data throughput per unit bandwidth, but instead to maximize the information throughput per unit energy. To avoid serious congestion near the sinks, data should be aggregated along the route toward the sinks.

Traditional transport-layer protocols such as TCP or UDP allow multiplexing of several ports over the same IP host. In contrast, nearby sensor network nodes are individually unreliable and collectively interchangeable. Thus, a transport layer connection is more likely to aggregate a cluster of nodes into the same virtual connection endpoint. New semantics need to be proposed for communication with such clustered endpoints, as well as new mechanisms for collective congestion control, packet ordering, data aggregation, and reliable delivery.

Data in sensor networks usually has a real-time nature. It is important to receive timely data even at the expense of other connection attributes such as throughput. Connection throughput may be limited by available network bandwidth which depends on network congestion levels. Traditional TCP congestion control attempts to alleviate congestion by slowing down the sending rate of the source. Clustered endpoints, discussed above, offer another means for congestion control, namely by controlling the level of temporal or spatial data aggregation within the source cluster. In general, congestion can be controlled by varying the *quality* of delivered information as a function of network load. Protocols are needed that allow defining quality of information in generic terms and automate its control at the transport layer in response to network conditions using mechanisms such as in-network aggregation.

In-network data aggregation methods are usually application-dependent. Experimental results [61] and analysis [80] have shown that in-network aggregation can achieve lower energy consumption and higher data delivery ratio than address-centric communication, although in-network aggregation may increase end-to-end delays. Mechanisms for in-network aggregation include triangulation in vehicle tracking (e.g., use of at least three sensor-reported timestamps and the locations of sensors to determine the exact location and speed of targets that are being tracked) and nested queries [20] (e.g., use of a light sensor reading to locally trigger queries on correlated image sensors), and require data be named based on application semantics. Directed diffusion names data by application-specific attribute tuples. A node can announce an interest on a particular type of data, and nodes whose data match the interest respond to the requester along the interest gradient. Filters can be set up on nodes to cache and aggregate data inside the network. Directed diffusion, however, is a network-layer protocol which does provide well-defined transport-layer abstractions.

A new problem that arises in sensor networks is that of migratory endpoints. It is common in sensor networks to address communication to destinations defined by their

[1] The ELFN message may be a ICMP *host unreachable* message or the *route failure* message that is already defined in DSR or AODV.

attributes and not by a logical node identifier. For example, one may wish to communicate with those nodes that are in the vicinity of a specific moving target such as a tank or an animal. The identity of the nodes in the vicinity of the target changes when the target moves. Yet, from the perspective of the programmer, it is desirable that the abstraction of a unique end-to-end connection be maintained. Such a connection would have its own state and may serve as a virtual communication port of sensor-network applications. One or both endpoints of such a connection may migrate across sensor nodes in response to events in the external environment. Transport layer protocols are needed that allow defining connection endpoints using environmental attributes, and handle the migration of those endpoints in response to environmental changes.

For example, in EnviroTrack [4], the transport layer provides an abstraction that allows applications to communicate end-to-end with abstract entities instead of individual sensors. This type of transport layer may be more appropriate for sensor networks than a traditional TCP layer.

The interplay between power management and transport-layer protocols offers additional fundamental challenges in sensor networks. Transport layer protocols on the Internet, such as TCP, have devoted much attention to congestion control, the underlying assumption being that network bandwidth is a scarce resource. Bandwidth consumption is fairly shared among all sources which need it. In contrast, in sensor networks, energy is another scarce resource. Nodes that are low on energy should be avoided the same way as nodes that are congested. One challenge for transport-layer protocols would be to modulate the rates of the senders to accommodate not only the bandwidth constraints, but also the energy constraints in the network.

Finally, fairness has often been deemed an important consideration in the design of transport layer protocols. In contrast, sensor network traffic is more likely to be tiered. Transport layer protocols should be designed to allow sources that are more important to consume a disproportionate amount of bottleneck resources at the expense of sources that are less important. Implementing such distributed prioritization among cooperative sources is an important challenge of sensor network transmission control protocols.

D. Multilayer Issues

There are two categories of power-saving approaches: 1) a set of leader nodes are selected (on, for example, a rotational basis) to be awake to maintain network connectivity, while other nodes can be put to sleep (and awakened periodically to check for activity) and 2) instead of having each device transmitting at its maximum power, a minimal transmission power is determined to maintain a minimally connected network; the resulting network is an always-on network with its devices transmitting at the minimum possible power. The former approach is termed as power management, and is the focus of Section III-D1. The latter approach is termed as topology/power control, and will be treated in Section III-D2. Since real time is also a multilayer issue, this topic is treated in Section III-D3.

1) Power Management: Power management is motivated by the observation that the energy consumption of a mobile node in the idle state is only slightly smaller than that in the transmission or receiving state [47]. There have been several energy-conserving protocols [28], [135],[62] for ad-hoc networking environments. Their key idea is to take advantage of route redundancy and turn off radios that will not affect network connectivity. A common approach is to construct an overlay backbone composed of a small number of active nodes to route all multihop packets, while letting other nodes sleep when they do not send or receive data. However, different strategies have been explored to build such an overlay backbone.

- LEACH [62] adopts a hierarchical strategy. Nodes are divided into clusters with a cluster head serving as the gateway to the rest of the network. In the cluster setup phase, each node invokes a randomized algorithm to decide whether it wants to serve as a cluster head. If yes, it announces itself as a cluster head; otherwise, it joins the cluster head with the strongest signal strength. After clusters are formed, all members of a cluster send their data to their head based on a TDMA schedule, and the head aggregates the data from the members and sends it to the base stations. The network periodically enters a new setup phase to form new clusters with possibly new cluster heads.
- GAF [135] also follows a hierarchical strategy, but its clusters are based on geography. Nodes are divided into fixed geographic grids each with a dynamically elected leader. All multihop packets are routed through the grid leaders. Under the assumption that the radio radius R is known and fixed, the grid size is small enough to ensure that the maximum distance between any pair of nodes in adjacent grids is within the transmission range of each other. Since each node in a grid can directly communicate to any nodes in any neighbor grids, nodes in a grid are equivalent to each other for routing packets from neighboring grids. The leader election scheme in each grid takes into account battery usage at each node, and a sleeping node wakes up periodically to attempt to elect itself as an active node. A simulation study shows that GAF extends the network lifetime by 30%–40%. However, experiments have shown that radio transmission range is highly probabilistic and dependent on the environment [51]. Future extensions are needed to handle such situations.
- SPAN [28] forms an overlay backbone in a peer-to-peer fashion. SPAN [28] is a distributed and randomized algorithm, and does not require any location information. In SPAN, nodes can make local decisions on whether they should sleep or join a overlay backbone as a coordinator. The nodes that choose to stay awake and maintain network connectivity/capacity are called coordinators. The rule for electing coordinators is that if two neighbor nodes of a noncoordinator nodes cannot directly communicate with each other nor through one

or two coordinators, then this node volunteers to be a coordinator. The information needed for electing oneself as a coordinator is exchanged among neighbors via HELLO messages. The coordinator announcement is broadcast, based on a delay interval reflecting the "benefit" that each neighbor will perceive and taking into account the total energy available. Since all the decisions are made locally, the solution scales well. The results show that SPAN saves energy by a factor of 3.5 or more, and increases the lifetime of the network by a factor of 2. Note that the energy consumption rates are not constant, and as nodes die out, a larger fraction of nodes need to stay awake. This "drain out" effect accounts for this discrepancy.

The above approaches assume that the network is densely populated. In addition, they are evaluated in scenarios where a group of nodes are dedicated to forwarding data packets, while source and sink nodes are static and always awake. There exists some work [111] that studies the performance of IEEE 802.11 PSM in a wireless LAN environment. However, little is known about how IEEE 802.11 PSM operates in multihop wireless networks.

2) Topology Control: $\mathrm{CBTC}(\alpha)$ [84] is a two-phase algorithm in which each node finds the minimum power p such that transmitting with p ensures that it can reach some node in every cone of degree α. The algorithm has been analytically shown to preserve the network connectivity if $\alpha < 5\pi/6$. It also ensures that every link between nodes is bidirectional. Several optimizations to the basic algorithm are also discussed, which include: i) a *shrink-back* operation can be added at the end to allow a boundary node to broadcast with less power, if doing so does not reduce the cone coverage; ii) if $\alpha < 2\pi/3$, asymmetric edges can be removed while maintaining the network connectivity; and iii) if there exists an edge from u to v_1 and from u to v_2, respectively, the longer edge can be removed while preserving connectivity, as long as $d(v_1, v_2) < \max\{d(u, v_1), d(u, v_2)\}$. An event-driven strategy is proposed to reconfigure the network topology in the case of mobility. Each node is notified when any neighbor leaves/joins the neighborhood and/or the angle changes. The mechanism used to realize this requires state to be kept and messages exchanged among neighboring nodes. The node then determines whether it needs to rerun the topology control algorithm.

Based on theoretical findings in [55], Narayanaswamy *et al.* have developed a power control protocol, called COMPOW [101]. The authors argue that if each node uses the smallest common power required to maintain the network connectivity, the traffic carrying capacity of the entire network is maximized, the battery life is extended, and the contention at the MAC layer is reduced. In COMPOW, each node runs several routing daemons in parallel, one for each power level. Each routing daemon maintains its own routing table by exchanging control messages at the specified power level. By comparing the entries in different routing tables, each node can determine the smallest common power that ensures the maximal number of nodes are connected. The major drawback of COMPOW is its significant message overhead, since each node runs multiple daemons, each of which has to exchange link state information with the counterparts at other nodes. COMPOW may also result in large transmission power in the case of an extremely nonuniform node distribution.

Rodoplu *et al.* [110] introduced the notions of *relay region* and *enclosure* for the purpose of power control. For any node i that intends to transmit to node j, node j is said to lie in the *relay region* of a third node r, if node i will consume less power when it chooses to relay through node r instead of transmitting directly to node j. The *enclosure* of node i is then defined as the union of the complement of relay regions of all the nodes that node i can reach by using its maximal transmission power. It was shown that the network is strongly connected if every node maintains links with the nodes in its enclosure. A two-phase distributed protocol was then proposed to find the minimum power topology for a static network. In phase one, each node i executes a local search to find the enclosure graph. This is done by examining nodes that the node can reach using its maximal power and keeping only those that do not lie in the relay regions of previously found nodes. In phase 2, each node runs the distributed Bellman-Ford shortest path algorithm on the enclosure graph, using the power consumption as the cost metric. When the node completes phase 2, it can either start data transmission or enter the sleep mode to conserve power. To deal with limited mobility, each node periodically executes the distributed protocol to find the enclosure graph. This algorithm assumes that there is only one data sink (destination) in the network, which may not hold in practice. Also, an explicit propagation channel model is needed to compute the *relay region*.

Li *et al.* [86] present a minimum spanning tree-based topology control algorithm, called local minimum spanning tree (LMST), for wireless multihop networks. In LMST, each node builds its local minimum spanning tree independently and only keeps on-tree nodes that are one hop away from its neighbors in the final topology. They analytically prove that: 1) the topology derived under LMST preserves the network connectivity; 2) the node degree of any node in the resulting topology is bounded by 6; and 3) the topology can be transformed into one with bidirectional links (without impairing the network connectivity) after removal of all unidirectional links.

Ramanathan *et al.* [107] present two centralized algorithms to minimize the maximum power used per node while maintaining the (bi)connectivity of the network. Algorithm CONNECT is a simple greedy algorithm that iteratively merges connected components until there is only one. Augmenting a connected network to a biconnected network is done by Algorithm BICONN-AUGMENT, which uses the same idea as in CONNECT to iteratively build the biconnected network. In addition, a post-processing phase can be applied to ensure per-node minimality by deleting redundant connections. These two algorithms require global information and cannot be directly deployed in the case of mobility. To deal with limited mobility, the authors introduced two distributed heuristics: LINT and LILT. In

LINT, each node is configured with three parameters: the "desired" node degree d_d, a high threshold on the node degree d_h, and a low threshold d_l. Every node periodically checks the number of active neighbors and changes its power level accordingly, so that the node degree falls within the thresholds. LILT further improves LINT by overriding the high degree threshold when the topology change (indicated by routing updates) leads to a network partition.

In spite of all the above research efforts, several research challenges remain that include study of the impact of power management on end-to-end delay, and the proper tradeoff between them. For example, when emergent events such as fire occur, the overlay backbone and topology should adapt to satisfy tight timing requirements.

3) Real Time: Real time is another issue that crosses all layers in the communication stack. RAP [89] is a multilayer real-time communication architecture for sensor networks. Communication on RAP is addressed by location. Applications specify queries and register for events in a geographic location/area together with their timing constraints. The query and event APIs provide a high-level abstraction to applications by hiding the specific location and status of each individual node. These APIs allow applications to specify the timing constraints of queries. The underlying layers of RAP are responsible for orchestrating the sensing and communications of relevant sensors to accomplish all query and event services. For example, the following API call registers a virus.count query for a virusFound event. If any viruses are found in a rectangular area with coordinates (0,0 100 100), the network returns the average density of the viruses of the $2 * 2$ square area centered at the event location (X_e, Y_e) every 1.5 s. Every reading should reach the base station within an end-to-end deadline of 5 s.

```
registerEvent {
 virusFound(0, 0, 100, 100),
 query {
   virus.count,
   area=(Xe − 1,Ye − 1, Xe + 1, Yevent + 1),
   period = 1.5, deadline = 5,
   base = (100 100)
 }
}
```

A query or event is sent to every node in the specified area. Query results are sent back to the base station based on its location provided by the query or event registration.

Communication in RAP is supported by a scalable and efficient protocol stack, which integrates a transport-layer location-addressed protocol (LAP), a geographic routing protocol [77], a velocity monotonic scheduling (VMS) layer, and a contention-based MAC that supports prioritization [1]. A cornerstone of RAP is a velocity monotonic scheduling (VMS) policy. VMS is based on a notion of packet requested velocity that reflects both distance and timing constraints of sensor network communication. Each packet can make its end-to-end deadline if it can move toward the destination at its requested velocity. VMS reduces end-to-end deadline miss ratios of sensor networks by giving higher priority to packets with higher requested velocities. The requested velocity of a packet can be computed statically or dynamically. The static VMS computes a fixed requested velocity at the sender of each packet. Assume a packet is sent from a sender at (x_0, y_0) to a destination at (x_d, y_d), and has an end-to-end deadline D s, then SVM sets its requested velocity to: $V = \mathrm{dis}(x_0, y_0, x_d, y_d)/D$ where $\mathrm{dis}(\mathrm{x0}, \mathrm{y0}, \mathrm{xd}, \mathrm{yd})$ is the geographic distance between $(\mathrm{x0}, \mathrm{y0})$ and $(\mathrm{xd}, \mathrm{yd})$. The requested velocity of a packet is fixed throughout the network.

The dynamic VMS recalculates the requested velocity of a packet upon its arrival at each intermediate node. Assume a packet arrives at a node at location (x_i, y_i); its destination is at (x_d, y_d); it has an end-to-end deadline D s, and its elapsed time, i.e., the time it has been in the network, is T_i s; its requested velocity V_i at (x_i, y_i) is set to $V_i = \mathrm{dis}(x_i, y_i, x_d, y_d)/(D - T_i)$. The requested velocity of a packet will be adjusted based on its actual progress (i.e., actual velocity). A packet's requested velocity increases if its previous progress toward the destination is slower (e.g., due to congestion) than its previous requested velocity. On the other hand, its requested velocity decreases if it moves faster than its previous requested velocity. This occurs so that packets ahead of schedule can give way to other more urgent packets. The requested velocity is mapped to a MAC-layer priority, which is enforced in a contention-based MAC layer. Simulation experiments show that RAP reduced the deadline miss ratio from 90.0% to 17.9% compared to DSR running over 802.11b. RAP demonstrates that a multilayer, location-based communication stack and velocity-based prioritization can effectively improve real-time performance in sensor networks.

IV. Operating System and Middleware Research Challenges

As detailed in the previous two sections, many challenges for sensor networks exist with regard to the communication aspects of these systems. However, many additional challenges exist at the operating system and middleware layers. These layers are responsible for adding functionality beyond communications, e.g., dealing with distributed resource management, aggregate control, and team formation to support various activities such as tracking objects through the sensor network. After discussing the basic need for a paradigm shift at the OS and middleware layers, we itemize the research challenges for the following topics: single node issues, new task and virtual machine models, context awareness, content-addressable space, distributed control, team formation, and data services. While this is not a comprehensive list of topics, it does identify many of the important topics and illustrates how they differ from traditional distributed systems because of the special constraints found in sensor networks.

Just as for communication, the paradigm shift in distributed computing brought about by the advent of sensor networks requires revisiting the basic operating system abstractions such as tasks and intertask communication, as well as developing support for fundamentally new distributed programming environments. Historically, several paradigms were developed for distributed computing with the purpose of creating appropriate abstractions for distributed application programmers and developing run-time systems that support these abstractions. Examples of successful paradigms include distributed object-oriented computing (e.g., CORBA [131]), group communication (e.g., ISIS [17]), remote procedure calls (RPC [19]), and distributed shared memory (e.g., MUNIN [25]). These paradigms presented convenient new entities of which the programmer's world is composed (e.g., objects and process groups), and implemented mechanisms for their interaction. Current paradigms for distributed computing, however, share in common the fact that their programming abstractions exist in a logical space that does not inherently represent or interact with objects and activities in the physical world. As such, these current paradigms fall short of the requirements of sensor networks.

One main aspect, which sets sensor networks apart from existing approaches to distributed computing, is their need for the integration of objects that live in physical time and space as components in the computational environment of the application [6]. Traditional operating system abstractions, such as processor sharing and virtual memory, stem from the hardware components of the classical machine architecture such as central processors and memory chips. In a system where the basic hardware architecture is inherently distributed and is better viewed as a part of a physical world in which the distributed machine is seamlessly embedded, the basic system abstractions must change. In this section, we review present research directions in system architecture for sensor networks and outline open challenges.

A. Single-Node Challenges

The lowest system support for sensor networks begins at the level of a single node. The severe resource limitations, reliability considerations, real-time constraints, and unpredictability of the environment call for creative implementations of basic kernel functions. New stripped-down kernels must be developed to manage the limited resources of a single sensor-equipped device in a robust manner. TinyOS [64] is perhaps one of the earliest operating system kernels developed exclusively for sensor nodes. With only 178 bytes of code, TinyOS provides support for communication, multitasking, and code modularity. Geared toward communication-intensive applications, it exports the abstraction of components, which can be integrated into structures similar to a protocol graph. Each component consists of command handlers, event handlers, and simple tasks. Communication protocols can be constructed easily in a modular manner by developing the appropriate handlers independently of others. While the notion of modular protocol stacks is not new, a great contribution of TinyOS is to implement such a composable framework within the memory and computing constraints of individual sensor nodes.

B. Tasks and Virtual Machines

Classical operating systems export the abstraction of tasks as schedulable entities that can own computing resources. Tasks are typically thought of as entities that partition a single CPU among multiple resource owners. This view is inherited from multitasking systems built around the premise that the CPU is powerful enough to execute multiple tasks concurrently. In contrast, in sensor networks, this one-to-many relation is reversed. Individual tasks (such as the identification of a given activity in the environment) typically require the collaboration of multiple sensors each of which is a dedicated device with little room for concurrency. Hence, new operating abstractions are needed to support a distributed task notion. Programming support is needed whereby users can write linear code at an appropriate level of abstraction that is executed as a distributed protocol among a group of several cooperative devices whose membership may depend on the physical environment and that meet timing requirements.

Distributed virtual machines have been proposed to provide convenient high-level abstractions to application programmers, while implementing low-level distributed protocols transparently in an efficient manner [118]. This approach is taken in MagnetOS [12], which exports the illusion of a single Java virtual machine on top of a distributed sensor network. The application programmer writes a single Java program. The run-time system is responsible for code partitioning, placement, and automatic migration such that total energy consumption is minimized.

Mate [82] is another example of a virtual machine developed for sensor networks. It implements its own bytecode interpreter, built on top of TinyOS. The interpreter provides high-level instructions (such as an atomic message send) which the machine can interpret and execute. Each virtual machine instruction executes in its own TinyOS task. Code is broken into capsules of 24 single-byte instructions. A send() instruction allows the capsule to be sent to another node as an active message. This provides a mechanism for the dissemination of new code into the network via an infection model. The programmer need not worry about coding for each individual sensor, but rather injects code into a single node, and lets it diffuse into the network in a virus-like fashion.

A somewhat different approach for providing high-level programming abstractions is to view the sensor network as a distributed database, in which sensors produce series of data values and signal processing functions generate abstract data types. The database management engine replaces the virtual machine in that it accepts a query language that allows applications to perform arbitrarily complex monitoring functions. This approach is implemented in the COUGAR sensor network database [21]. A middleware implementation of the same general abstraction is also found in SINA [115], a sensor information networking architecture that abstracts the sensor network into a collection of distributed objects.

While these pioneering efforts have produced novel prototypes of distributed sensor systems with convenient familiar programming interfaces, the final vision for sensor network computing environments is far from being settled. Rather than extending familiar computing paradigms to a new environment, fundamentally different paradigms and programming systems are possible that are inspired by chemical and biological metaphors. One of the most prominent examples of this direction is the work on *amorphous computing* [5]. Amorphous computing environments are those composed of millions of randomly interconnected unreliable computing devices which must coordinate to perform high-level tasks. This is akin to the coordination of cells in a living body to perform specific functions. An analogy can be made between program execution in an amorphous computing environment and the execution of DNA code to produce a complicated biological entity from a single cell. It is well-known that chemical diffusion is the key to cell differentiation in biological systems, which is how complex biological patterns are formed. Hence, a diffusion-based programming paradigm can be used to organize amorphous computing systems in an arbitrarily complex manner. A programming language based on this observation is the growing point language [37]. The main language abstraction is growing points, entities which can diffuse through the network, emit pheromones, or deposit state in the cells they encounter. Pheromones, in turn, can attract or repel growing points as encoded by the program. With appropriate coding, these simple primitives were shown to be sufficient to generate arbitrarily complex deposit patterns [37]. The operating system in this case merely enforces proper diffusion laws associated with the pheromones. The application program merely dictates the growing-point propagation patterns as a static function of pheromone concentrations. An interesting challenge is to develop techniques for reverse engineering the desired end-products into the "genetic code" needed to produce them at run-time. Another challenge is to develop techniques to utilize this genetic paradigm in real-time situations.

C. Context Awareness

Sensor networks also offer exciting new possibilities in designing operating system support for innovative human–computer interaction modes. Humans typically communicate their perceptions using a set of identifiers, which name objects in the physical world that are defined by specific properties perceivable by the human senses. Such communication is impossible in conventional computing environments due to the lack of appropriate sensory devices that would relay information germane to the definition and identification of the object. Sensor networks, however, offer a unique opportunity to leverage myriad available sensing modes (such as temperature, pressure, motion, vibration, humidity, light, sound, magnetic field, position, velocity, and acceleration) to develop a vocabulary and communicate perceptions which relate to the physical world. A computing system with such a capability is called *context aware*.

The need to build distributed sensing, computing, and actuation systems, which share common perceptions with their users about the physical environment has been most clearly articulated in the *sentient computing* project [6]. More generally, context-aware computing systems motivate research into new communication and coordination protocols, as well as new types of programming environments in which the computational and physical environments are seamlessly integrated. For example, in a future environmental protection sensor network, it would simplify application development if programmers could express a physical condition called "fire" and bind processing to it in the sensor network. The processing would monitor such events when and where they occur, communicate their status to specific locations, respond to queries about environmental information at the locations of such events, and possibly perform emergency intervention or report alarms to authorities.

While the full vision of context-aware computing remains a research challenge, much progress has been made on integrating partial awareness of the physical environment into the computing system. In particular, location-awareness has been investigated at length. Starting with the network layer, location-assisted routing protocols have received much attention such as LAR [78] and DREAM [14]. A real-time version of location-based routing was introduced in [89]. For networks relying on identifier-based routing, scalable location services have been proposed to keep track of locations of identified destinations [83]. System prototypes have been developed in which location was an essential attribute of system objects [63]. Most such systems, such as Cooltown [42] and Cricket [106], are geared toward a distributed environment of mobile, networked devices that compose a system in which locations of the participants are known and used to provide new services and functionality. In contrast, in sensor networks, locations will be associated with events in the physical environment that may be of interest to network users. This presents additional challenges, since no devices are associated with such events the way networked PDAs or mobile phones may be associated with human users.

Note that location is only one dimension of the physical world. In a sensor network, this dimension is augmented with other physical attributes of the world to which the network has access such as optical, audio, thermal, and magnetic inputs and measured time. In an ideal scenario, operating system and programming environments should explicitly take them into consideration within some single unified framework. A vision of such a unified framework is presented next.

D. Content-Addressable Space

One main responsibility of a distributed operating system is to define a suitable address space for applications. For example, distributed shared memory systems export a global virtual memory space, which is independent of machine boundaries. Object-based systems export a space of objects. Sensor network requirements suggest a space of addressable entities that are more tightly coupled with the physical world. For example, the distributed operating system might export a space of identifiers, which refer to specific instances of programmer-defined physical conditions monitored in the environment. The paradigm is a variation of what is often called content-addressable networks [109], i.e., networks in which destinations are addressed by their content attributes, not by their machine identity.

In a sensor network, addressable identifiers may be associated with localized entities in the physical environment that the network can sense. For example, the sound, motion, and magnetic signature of a moving vehicle can be associated with an identifier that tags this vehicle and essentially follows it around in the network. Programmers should be able to associate monitoring or other processing code with these identifiers such that execution of this code is triggered by the corresponding environmental stimuli and such that execution occurs only where needed by the physical environment. Logically, one can think of these attached objects as residing on a virtual host, which moves in space with its identifier in a manner decided by the physical environment.

For the special case of relatively static content-addressable destinations, directed diffusion, described in Section III-C, has been proposed the underlying communication scheme [68]. The scheme has been generalized to an infrastructure for attribute-based naming [61]. The infrastructure maintains an object name space in which names are associated with locales which match certain attribute profiles of the external environment. Flexible rules are applied to determine the matches. The framework is integrated with a capability for in-network processing, which may be initiated at the locations where attribute-based matches occur. Hence, for example, the framework allows one to query the network for all the locations where motion is detected, and to initiate monitoring tasks precisely at these locations.

Several challenges still remain in content-addressable networks. For example, what is the most efficient way of propagating interests and queries to matching sensors without resorting to complete broadcast and under deadline constraints? How to maintain bidirectional communication with the content-addressable entity when environmental conditions cause the entity to move? How to efficiently support code mobility? What connection abstractions and transport-layer protocols are needed? How to implement connection end-points when the "area" matching the query contains multiple neighboring sensors? These issues are topics of active research.

E. Distributed Control

Sensor networks differ from traditional computing systems in their massive scale and unattended operation. Self-stabilizing localized algorithms are needed which operate on local information, but collectively produce desired robust global effects [46]. One possible direction is to cast these algorithms as optimization problems (such as energy minimization). This approach is taken in [94] where localized optimization algorithms are developed. Another possibility is to cast them as problems of distributed control. Control theory has been identified as an important tool for stability analysis in complex systems. Hence, integration of control-theoretic foundations with properly designed localized algorithms can lead to a framework in which global requirements can be specified, analyzed, and the ability of the system to converge to the desired global specifications be ascertained. In [120], preliminary results are reported on applying a control-theoretic framework to model the behavior of different localized algorithms for global performance control in real-time environments. The more general problem of analyzing arbitrary protocols in large ad-hoc wireless networks within a control-theory framework remains open.

F. Team Formation

Group management and team formation present fundamental new challenges in sensor networks. Most prior membership and group communication services assume reasonably static systems [3], [18], [44]. Group members in such systems do not have a high turnaround rate. Hence, strong semantics could be achieved such as virtual synchrony [17] where messages are delivered atomically and in order, and all members have consistent membership views. Such semantics are impossible to achieve in sensor networks, where groups are highly dynamic and membership changes occur at a very high rate compared to the time scales of basic algorithm functions such as message transmission. Relaxed, yet meaningful semantics are needed for group communication and coordination functions. New group coordination algorithms are required to maintain novel application-specific group invariants. For example, a group may be formed to track an evader. As the evader moves in the sensor network, the membership of this group changes dynamically to reflect the sensors closest to it. The group must maintain an invariant, namely, it must contain at least three members at any given time for proper triangulation of the evader's position. A group management algorithm will need to provide such guarantees.

Future challenges in sensor network group communication algorithms may also include incorporation of physical properties into the group communication semantics of the embedded system. For example, a group may be required to maintain a given radius such that all nodes falling within that radius must be included reliably in all group multicasts. Alternatively (in the target tracking example presented above), a group management algorithm may need to guarantee a maximum propagation speed. This speed can be defined as the maximum target speed at which the group communication semantics are correctly maintained. Integration of such physical constraints is unique to sensor networks and has not been addressed in previous group communication and membership research. Little work has been done on guaranteeing real-time properties of group management protocols, but such guarantees are required.

G. Data Services

Data communication among different tasks is at the core of modern operating system services. In an address space made of distributed entities created, located, and deleted by activities that take place in a physical environment, data communication abstractions and protocols face fundamental challenges. Traditional point-to-point communication abstractions such as pipes, sockets, and RPC are not suitable in a computing environment where only *collective* information is useful, as opposed to individual sensor state. Programming systems should allow acquisition and exchange of collective information beneath convenient high-level abstractions. Protocols for exporting these abstractions need to consider resource constraints such as power and communication bandwidth, as well as quality of information constraints such as timeliness, staleness, and statistical confidence.

Sensor networks offer new tradeoffs between resource constraints and information quality constraints. Algorithms are needed which exploit such tradeoffs in a manner consistent with application priorities in order to maximize the total sensor network utility. For example, content distribution protocols may be designed whose purpose is to achieve a utility-maximizing balance between network power consumption and the staleness of delivered content. Meeting information quality constraints in the presence of faults is another fundamental challenge. What quality semantics are ensured when data operations may fail due to resource constraints? What failure semantics should be assumed? How to survive violations of the failure hypotheses? So far, these questions remain unanswered in the context of sensor networks offering rich opportunities for future exploration.

V. Summary

Sensor networks represent an exciting new field with great potential for many applications including antiterrorism, smart spaces, numerous military sensing and command and control applications, and entertainment. However, sensor networks are fundamentally different from classical distributed computing technology and ad-hoc networks, although they do build upon these areas. Why and how sensor networks are different is articulated throughout the paper. This paper also highlights the state of the art in sensor networks and presents many open research questions that must be solved. Many of these challenges derive from the severe constraints under which sensor networks operate as well as the fact that they operate in tight coordination and control of physical environments. These factors give rise to the need for new paradigms for both communications services as well as for services supported by operating systems and middleware.

References

[1] I. Aad and C. Castellucia, “Differentiation mechanism for IEEE 802.11,” in *Proc. IEEE INFOCOM Conf.*, Apr. 2001, pp. 209–218.

[2] T. Abdelzaher, J. Stankovic, C. Lu, R. Zhang, and Y. Lu, “Feedback performance control in software services—Using a control-theoretic approach to achieve quality of service guarantees,” *IEEE Contr. Syst. Mag.*, vol. 23, pp. 74–90, June 2003.

[3] T. Abdelzaher, S. Dawson, W.-C. Feng, F. Jahanian, S. Johnson, A. Mehra, T. Mitton, A. Shaikh, K. Shin, Z. Wang, H. Zou, M. Bjorkland, and P. Marron, “ARMADA middleware and communication services,” *Real-Time Syst.*, vol. 16, pp. 127–153, May 1999.

[4] T. Abdelzaher, B. Blum, D. Evans, J. George, S. George, L. Gu, T. He, C. Huang, P. Nagaraddi, S. Son, P. Sorokin, J. Stankovic, and A. Wood, “EnviroTrack: An environmental programming model for tracking applications in distributed sensor networks,” *ICDCS*, submitted for publication.

[5] H. Abelson *et al.*, “Amorphous computing,” Massachusetts Inst. Technol., Cambridge, MIT AI Memo 1665, Aug. 1999.

[6] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper, “Implementing a sentient computing system,” *IEEE Comput.*, vol. 34, pp. 50–56, Aug. 2001.

[7] “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” ANSI/IEEE, ANSI/IEEE Std. 802.11, 1999 Edition.

[8] IEEE 802.11 Wireless Local Area Networks. ANSI/IEEE. [Online]. Available: http://grouper.ieee.org/groups/802/11/

[9] D. J. Baker and J. Wieselthier, “A distributed algorithm for scheduling the activation of links in a self-organizing, mobile, radio network,” in *Proc. Int. Conf. Commun.*, 1982, pp. 2F.6.1–2F.6.5.

[10] D. J. Baker, A. Ephremides, and J. Flynn, “The design and simulation of a mobile radio network with distributed control,” *IEEE Trans. Select. Areas Commun.*, vol. SAC-2, pp. 226–237, Jan. 1984.

[11] L. Bao and J. J. Garcia-Luna-Aceves, “A new approach to channel access scheduling for ad hoc networks,” in *Proc. ACM Int. Conf. Mobile Computing and Networking*, July 2001.

[12] R. Barr *et al.*, “On the need for system-level support for ad hoc and sensor networks,” *Oper. Syst. Rev.*, vol. 36, pp. 1–5, Apr. 2002.

[13] M. Barry, A. T. Campbell, and A. Veres, “Distributed control algorithms for service differentiation in wireless packet networks,” in *Proc. IEEE INFOCOM Conf.*, Apr. 2001, pp. 582–590.

[14] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, “A distance routing effect algorithm for mobility (DREAM),” in *Proc. ACM Int. Conf. Mobile Computing and Networking*, Oct. 1998, pp. 76–84.

[15] J. Behrens and J. J. Garcia-Luna-Aceves, “Distributed, scalable routing based on vectors of link states,” *IEEE J. Select. Areas Commun.*, vol. 13, pp. 1383–1395, Oct. 1995.

[16] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, “MACAW: A media access protocol for wireless LANs,” in *Proc. ACM SIGCOMM Conf.*, vol. 24, Aug. 1994, pp. 212–225.

[17] K. Birman, A. Schiper, and P. Stephenson, “Lightweight causal and atomic group multicast,” *ACM Trans. Comput. Syst.*, vol. 9, pp. 272–314, Aug. 1991.

[18] K. Birman, B. Constable, M. Hayden, J. Hickey, C. Kreitz, R. Van Renesse, O. Rodeh, and W. Vogels, “The horus and ensemble projects: Accomplishments and limitations,” presented at the DISCEX, Hilton Head, SC, Jan. 2000.

[19] A. D. Birrel and B. J. Nelson, “Implementing remote procedure calls,” *ACM Trans. Comput. Syst.*, vol. 2, pp. 39–59, Feb. 1984.

[20] P. Bonnet, J. Gehrke, T. Mayr, and P. Seshadri, “Query processing in a device database system,” Cornell Univ., Ithaca, NY, Tech. Rep. TR99-1775, Oct. 1999.

[21] P. Bonnet, J. Gehrke, and P. Seshardi, “Toward sensor database systems,” in *2nd Int. Conf. Mobile Data Management*, Hong Kong, Jan. 2001, pp. 3–14.

[22] R. Boppana and S. Konduru, “An adaptive distance vector routing algorithm for mobile, ad hoc networks,” in *Proc. IEEE INFOCOM Conf.*, Mar. 2001, pp. 1753–1762.

[23] J. Broch, D. A. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols,” in *Proc. ACM/IEEE Int. Conf. Mobile Computing and Networking*, Oct. 1998, pp. 85–97.

[24] M. Caccamo, L. Y. Zhang, L. Sha, and G. Buttazzo, “An implicit prioritized access protocol for wireless sensor networks,” in *Proc. IEEE Real-Time Systems Symp.*, Dec. 2002, pp. 39–48.

[25] J. B. Carter, J. K. Bennet, and W. Zwaenepoel, “Implementation and performance of munin,” in *Proc. ACM Symp. Operating Systems Principles*, Oct. 1991, pp. 152–164.

[26] A. Chandra, V. Gummalla, and J. O. Limb, “Wireless medium access control protocols,” *IEEE Commun. Surv.*, pp. 2–15. Second Quarter 2000.

[27] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, “A feedback based scheme for improving TCP performance in ad-hoc wireless networks,” in *Proc. Int. Conf. Distributed Computing Systems*, 1998, pp. 472–479.

[28] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, “Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks,” in *Proc. 7th ACM Int. Conf. Mobile Computing and Networking*, July 2001, pp. 85–96.

[29] T. W. Chen and M. Gerla, “Global state routing: A new routing scheme for ad-hoc wireless networks,” in *Proc. IEEE ICC Conf.*, June 1998, pp. 171–175.

[30] I. Chlamtac and A. Lerner, “A link allocation protocol for mobile multihop networks,” in *Proc. GLOBECOM Conf.*, Dec. 1985, pp. 238–242.

[31] I. Chlamtac and S. Kutten, "A spatial reuse TDMA/FDMA for mobile multihop radio networks," in *Proc. IEEE INFOCOM Conf.*, Mar. 1985, pp. 389–394.

[32] I. Chlamtac and A. Lerner, "Fair algorithms for maximal link activation in multihop radio networks," *IEEE Trans. Commun.*, vol. COM-35, pp. 739–746, July 1987.

[33] I. Chlamtac and S. Pinter, "Distributed node organization algorithm for channel access in a multihop dynamic radio network," *IEEE Trans. Comput.*, vol. C-36, pp. 728–737, June 1987.

[34] I. Chlamtac and A. Farago, "Making transmission schedules immune to topology changes in multi-hop packet radio networks," *IEEE/ACM Trans. Networking*, vol. 2, pp. 23–29, Feb. 1994.

[35] I. Cidon and M. Sidi, "Distributed assignment algorithms for multihop packet radio networks," *IEEE Trans. Comput.*, vol. 38, pp. 1353–1361, Oct. 1989.

[36] D. Coore, R. Nagapal, and R. Weiss, "Paradigms for structure in an amorphous computer," MIT AI Lab, Cambridge, MA, A.I. Memo 1614, Oct. 1997.

[37] D. Coore, "Botanical computing: A developmental approach to generating interconnect topologies on an amorphous computer," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, Dec. 1999.

[38] S. Corson *et al.*, "Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations," Network Working Group RFC 2501, 1999.

[39] S. Corson *et al.*, "Internet MANET encapsulation protocol (IMEP)," Internet Draft, IEFT MANET Working Group, 2001.

[40] D. Culler *et al.*. TinyOS: A component-based OS for the networked sensor regime. [Online]. Available: http://today.cs.berkeley.edu/tos/

[41] B. Das, R. Sivakumar, and V. Bharghavan, "The clade vertebrate: Spines and routing in ad-hoc networks," in *Proc. IEEE Symp. Computers and Communication*, 1998, pp. 599–605.

[42] P. Debaty and D. Caswell, "Uniform web presence architecture for people, places, and things," *IEEE Pers. Commun.*, vol. 8, pp. 46–51, Aug. 2001.

[43] J. Deng and Z. Haas, "Dual busy tone multiple access (DBTMA): A new medium access control for packet radio networks," in *Proc. IEEE ICUPC Conf.*, vol. 1, Oct. 1998, pp. 973–977.

[44] R. De Prisco, A. Fekete, N. Lynch, and A. Shvartsman, "A dynamic primary configuration group communication service," in *13th Int. Symp. Distributed Computing*, Bratislava, Slovak Republic, Sept. 1999, pp. 64–78.

[45] R. Dube, C. D. Rais, K. Wang, and S. K. Trpathi, "Signal stability based adaptive routing (SSA) for ad-hoc mobile networks," *IEEE Pers. Commun.*, vol. 4, pp. 36–45, Feb. 1997.

[46] D. Estrin, R. Govindan, S. Kumar, and J. Heeidemann, "Next century challenges: Scalable coordination in sensor networks," in *Proc. 5th Annu. IEEE/ACM Int. Conf. Mobile Computing and Networking*, Aug. 1999, pp. 263–270.

[47] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proc. IEEE INFOCOM Conf.*, Apr. 2001, pp. 1548–1557.

[48] C. L. Fullmer and J. J. Garcia-Luna-Aveces, "Floor acquisition multiple access for packet-radio networks," in *Proc. ACM SIGCOMM Conf.*, 1995, pp. 262–273.

[49] ——, "Solutions to hidden terminal problems in wireless networks," in *Proc. ACM SIGCOMM Conf.*, 1997, pp. 39–49.

[50] E. M. Gafni and D. P. Bertsekas, "Distributed algorithms for generating loop-free routes in networks with frequently changing topology," *IEEE Trans. Commun.*, vol. COM-29, pp. 11–18, Jan. 1981.

[51] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "An empirical study of epidemic algorithms in large scale multihop wireless networks," Dept. Comput. Sci., Univ. California, Los Angeles, Tech. Rep. UCLA/CSD-TR 02-0013.

[52] M. Gerla, K. Tang, and R. Bagrodia, "TCP performance in wireless multi-hop networks," in *Proc. IEEE WMCSA Conf.*, 1999, pp. 41–50.

[53] M. Gerla, R. Bagrodia, L. Zhang, K. Tang, and L. Wang, "TCP over wireless multihop protocols: Simulation and experiments," in *Proc. IEEE Int. Conf. Communications*, 1999, pp. 1089–1094.

[54] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurithi, "Packet reservation multiple access for local wireless communications," *IEEE Trans. Commun.*, vol. 37, pp. 885–890, June 1990.

[55] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inform. Theory*, vol. 46, pp. 388–404, Mar. 2000.

[56] Z. Haas, "A new routing protocol for the reconfigurable wireless networks," in *Proc. IEEE Int. Conf. Universal Personal Communications*, Oct. 1997, pp. 562–566.

[57] B. Hajek, "Balanced scheduling in a packet synchronized spread spectrum network," in *Proc. IEEE INFOCOM Conf.*, 1983, pp. 56–65.

[58] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE Trans. Inform. Theory*, vol. 34, pp. 910–918, Sept. 1988.

[59] C. C. Han, K. Lin, and C. Hou, "Distance-constrained scheduling and its applications to real-time systems," *IEEE/ACM Trans. Comput.*, vol. 45, pp. 814–826, July 1996.

[60] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A real-time routing protocol for sensor networks," in *Proc. ICDCS*, May 2003, pp. 46–55.

[61] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, "Building efficient wireless sensor networks with low-level naming," in *Proc. Symp. Operating Systems Principles*, Oct. 2001, pp. 146–159.

[62] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks," in *Proc. Hawaiian Int. Conf. Systems Science*, Jan. 2000, p. 8020.

[63] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *IEEE Comput.*, vol. 34, pp. 50–56, Aug. 2001.

[64] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for network sensors," in *Proc. Int. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Cambridge, MA, Nov. 2000.

[65] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, "System architecture directions for networked sensors," in *Proc. Int. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2000, pp. 93–104.

[66] G. Holland and N. H. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," in *Proc. 5th Annu. Int. Conf. Mobile Computing and Networking*, 1999, pp. 219–230.

[67] Y.-C. Hu and D. Johnson, "Caching strategies in on-demand routing protocols for wireless ad hoc networks," in *Proc. 6th ACM/IEEE Int. Conf. Mobile Computing and Networking*, Aug. 2000, pp. 231–242.

[68] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. 6th Annu. IEEE/ACM Int. Conf. Mobile Computing and Networking*, Aug. 2000, pp. 56–67.

[69] P. Johansson, T. Larsson, N. Hedman, and B. Mielczarek, "Routing protocols for mobile ad hoc networks—A comparative performance analysis," in *Proc. 5th ACM/IEEE Int. Conf. Mobile Computing and Networking*, Aug. 1999.

[70] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based performance analysis of routing protocols for mobile ad-hoc networks," in *Proc. 4th ACM/IEEE Int. Conf. Mobile Computing and Networking*, Aug., pp. 195–206.

[71] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad-hoc wireless networks," in *Mobile Computing*. Norwell, MA: Kluwer, 1996, pp. 153–181.

[72] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen, "A survey of energy efficient network protocols for wireless networks," *Wireless Networks*, vol. 7, pp. 343–358, Aug. 2001.

[73] J. H. Ju and V. O. K. Li, "An optimal topology-transparent scheduling method in multihop packet radio networks," *IEEE/ACM Trans. Networking*, vol. 6, pp. 298–306, June 1998.

[74] ——, "TDMA scheduling design of multihop packet radio networks based on Latin squares," *IEEE J. Select Areas Commun.*, vol. 17, pp. 1345–1352, Aug. 1999.

[75] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, "Distributed multi-hop scheduling with delay and throughput constraints," *Proc. ACM Int. Conf. Mobile Computing and Networking*, submitted for publication.

[76] P. Karn, "MACA—A new channel access method for packet radio," presented at the ARRL/CRRL Amateur Radio 9th Computer Networking Conf., 1990.

[77] B. Karp and H. T. Kung, "Greedy perimeter stateless routing for wireless networks," in *Proc. 6th Annu. ACM/IEEE Int. Conf. Mobile Computing and Networking*, Aug. 2000, pp. 243–254.

[78] Y. B. Ko and N. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," in *Proc. 4th ACM/IEEE Int. Conf. Mobile Computing and Networking*, Oct., pp. 66–75.

[79] Y. Ko and N. Vaidya, "Geocasting in mobile ad hoc networks: Location-based multicast algorithms," in *Proc. 2nd IEEE Workshop Mobile Computing Systems and Applications*, Feb. 1999, pp. 101–110.

[80] B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proc. Int. Workshop Distributed Event-Based Systems*, July 2002, pp. 575–578.

[81] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," in *Proc. 5th Annu. IEEE/ACM Int. Conf. Mobile Computing and Networking*, Aug. 1999.

[82] P. Levis and D. Culler, "Mate: A tiny virtual machine for sensor networks," in *Proc. Int. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, San Jose, CA, Oct. 2002.

[83] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proc. Int. Conf. Mobile Computing and Networking*, Boston, MA, Aug. 2000.

[84] L. Li, J. Y. Halpern, P. Bahl, Y. M. Wang, and R. Wattenhofer, "Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks," in *Proc. Annu. ACM Symp. Principles of Distributed Computing (PODC)*, Aug. 2001.

[85] V. O. K. Li, "Fair spatial TDMA channel access protocols for multihop radio networks," in *Proc. IEEE INFOCOM Conf.*, Apr. 1991, pp. 1064–1073.

[86] N. Li, J. C. Hou, and L. Sha, "Design and analysis of an MST-based topology control algorithm," presented at the Proc. 5th ACM Int. Workshop on Wireless Mobile Multimedia, Sept. 2002.

[87] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 1265–1275, Sept. 1997.

[88] ——, "Real-time support in multihop wireless networks," *ACM/Baltzer Wireless Networks*, vol. 5, no. 2, pp. 125–135, 1999.

[89] C. Lu, B. Blum, T. Abdelzaher, J. Stankovic, G. Tao, and S. Son, "RAP: A communication architecture for large-scale wireless sensor networks," in *Proc. IEEE Real-Time and Embedded Technology and Applications Symp.*, Sept. 2002, pp. 55–66.

[90] C. Lu, J. Stankovic, G. Tao, and S. Son, "The design and evaluation of a feedback control EDF scheduling algorithm," in *Proc. Real-Time Systems Symp.*, Dec. 1999, pp. 56–67.

[91] ——, "Feedback control real-time scheduling: Framework, modeling and algorithms," *RT Syst. J. (Special Issue on Control-Theoretic Approaches to Real-Time Computing)*, vol. 23, pp. 85–126, July/Sept. 2002.

[92] D. A. Maltz, J. Broch, J. Jetcheva, and D. Johnson, "The effects of on-demand behavior in routing protocols for multi-hop wireless ad hoc networks," *IEEE J. Select. Areas Commun. (Special Issue on Mobile and Wireless Networks)*, vol. 17, pp. 1439–1453, Aug. 1999.

[93] A. B. McDonald, "A mobility-based framework for adaptive dynamic cluster-based hybrid routing in wireless ad-hoc networks," Ph.D. dissertation proposal, Univ. Pittsburgh, Pittsburgh, PA, 1999.

[94] S. Meguerdichian, S. Slijepcevic, V. Karayan, and M. Potkonjak, "Localized algorithms in wireless ad-hoc networks: Location discovery and sensor exposure," in *Proc. Int. Conf. MobiHoc*, Long Beach, CA, Oct. 2001, pp. 106–116.

[95] J. Minglinag *et al.*, "Cluster based routing protocol,", Internet Draft, IETF MANET Working Group, 1998.

[96] J. P. Monks, P. Sinha, and V. Bharghavan, "Limitations of TCP-ELFN for ad hoc networks," in *Proc. 7th Int. Workshop Mobile Multimedia Communications*, Oct. 2000.

[97] J. P. Monks, V. Bharghavan, and W. Hwu, "A power controlled multiple access protocol for wireless packet networks," in *Proc. IEEE INFOCOM Conf.*, Apr. 2001, pp. 219–228.

[98] S. Murthy and J. J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *ACM Balzer Mobile Netw. Applicat. J. (Special Issue on Routing in Mobile Communications Networks)*, pp. 183–197, 1996.

[99] ——, "A path-finding algorithm for loop-free routing," *IEEE/ACM Trans. Networking*, vol. 5, pp. 148–160, Feb. 1997.

[100] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, "Achieving MAC layer fairness in wireless packet networks," in *Proc. ACM Int. Conf. Mobile Computing and Networking*, Aug. 2000, pp. 87–98.

[101] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, "Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol.," in *Proc. Eur. Wireless 2002. Next Generation Wireless Networks: Technologies, Protocols, Services and Applications*, Feb. 2002, pp. 156–162.

[102] R. Nelson and L. Kleinrock, "Spatial TDMA: A collision free multihop channel access protocol," *IEEE Trans. Commun.*, vol. COM-33, pp. 934–944, Sept. 1985.

[103] V. Park and S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proc. IEEE INFOCOM Conf.*, Apr. 1997, pp. 1405–1413.

[104] C. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers," in *Proc. ACM SIGCOMM Conf.*, 1994, pp. 212–225.

[105] C. E. Perkins and E. M. Royer, "Ad hoc on demand distance vector routing," IETF Internet Draft, draft-IETF-MANET-AODV-02.txt, Nov. 1998.

[106] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proc. 6th Int. Conf. ACM Mobile Computing and Networking*, Boston, MA, Aug. 2000, pp. 32–43.

[107] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proc. INFOCOM Conf.*, Mar. 2000, pp. 404–413.

[108] R. Ramanathan and M. Steenstrup, "Hierarchically-organized, multihop mobile networks for multimedia support," *ACM/Baltzer Mobile Networks Applicat.*, vol. 1, no. 2, pp. 89–103, 1998.

[109] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proc. ACM SIGCOMM Conf.*, San Diego, CA, Aug. 2001.

[110] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1333–1344, Aug. 1999.

[111] C. Rohl, H. Woesner, and A. Wolisz, "A short look on power saving mechanisms in the wireless LAN standard draft IEEE 802.11," in *Proc. 6th WINLAB Workshop Third Generation Wireless System*, Mar. 1997, pp. 183–188.

[112] E. M. Royer and C. E. Perkins, "Multicast ad hoc on-demand distance vector (MAODV) routing," in *Proc. 2nd IEEE Workshop Mobile Computing Systems and Applications*, 1999.

[113] R. Rozovsky and P. R. Kumar, "SEEDEX: A MAC protocol for ad hoc networks," in *Proc. Int. Conf. ACM MobiHoc*, Oct. 2001, pp. 67–75.

[114] P. Santi, D. M. Blough, and F. Vainstein, "A probabilistic analysis for the range assignment problem in ad hoc networks," in *Proc. Int. Conf. MobiHoc*, 2001, pp. 212–220.

[115] C. Shen, C. Srisathapornphat, and C. Jaikeo, "Sensor information networking architecture and applications," *IEEE Pers. Commun.*, vol. 8, pp. 52–59, Aug. 2001.

[116] T. J. Shepard, "A channel access scheme for large dense packet radio networks," in *Proc. ACM SIGCOMM Conf.*, 1996, pp. 219–230.

[117] S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proc. 4th Annu. IEEE/ACM Int. Conf. Mobile Computing and Networking*, Oct. 1998, pp. 181–190.

[118] E. Sirer, R. Grimm, A. Gregory, and B. Bershad, "Design and implementation of a distributed virtual machine for networked computers," in *Proc. ACM Symp. Operating System Principles*, Kiawah Island, SC, Dec. 1999, pp. 202–216.

[119] *Core Extraction Distributed Ad Hoc Routing (CEDAR) Specification*, Internet Draft, IETF MANET Working Group, Oct. 1998.

[120] J. Stankovic, T. He, T. Abdelzaher, M. Marley, G. Tao, and S. Son, "Feedback control scheduling in distributed systems," in *Proc. IEEE Real-Time Systems Symp.*, London, U.K., Dec. 2001, pp. 59–70.

[121] D. S. Stevens and M. H. Ammar, "Evaluation of slot allocation strategies for TDMA protocols in packet radio networks," in *Proc. IEEE MILCOM Conf.*, 1990, pp. 835–839.

[122] L. Subramanian and R. Katz, "An architecture for building self-configurable systems," presented at the IEEE/ACM Workshop Mobile Ad Hoc Networking and Computing, Aug. 2000.

[123] T. N. Tae-Eun, "Achieving MAC layer fairness in wireless packet networks," in *Proc. 6th Annu. Int. Conf. Mobile Computing and Networking*, 2000, pp. 87–98.

[124] F. Talucci and M. Gerla, "MACA-BI (MACA by invitation): A wireless MAC protocol for high speed ad hoc networking," in *Proc. IEEE ICUPC Conf.*, vol. 2, Oct. 1997, pp. 913–917.

[125] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part II—The hidden terminal problem in carrier sense multiple access and the busy tone solution," *IEEE Trans. Commun.*, vol. COS-23, pp. 1417–1433, Dec. 1975.

[126] C.-K. Toh, "Associativity-based routing for ad-hoc networks," *Wireless Pers. Commun.*, vol. 4, pp. 103–139, Mar. 1997.

[127] T. Truong, "TDMA in mobile radio networks: Graph partitioning and time slot assignment," in *Proc. IEEE MILCOM Conf.*, Oct. 1985.

[128] J. Tsai and M. Gerla, "Multicluster mobile multimedia radio network," *Wireless Networks*, vol. 1, pp. 255–265, Oct. 1995.

[129] N. H. Vaidya *et al.*, "A cluster-based approach for routing in dynamic networks," in *ACM SIGCOMM Computer Communication Rev.*, Apr. 1997, pp. 49–65.

[130] N. H. Vaidya, P. Bahl, and S. Gupta, “Distributed fair scheduling in a wireless LAN,” in *Proc. ACM Mobile Computing and Networking*, Aug. 2000, pp. 167–178.

[131] S. Vinoski, “CORBA: Integrating diverse applications within distributed heterogeneous environments,” *IEEE Commun. Mag.*, vol. 14, pp. 46–55, Feb. 1997.

[132] A. Woo and D. E. Culler, “A transmission control scheme for media access in sensor networks,” in *Proc. ACM Int. Conf. Mobile Computing and Networking*, 2001.

[133] A. Wood and J. Stankovic, “Denial of service in sensor networks,” *IEEE Computer*, vol. 35, pp. 54–62, Oct. 2002.

[134] C. Wu and V. O. K. Li, “Receiver-initiated busy tone multiple access in packet radio networks,” in *Proc. ACM SIGCOMM Conf.*, 1988, pp. 336–342.

[135] Y. Xu, J. Heidemann, and D. Estrin, “Geography-informed energy conservation for ad hoc routing,” in *Proc. 7th ACM Int. Conf. Mobile Computing and Networking*, July 2001, pp. 70–84.

[136] Y. Xue and B. Li, “A location-aided power-aware routing protocol in mobile ad hoc networks,” in *Proc. IEEE Globecom*, vol. 5, Nov. 2001, pp. 2837–2841.

[137] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” in *Proc. 21st Int. Annu. Joint Conf. IEEE Computer and Communications Societies*, June 2002, pp. 1567–1576.

[138] R. Zheng and C.-J. Hou, “Toward alleviating the effect of path asymmetry on the TCP performance in ad hoc networks,” submitted for publication.

John A. Stankovic (Fellow, IEEE) received the Ph.D. degree from Brown University, Providence, RI.

He is the BP America Professor and Chair of the Computer Science Department at the University of Virginia, Charlottesville. Before joining the University of Virginia, he taught at the University of Massachusetts, where he won an Outstanding Scholar Award. He has also held visiting positions in the Computer Science Department at Carnegie Mellon University, Pittsburgh, PA, at INRIA, France, and Scuola Superiore S. Anna, Pisa, Italy. His research interests are in distributed computing, real-time systems, operating systems, and ad hoc networks for pervasive computing.

Dr. Stankovic is a Fellow of the Association for Computing Machinery (ACM). He serves as Treasurer for the Board of Directors of the Computer Research Association. He received the IEEE Real-Time Systems Technical Committee’s Award for Outstanding Technical Contributions and Leadership. He was the Editor-in-Chief for the IEEE TRANSACTIONS ON DISTRIBUTED AND PARALLEL SYSTEMS and is a Co-Editor-in-Chief of the *Real-Time Systems Journal*.

Tarek F. Abdelzaher (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical and computer engineering from Ain Shams University, Cairo, Egypt, in 1990 and 1994, respectively, and the Ph.D. degree in computer science with a specialization in real-time systems from the University of Michigan, Ann Arbor, in 1999.

He is an Assistant Professor in the Computer Science Department at the University of Virginia, Charlottesville. He is author and co-author of more than 45 refereed publications. His current research interests include real-time systems, sensor networks, QoS control, networking, multimedia applications, next generation Web architecture, fault-tolerance, and dependable computing. He has served on many conference committees and is the designated inventor of a European Patent on Adaptive Web Servers.

Dr. Abdelzaher received the Distinguished Achievement Award in Computer Science and Engineering from the University of Michigan in 1999, and the NSF CAREER Award in 2001. He is a Co-Editor of IEEE DISTRIBUTED SYSTEMS ONLINE and has served as a Guest Editor of *Computer Communication* and the *Journal of Real-Time Systems*.

Chenyang Lu received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 1995, the M.S. degree from the Chinese Academy of Sciences, Beijing, China, in 1997, and the Ph.D. degree from the University of Virginia, Charlottesville, in 2001, all in computer science.

He is an Assistant Professor in the Department of Computer Science and Engineering at Washington University, St Louis, MO. His current research interests include real-time systems and middleware, wireless sensor networks, Internet servers, and network storage. A theme of his research is developing adaptive distributed real-time middleware and protocols that provide performance-assured services in unpredictable environments.

Lui Sha (Fellow, IEEE) received the Ph.D. degree from Carnegie Mellon University, Pittsburgh, PA, in 1985.

Currently, he is a Professor of Computer Science at the University of Illinois, Urbana Champaign. He is interested in distributed real-time systems.

Dr. Sha was the Chair of IEEE Real Time Systems Technical Committee from 1999 to 2000. He is currently an Associate Editor of the *Journal of Real Time Systems*.

Jennifer C. Hou (Senior Member, IEEE) received the B.S. degree in electrical engineering from National Taiwan University, Taiwan, R.O.C., in 1987, the M.S.E degrees in EECS and in I&OE in 1989 and in 1991, respectively, and the Ph.D. degree in EECS in 1993, all from the University of Michigan, Ann Arbor.

She was an Assistant Professor of Electrical and Computer Engineering at the University of Wisconsin, Madison, from 1993 to 1996, and an Assistant/Associate Professor of Electrical Engineering at Ohio State University, Columbus, from 1996 to 2001. Since August 2001, she has been with the Department of Computer Science at the University of Illinois, Urbana, where she is currently an Associate Professor. She has served on the TPC of several major networking, real-time, and distributed systems conferences/symposiums. Her recent research interests are in the areas of wireless sensor networks, network modeling, simulation, and emulation.

Dr. Hou is a member of the Association for Computing Machinery (ACM). She was a recipient of the Lumley Research Award from Ohio State University in 2001, the NSF CAREER award in 1996, and the Women in Science Initiative Award from the University of Wisconsin, Madison, in 1993.