Wireless Sensor Networks

John A. Stankovic

Department of Computer Science

University of Virginia

Charlottesville, Virginia 22904

E-mail: stankovic@cs.virginia.edu

June 19, 2006

1 Introduction

A wireless sensor network is a collection of nodes organized into a cooperative network [10]. Each node consists of processing capability (one or more microcontrollers, CPUs or DSP chips), may contain multiple types of memory (program, data and flash memories), have a RF transceiver (usually with a single omni-directional antenna), have a power source (e.g., batteries and solar cells), and accommodate various sensors and actuators. The nodes communicate wirelessly and often self-organize after being deployed in an ad hoc fashion. Systems of 1000s or even 10,000 nodes are anticipated. Such systems can revolutionize the way we live and work.

Currently, wireless sensor networks are beginning to be deployed at an accelerated pace. It is not unreasonable to expect that in 10-15 years that the world will be covered with wireless sensor networks with access to them via the Internet. This can be considered as the Internet becoming a physical network. This new technology is exciting with unlimited potential for numerous application areas including environmental, medical, military, transportation, entertainment, crisis management, homeland defense, and smart spaces.

Since a wireless sensor network is a distributed real-time system a natural question is how many solutions from distributed and real-time systems can be used in these new systems? Unfortunately, very little prior

work can be applied and new solutions are necessary in all areas of the system. The main reason is that the set of assumptions underlying previous work has changed dramatically. Most past distributed systems research has assumed that the systems are wired, have unlimited power, are not real-time, have user interfaces such as screens and mice, have a fixed set of resources, treat each node in the system as very important and are location independent. In contrast, for wireless sensor networks, the systems are wireless, have scarce power, are real-time, utilize sensors and actuators as interfaces, have dynamically changing sets of resources, aggregate behavior is important and location is critical. Many wireless sensor networks also utilize minimal capacity devices which places a further strain on the ability to use past solutions.

This Chapter presents an overview of some of the key areas and research in wireless sensor networks. In presenting this work, we use examples of recent work to portray the state of art and show how these solutions differ from solutions found in other distributed systems. In particular, we discuss the MAC layer (section 2), routing (section 3), node localization (section 4), clock synchronization (section 5), and power management (section 6). We also present a brief discussion of two current systems (section 7) in order to convey overall capabilities of this technology. We conclude in section 8.

2 MAC

A medium access control (MAC) protocol coordinates actions over a shared channel. The most commonly used solutions are contention-based. One general contention-based strategy is for a node which has a message to transmit to test the channel to see if it is busy, if not busy then it transmits, else if busy it waits and tries again later. After colliding, nodes wait random amounts of time trying to avoid re-colliding. If two or more nodes transmit at the same time there is a collision and all the nodes colliding try again later. Many wireless MAC protocols also have a doze mode where nodes not involved with sending or receiving a packet in a given timeframe go into sleep mode to save energy. Many variations exist on this basic scheme.

In general, most MAC protocols optimize for the general case and for arbitrary communication patterns and workloads. However, a wireless sensor network has more focused requirements that include a local uni- or broad-cast, traffic is generally from nodes to one or a few sinks (most traffic is then in one direction), have periodic or rare communication and must consider energy consumption as a major factor. An effective

MAC protocol for wireless sensor networks must consume little power, avoid collisions, be implemented with a small code size and memory requirements, be efficient for a single application, and be tolerant to changing radio frequency and networking conditions.

One example of a good MAC protocol for wireless sensor networks is B-MAC [24]. B-MAC is highly configurable and can be implemented with a small code and memory size. It has an interface that allows choosing various functionality and only that functionality as needed by a particular application. B-MAC consists of four main parts: clear channel assessment (CCA), packet backoff, link layer acks, and low power listening. For CCA, B-MAC uses a weighted moving average of samples when the channel is idle in order to assess the background noise and better be able to detect valid packets and collisions. The packet backoff time is configurable and is chosen from a linear range as opposed to an exponential backoff scheme typically used in other distributed systems. This reduces delay and works because of the typical communication patterns found in a wireless sensor network. B-MAC also supports a packet by packet link layer acknowledgement. In this way only important packets need pay the extra cost. A low power listening scheme is employed where a node cycles between awake and sleep cycles. While awake it listens for a long enough preamble to assess if it needs to stay awake or can return to sleep mode. This scheme saves significant amounts of energy. Many MAC protocols use a request to send (RTS) and clear to send (CTS) style of interaction. This works well for ad hoc mesh networks where packet sizes are large (1000s of bytes). However, the overhead of RTS-CTS packets to set up a packet transmission is not acceptable in wireless sensor networks where packet sizes are on the order of 50 bytes. B-MAC, therefore, does not use a RTS-CTS scheme.

Recently, there has been new work on supporting multi-channel wireless sensor networks. In these systems it is necessary to extend MAC protocols to multi-channel MACs. One such protocol is MMSN [36]. These protocols must support all the features found in protocols such as B-MAC, but must also assign frequencies for each transmission. Consequently, multi-frequency MAC protocols consist of two phases: channel assignment and access control. The details for MMSN are quite complicated and are not described here. On the other hand, we expect that more and more future wireless sensor networks will employ multiple channels (frequencies). The advantages of multi-channel MAC protocols include providing greater packet throughput and being able to transmit even in the presence of a crowded spectrum, perhaps arising from

competing networks or commercial devices such as phones or microwave ovens.

3 Routing

Multihop routing is a critical service required for WSN. Because of this, there has been a large amount of work on this topic. Internet and MANET routing techniques do not perform well in WSN. Internet routing assumes highly reliable wired connections so packet errors are rare; this is not true in WSN. Many MANET routing solutions depend on symmetric links (i.e., if node A can reliably reach node B, then B can reach A) between neighbors; this is too often not true for WSN. These differences have necessitated the invention and deployment of new solutions.

For WSN, which are often deployed in an ad hoc fashion, routing typically begins with neighbor discovery. Nodes send rounds of messages (packets) and build local neighbor tables. These tables include the minimum information of each neighbor's ID and location. This means that nodes must know their geographic location prior to neighbor discovery. Other typical information in these tables include nodes' remaining energy, delay via that node, and an estimate of link quality.

Once the tables exist, in most WSN routing algorithms messages are directed from a source location to a destination address based on geographic coordinates, not IDs. A typical routing algorithm that works like this is Geographic Forwarding (GF) [12].

In GF, a node is aware of its location, and a message that it is "routing" contains the destination address. This node can then computes which neighbor node makes the most progress towards the destination by using the distance formula from geometry. It then forwards the message to this next hop. In variants of GF, a node could also take into account delays, reliability of the link and remaining energy.

Another important routing paradigm for WSN is directed diffusion [11]. This solution integrates routing, queries and data aggregation. Here a query is disseminated indicating an interest in data from remote nodes. A node with the appropriate requested data responds with an attribute-value pair. This attribute-value pair is drawn towards the requestor based on gradients, which are set up and updated during query dissemination and response. Along the path from the source to the destination, data can be aggregated to reduce communication costs. Data may also travel over multiple paths increasing the robustness of routing.

Beyond the basics of WSN routing just presented, there are many additional key issues including:

- Reliability,
- Integrating with wake/sleep schedules,
- Unicast, multicast and anycast semantics,
- Real-time,
- Mobility,
- Voids,
- Security, and
- Congestion.

Reliability: Since messages travel multiple hops it is important to have a high reliability on each link, otherwise the probability of a message transiting the entire network would be unacceptably low. Significant work is being done to identify reliable links using metrics such as received signal strength, link quality index which is based on "errors," and packet delivery ratio. Significant empirical evidence indicates that packet delivery ratio is the best metric, but it can be expensive to collect. Empirical data also shows that many links in a WSN are asymmetric, meaning that while node A can successfully transmit a message to node B, the reverse link from B to A may not be reliable. Asymmetric links are one reason MANET routing algorithms such as DSR and AODV do not work well in WSN because those protocols send a discovery message from source to destination and then use the reverse path for acknowledgements. This reverse path is not likely to be reliable due to the high occurrence of asymmetry found in WSN.

Integration with wake/sleep schedules: To save power many WSN place nodes into sleep states. Obviously, an awake node should not choose an asleep node as the next hop (unless it first awakens that node).

Unicast, multicast and anycast semantics: As mentioned above, in most cases a WSN routes messages to a geographic destination. What happens when it arrives at this destination? There are several possibilities. First, the message may also include an ID with a specific unicast node in this area as the target, or the

semantics may be that a single node closest to the geographic destination is to be the unicast node. Second, the semantics could be that all nodes within some area around the destination address should receive the message. This is an area multicast. Third, it may only be necessary for any node, called anycast, in the destination area to receive the message. The SPEED [5] protocol supports these 3 types of semantics. There is also often a need to flood (multicast) to the entire network. Many routing schemes exist for supporting efficient flooding.

Real-Time: For some applications, messages must arrive at a destination by a deadline. Due to the high degree of uncertainty in WSN it is difficult to develop routing algorithms with any guarantees. Protocols such as SPEED [5] and RAP [16] use a notion of velocity to prioritize packet transmissions. Velocity is a nice metric that combines the deadline and distance that a message must travel.

Mobility: Routing is complicated if either the message source or destination or both are moving. Solutions include continuously updating local neighbor tables or identifying proxy nodes which are responsible for keeping track of where nodes are. Proxy nodes for a given node may also change as a node moves further and further away from its original location.

Voids: Since WSN nodes have a limited transmission range, it is possible that for some node in the routing path there are no forwarding nodes in the direction a message is supposed to travel. Protocols like GPSR [13] solve this problem by choosing some other node "not" in the correct direction in an effort to find a path around the void.

Security: If adversaries exist, they can perpetrate a wide variety of attacks on the routing algorithm including selective forwarding, black hole, Sybil, replays, wormhole and denial of service attacks. Unfortunately, almost all WSN routing algorithms have ignored security and are vulnerable to these attacks. Protocols such as SPINS [23] have begun to address secure routing issues.

Congestion: Today, many WSN have periodic or infrequent traffic. Congestion does not seem to be a big problem for such networks. However, congestion is a problem for more demanding WSN and is expected to be a more prominent issue with larger systems that might process audio, video and have multiple base stations (creating more cross traffic). Even in systems with a single base station, congestion near the base station is a serious problem since traffic converges at the base station. Solutions use backpressure, reducing

source node transmission rates, throwing out less important messages, and using scheduling to avoid as many collisions as possible which only exacerbate the congestion problem.

4 Node Localization

Node localization is the problem of determining the geographical location of each node in the system. Localization is one of the most fundamental and difficult problems that must be solved for WSN. Localization is a function of many parameters and requirements potentially making it very complex. For example, issues to consider include: the cost of extra localization hardware, do beacons (nodes which know their locations) exist and if so, how many and what are their communication ranges, what degree of location accuracy is required, is the system indoors/outdoors, is there line of sight among the nodes, is it a 2D or 3D localization problem, what is the energy budget (number of messages), how long should it take to localize, are clocks synchronized, does the system reside in hostile or friendly territory, what error assumptions are being made, and is the system subject to security attacks?

For some combination of requirements and issues the problem is easily solved. If cost and form factor are not major concerns and accuracy of a few meters is acceptable, then for outdoor systems, equipping each node with GPS is a simple answer. If the system is manually deployed one node at a time, then a simple GPS node carried with the deployer can localize each node, in turn, via a solution called Walking GPS [27]. While simple, this solution is elegant and avoids any manual keying in the location for each node.

Most other solutions for localization in WSN are either range-based or range- free. Range-based schemes use various techniques to first determine distances between node (range) and then compute location using geometric principles. To determine distances, extra hardware is usually employed, e.g., hardware to detect the time difference of arrival of sound and radio waves. This difference can then be converted to a distance measurement. In range-free schemes distances are not determined directly, but hop counts are used. Once hop counts are determined, distances between nodes are estimated using an average distance per hop, and then geometric principles are used to compute location. Range-free solutions are not as accurate as range-based solutions and often require more messages. However, they do not require extra hardware on every node.

Several early localization solutions include centroid [1] and APIT [6]. Each of these protocols solves the localization problem for a particular set of assumptions. Two recent and interesting solutions are Spotlight [26] and Radio Interferometric Geolocation [20]. Spotlight removes most of the localization code and overhead to a centralized laser device. Spotlight requires line of sight and clock synchronization. Radio interferometric geolocation uses a novel in-network processing technique that relies on nodes emitting radio waves simultaneously at slightly different frequencies. This solution is subject to multi-path problems in some deployments and can require many messages. Both of these recent solutions provide a high accuracy in the cm range.

5 Clock Synchronization

The clocks of each node in a WSN should read the same time within epsilon and remain that way. Since clocks drift over time, they must be periodically re- synchronized and in some instances when very high accuracy is required it is even important for nodes to account for clock drift between synchronization periods.

Clock synchronization is important for many reasons. When an event occurs in a WSN it is often necessary to know where and when it occurred. Clocks are also used for many system and application tasks. For example, sleep/wake-up schedules, some localization algorithms, and sensor fusion are some of the services that often depend on clocks being synchronized. Application tasks such as tracking and computing velocity are also dependent on synchronized clocks.

The NTP protocol [21] used to synchronize clocks or the Internet is too heavyweight for WSN. Placing GPS on every node is too costly. Representative clock synchronization protocols that have been developed for WSN are: RBS [3], TPSN [4] and FTSP [19].

In RBS a reference time message is broadcast to neighbors. Receivers record the time when the message is received. Nodes exchange their recorded times and adjust their clocks to synchronize. This protocol suffers no transmitter side non-determinism since timestamps are only on the receiver side. Accuracies are around 30 microseconds for 1 hop. This work did not address multi-hop systems, but could be extended.

In TPSN a spanning tree is created for the entire network. This solution assumes that all links in the spanning tree are symmetric. Then pairwise synchronization is performed along the edges of the tree

starting at the root. Since there is no broadcasting as in RBS, TPSN is expensive. A key attribute of this protocol is that the timestamps are inserted into outgoing messages in the MAC layer thereby reducing non-determinism. Accuracy is in the range of 17 microseconds.

In FTSP, there are radio-layer timestamps, skew compensation with linear regression, and periodic flooding to make the protocol robust to failures and topology changes. Both transmission and reception of messages are timestamped in the radio layer and differences are used to compute and adjust clock offsets. Accuracy is in the range of 1-2 microseconds.

Considerations in using a clock synchronization protocol include choosing the frequency of resynchronization, determination if clock drift between synchronization times is required, how to handle the multi-hop/network problem, and minimizing overhead costs in terms of energy and added network congestion.

6 Power Management

Many devices such as Mica2 and MicaZ that are used in WSN run on two AA batteries. Depending on the activity level of a node, its lifetime may only be a few days if no power management schemes are used. Since most systems require much longer lifetime, significant research has been undertaken to increase lifetime while still meeting functional requirements.

At the hardware level it is possible to add solar cells or scavenge energy from motion or wind. Batteries are also improving. If form factor is not a problem then it is also possible to add even more batteries. Low power circuits and microcontrollers are improving. Most hardware platforms allow multiple power saving states (off, idle, on) for each component of the device (each sensor, the radio, the microcontroller). In this way, only the components required at a particular time need to be active.

At the software level power management solutions are targeted at (i) minimizing communications since transmitting and listening for messages is energy expensive, and (ii) creating sleep/wake-up schedules for nodes or particular components of nodes.

Minimizing the number of messages is a cross-cutting problem. For example, with a good MAC protocol there are fewer collisions and retries. With good routing, short paths and congestion avoidance or minimization can be achieved and this minimizes the number of messages sent. Efficient neighbor discovery,

time synchronization, localization, query dissemination and flooding can all reduce the number of messages thereby increasing lifetime.

Solutions to schedule sleep/wake-up patterns vary considerably. Many solutions attempt to keep awake the minimum number of nodes, called sentries, to provide the required sensing coverage while permitting all the others to sleep. To balance energy consumption a rotation is performed periodically where new sentries are selected for the next period of time. Another common technique is to duty-cycle nodes. As an example, a node may be awake for 200 milliseconds out of each second for a 20% duty cycle. The duty cycle percentage chosen depends on application requirements, but the end result is usually a very significant savings in energy. Note that duty cycle and sentry solutions can be combined as was done in the VigilNet military surveillance system [7, 9].

7 Applications and Systems

To demonstrate the capabilities of wireless sensor networks we present two examples of applications and associated systems for those applications.

7.1 Surveillance and Tracking

The VigilNet system is a long-lived real-time wireless sensor network for military surveillance. The general objective of VigilNet is to alert military command and control units of the occurrence of events of interest in hostile regions. The events of interest are the presence of people, people with weapons, and large and small vehicles. Successful detection, tracking and classification require that the application obtain the current position of an object with acceptable precision and confidence. When the information is obtained, it is reported to a remote base station within an acceptable latency. VigilNet is an operational self-organizing sensor network (of over 200 XSM mote nodes) to provide tripwire-based surveillance with a sentry-based power management scheme, in order to achieve minimum 3 to 6 months lifetime. The tripwire also activates additional external (i.e., out of the Vigilnet system proper) sensors, e.g., infrared cameras, only when necessary, thereby also increasing their lifetimes as well.

Figure 1.1 provides an overview of the VigilNet architecture, in which there are three categories of

components: 1) Application components, 2) Middleware components, and 3) TinyOS system components. The application components are specially designed for surveillance purposes. It includes 1) an entity-based tracking service, 2) classification components, which provide four types of target differentiation, 3) velocity calculation, which provides target speed and bearing estimation, and 4) false alarm filtering, which differentiates between real and false targets.

Middleware components are designed to be application independent. Time synchronization, localization, and routing comprise the lower-level components and form the basis for implementing the higher-level middleware services, such as aggregation and power management. Time synchronization and localization are important for a surveillance application because the collaborative detection and tracking process relies on the spatiotemporal correlation between the tracking reports sent by multiple motes. The time synchronization module is responsible for synchronizing the local clocks of the motes with the clock of the base station. The localization module is responsible for ensuring that each mote is aware of its location. The configuration module is responsible for dynamically reconfiguring the system when system requirements change. Asymmetric detection is designed to aid the routing module to select high-quality communication links. The radio wakeup module is used to alert non-sentry motes when significant events happen. Power management and collaborative detection are two key higher-level services provided by VigilNet. The sentry service and tripwire management are responsible for power management, while the group management component is responsible for collaborative detection and tracking of events. The sentry and tripwire services conserve energy of the sensor network by selecting a subset of motes, which are defined as sentries, to monitor events. The remaining motes are allowed to remain in a low-power state until an event occurs. When an event occurs, the sentries awaken the other motes in the region of the event and the group management component dynamically organizes the motes into groups in order to collaboratively track. Together, these two components are responsible for energy-efficient event tracking.

The VigilNet architecture was built on top of TinyOS. TinyOS is an event driven computation model, written in NesC specifically for the motes platform. TinyOS provides a set of essential components such as hardware drivers, a scheduler and basic communication protocols. These components provide low-level support for VigilNet modules, which are also written in NesC. Components from TinyOS and VigilNet

applications are processed by the NesC compiler into a running executable, which runs (in the VigilNet case) on the XSM (and MICA2) mote platforms.

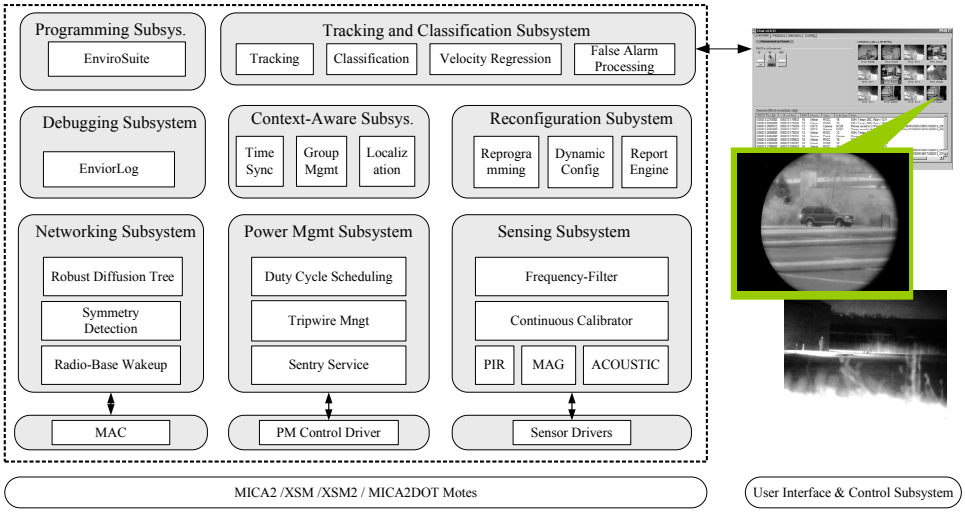


Figure 1.1: VigilNet Architecture

7.2 Assisted Living Facilities

AlarmNet [32, 28], a medical-oriented sensor network system for large-scale assisted living facilities, integrates heterogeneous devices, some wearable on the patient and some placed inside the living space. Together they inform the healthcare provider about the health status of the resident. Data is collected, aggregated, pre-processed, stored, and acted upon using a variety of replaceable sensors and devices (activity sensors, physiological sensors, environmental sensors, pressure sensors, RFID tags, pollution sensors, floor sensors, etc.). Multiple body networks are present in the system. Traditional healthcare provider networks may connect to the system by a residential gateway, or directly to their distributed databases. Some elements of the network are mobile such as the body networks as well as some of the infrastructure network nodes, while others are stationary. Some nodes can use line power, but others depend on batteries. The system is designed to exist across a large number of living units. The system architecture for AlarmNet is shown in Figure 1.2. Each tier of the architecture is briefly described below.

- **Body Networks and Front-ends**. The body network is composed of tiny portable devices equipped with a variety of sensors (such as heart-rate, heart-rhythm, temperature, pulse oximeter, accelerome-

ter), and performs biophysical monitoring, patient identification, location detection, and other desired tasks. Their energy consumption is also optimized so that the battery is not required to be changed regularly. They may use kinetic recharging. Actuators notify the wearer of important messages from an external entity. For example, an actuator can remind an early Alzheimer patient to check the oven because sensors detect an abnormally high temperature. Or, a tone may indicate that it is time to take medication. A node in the body network is designated as the gateway to the emplaced sensor network. Due to size and energy constraints, nodes in this network have little processing and storage capabilities.

- **Emplaced Sensor Network**. This network includes sensor devices deployed in the assisted living environment (rooms, hallways, units, furniture) to support sensing and monitoring, including: motion, video cameras, temperature, humidity, acoustic, smoke, dust, pollen, and gas. All devices are connected to a more resourceful backbone. Sensors communicate wirelessly using multi-hop routing and may use either wired or battery power. Nodes in this network may be physically moved and may vary in their capabilities, but generally do not perform extensive calculation or store much data.

- **Backbone**. A backbone network connects traditional systems, such as PDAs, PCs, and in-network databases, to the emplaced sensor network. It also connects sensor nodes by a high-speed relay for efficient routing. The backbone may communicate wirelessly or may overlay onto an existing wired infrastructure. Some nodes possess significant storage and computation capability, for query processing and location services. Yet, their number, depending on the topology of the building, is minimized to reduce cost.

- **In-network and Back-end Databases**. One or more nodes connected to the backbone are dedicated in-network databases for real-time processing and temporary caching. If necessary, nodes on the backbone may serve as in-network databases themselves. Back-end databases are located at the medical center for long-term archiving, monitoring and data mining for longitudinal studies.

- **Human Interfaces**. Patients and caregivers interface with the network using PDAs, PCs, or wearable devices. These are used for data management, querying, object location, memory aids, and config-

uration, depending on who is accessing the system and for what purpose. Limited interactions are supported with the on-body sensors and control aids. These may provide memory aids, alerts, and an emergency communication channel. PDAs and PCs provide richer interfaces to real-time and historical data. Caregivers use these to specify medical sensing tasks and to view important data.

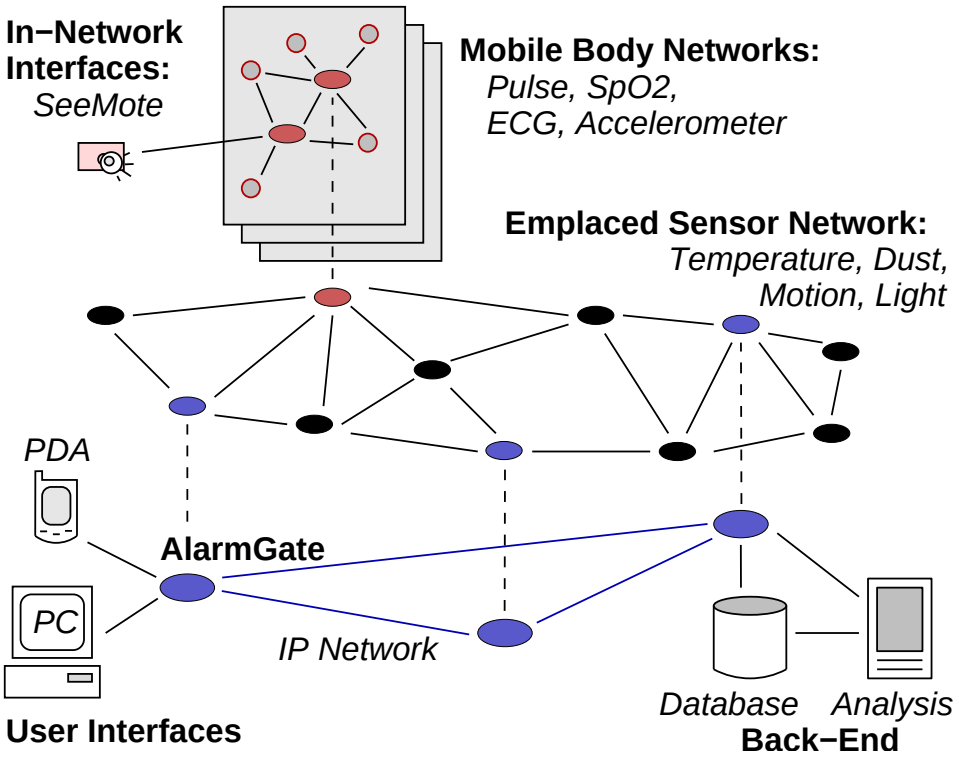
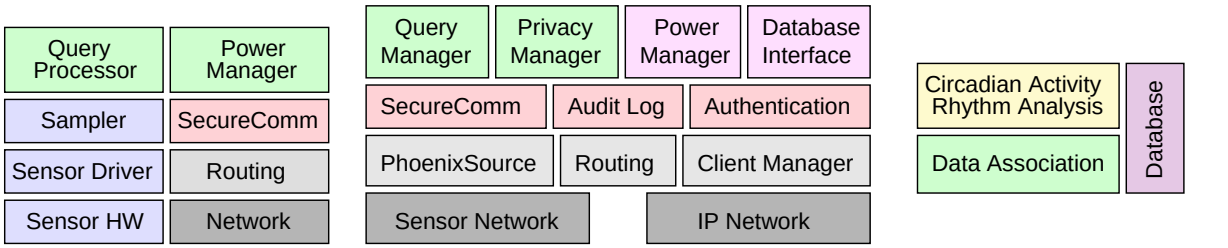


Figure 1.2: AlarmNet System Architecture

The software components of the AlarmNet architecture are shown in Figure 1.3. Sensor devices require components for sensing, networking, power management, handling queries and supporting security. The Stargates implement significantly more functionality including increased security, privacy, query management and database support. In the back-end many functions are performed including circadian rhythm analysis measuring residents' behaviors on a 24 hour basis, data association, security and privacy, and multiple database functions.



(a) Sensor device software stack. (b) AlarmGate software stack. (c) Back-end analysis and storage.

Figure 1.3: AlarmNet component software.

8 Conclusions

This Chapter has discussed WSN issues and example solutions for the MAC layer, routing, localization, clock synchronization, and power management. Why these solutions are different from past networking solutions was stressed. The Chapter also provided a short description of two representative WSN systems: a military surveillance, tracking and classification system and an assisted living facility system.

While these topics are some of the key issues regarding WSN, there are many important topics not discussed in this Chapter. For example, security and privacy are critical services needed for these systems [22, 23, 31]. Programming abstractions and languages for WSN are very active areas of research [17, 15, 14, 29]. Significant and important studies have been collecting empirical data on the performance of WSN [34, 35, 30, 2]. Such data is critical to developing improved models and solutions. Tools for debugging and management of WSN are appearing [18, 25]. Several other areas of research are just beginning, but are critical to the eventual success of WSN. These include in-field auto-calibration and re-calibration of sensors, low cost signal processing techniques that can be run on microcontrollers with minimum power and memory, and low cost AI learning schemes to support self-organization, parameter tuning and calibration.

All of this sensor network research is producing a new technology which is already appearing in many practical applications. The future should see an accelerated pace of adoption of this technology.

References

[1] N. Bulusu, J. Heidemann, and D. Estrin, GPS-less Low Cost Outdoor Localization for Very Small Devices, *IEEE Personal Communications Magazine*, October 2000.

[2] A. Cerpa, J. Wong, L. Kuang, M. Potkonjak, and D. Estrin, Statistical Model of Lossy Links in Wireless Sensor Networks, *IPSN*, April 2005.

[3] J. Elson, L. Girod, and D. Estrin, Fine-Grained Network Time Synchronization Using Reference Broadcasts, *OSDI*, December 2002.

[4] S. Ganeriwal, R. Kumar, and M. Srivastava, Timing-sync Protocol for Sensor Networks, *ACM SenSys*, November 2003.

[5] T. He, J. Stankovic, C. Lu and T. Abdelzaher, A Spatiotemporal Communication Protocol for Wireless Sensor Networks, *IEEE Transactions on Parallel and Distributed Systems*, to appear.

[6] T. He, C. Huang, B. Blum, J. Stankovic, T. Abdelzaher, Range-Free Localization and Its Impact on Large Scale Sensor Networks, *ACM Transactions on Embedded Computing System*, to appear.

[7] T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, T. Yan, R. Stoleru, L. Gu, G. Zhou, J. Hui and B. Krogh, VigilNet: An Integrated Sensor Network System for Energy Efficient Surveillance, *ACM Transactions on Sensor Networks*, to appear.

[8] T. He, P. Vicaire, T. Yan, L. Luo, L. Gu, G. Zhou, R. Stoleru, Q. Cao, J. Stankovic, and T. Abdelzaher, Real-Time Analysis of Tracking Performance in Wireless Sensor Networks, *IEEE Real-Time Applications Symposium*, May 2006.

[9] T. He, P. Vicaire, T. Yan, Q. Cao, L. Luo, L. Gu, G. Zhou, J. Stankovic, and T. Abdelzaher, Achieving Long Term Surveillance in VigilNet, *Infocom*, April 2006.

[10] J. Hill, R. Szewczyk, A, Woo, S. Hollar, D. Culler, and K. Pister, System Architecture Directions for Networked Sensors, *ASPLOS*, November 2000.

[11] C. Intanagonwiwat, R. Govindan, and D. Estrin, Directed Diffusion: A Scalable Routing and Robust Communication Paradigm for Sensor Networks, *Mobicom*, August 2000.

[12] B. Karp, Geographic Routing for Wireless Networks, PhD Dissertation, Harvard University, October 2000.

[13] B. Karp and H. T. Kung, GPSR: Greedy Perimeter Stateless Routing for Wireless Sensor Networks, *IEEE Mobicom*, August 2000.

[14] P. Levis and D. Culler, Mate: A Tiny Virtual Machine for Sensor Networks, *Int. Conf. on Architectural Support for Programming Languages and Operating Systems*, October 2002.

[15] J. Liu, M. Chu, J.J. Liu, J. Reich and F. Zhao, State-centric Programming for Sensor and Actuator Network Systems, *IEEE Pervasive Computing*, October 2003.

[16] C. Lu, B. Blum, T. Abdelzaher, J. Stankovic, and T. He, RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks, *IEEE Real-Time Applications Symposium*, June 2002.

[17] L. Luo, T. Abdelzaher, T. He, and J. Stankovic, EnviroSuite: An Environmentally Immersive Programming Framework for Sensor Networks, *ACM Transactions on Embedded Computing Systems*, to appear.

[18] L. Luo, T. He, T. Abdelzaher, J. Stankovic, G. Zhou and L. Gu, Achieving Repeatability of Asynchronous Events in Wireless Sensor Networks with EnviroLog, *Infocom*, April 2006.

[19] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, The Flooding Time Synchronization Protocol, *ACM SenSys*, November 2004.

[20] M. Maroti, et. al., Radio Interferometric Geolocation, *ACM SenSys*, November 2005.

[21] D. Mills, Internet Time Synchronization: The Network Time Protocol, In Z. Yang and T. Marsland, editors, *Global States and Time in Distributed Systems*, IEEE Computer Society Press, 1994.

[22] A. Perrig, J. Stankovic, and D. Wagner, Security in Wireless Sensor Networks, invited paper, *CACM*, Vol. 47, No.6, June 2004, pp. 53-57, rated Top 5 Most Popular Magazine and Computing Surveys Articles Downloaded in August 2004, translated into Japanese.

[23] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. Culler, SPINS: Security Protocols for Sensor Networks, *ACM Journal of Wireless Networks*, September 2002.

[24] J. Polastre, J. Hill and D. Culler, Versatile Low Power Media Access for Wireless Sensor Networks, *ACM SenSys*, November 2004.

[25] N. Ramanathan, K. Chang, R, Kapur, L. Girod, E. Kohler, and D. Estrin, Sympathy for the Sensor Network Debugger, *ACM SenSys*, November 2005.

[26] R. Stoleru, T. He, J. Stankovic, Spotlight: A High Accuracy, Low-Cost Localization System for Wireless Sensor Networks, *ACM Sensys*, November 2005.

[27] R. Stoleru, T. He, and J. Stankovic, Walking GPS: A Practical Localization System for Manually Deployed Wireless Sensor Networks, *IEEE EmNets*, 2004.

[28] G. Virone, A. Wood, L. Selavo, Q. Cao, L. Fang, T. Doan, Z. He, R. Stoleru, S. Lin, and J. Stankovic, An Assisted Living Oriented Information System Based on a Residential Wireless Sensor Network, *Proceedings D2H2*, May 2006.

[29] M. Welsh and G. Mainland, Programming Sensor Networks with Abstract Regions, *USENIX/ACM NSDI*, 2004.

[30] K. Whitehouse, C. Karlof, A. Woo, F. Jiang, and D. Culler, The Effects of Ranging Noise on Multihop Localization: An Empirical Study, *IPSN*, April 2005.

[31] A. Wood and J. Stankovic, Denial of Service in Sensor Networks, *IEEE Computer*, Vol. 35, No. 10, October 2002, pp. 54-62.

[32] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, J. Stankovic, AlarmNet, *ACM SenSys*, April 2005.

[33] T. Yan, T. He and J. Stankovic, Differentiated Surveillance for Sensor Networks, *ACM Sensys*, November 2003.

[34] G. Zhou, T. He, J. Stankovic and T. Abdelzaher, RID: Radio Interference Detection in Wireless Sensor Networks, *Infocom*, 2005.

[35] G. Zhou, T. He, S. Krishnamurthy, J. Stankovic, Impact of Radio Asymmetry on Wireless Sensor Networks, *Mobisys*, June 2004.

[36] G. Zhou, C. Huang, T. Yan, T. He and J. Stankovic, MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks, *Infocom*, April 2006.

Index