Workplace Applications of Sensor Networks [†]

W. Steven Conner[1], John Heidemann[2], Lakshman Krishnamurthy[1], Xi Wang[2], Mark Yarvis[1]

[1] **Intel Research and Development**

[2] **University of Southern California, Information Sciences Institute**

USC/ISI Technical Report ISI-TR-2004-591

1. Introduction

The current generation of interactive devices and networks foster a wide class of interactive ubiquitous computing applications [7]. The recent trend to integrate wireless networking into interactive devices such as PDAs, cellular phones, and portable computers has led to the availability of information such as news and stock quotes, as well as services such as email, appointment tracking, and multimedia content from any location at any time. These applications have significantly improved workplace productivity, despite the fact that human participation is often required in the compute loop. These applications have traditionally interacted with *virtual* content such as email, financial records, and text documents.

Today millions of sensors are scattered throughout workplaces in both industrial and non-industrial office environments. These sensors include HVAC-monitoring devices such as thermometers, barometers, and moisture gauges, safety monitors such as carbon monoxide and smoke detectors, security monitors such as motion and glass break detectors, and access control devices such as RFID badge readers. In most cases, sensors are deployed for a specific application and access to sensor output is only available locally. A person typically must walk up to a sensor to obtain its current reading. In some cases, sensors may be wired to a nearby closed-loop monitoring station, but such monitoring stations are generally application-specific. While these sensors serve useful purposes to the individuals who deploy them, in practice each sensor is typically used only for a single specific monitoring application.

By networking these devices to provide ubiquitous access to remote information and actuation capabilities, many new applications emerge. The advent of inexpensive, low-power wireless sensors and self-configuring network technologies allows sensors to be easily deployed in a ubiquitous, ad-hoc manner. These deployments interface to the physical work and promise to make everyday tasks easier, enhancing our ability to examine and optimize the environments in which we live and work. Recent advances in sensor hardware make it feasible to deploy small sensors in office environments, but

---

[*] Other names and brands may be claimed as the property of others.

[†] To appear as a chapter in *Wireless Sensor Networks: A Systems Perspective,* Nirupama Bulusu and Sanjay Jha, editors, Artech House, publisher, 2004.

Authors are listed alphabetically.

For this work, John Heidemann and Xi Wang are partially supported through the NSF Division of Civil and Mechanical Systems, grant number E01-CMS-0112665.

many challenges remain. This chapter looks at two case studies in detail to explore those challenges: an application to assist workers in finding conference rooms, and another that guides visitors around an office environment. In addition to illustrating the challenges in developing and evaluating prototypes of real applications, these applications illustrate problems paramount to the office environment. The conference room application must integrate with existing networking and sensor infrastructure and interact with users in a useful manner. The visitor guidance application must consider human movement constraints and be easy to deploy and maintain.

In addition, both applications require self-configuring wireless networks and low-power operation (as do many other applications in sensor networks). These requirements might be surprising for in-building applications where power and networking are both comparatively plentiful. However, it is not always feasible to locate sensors near power or network outlets. Additional wiring would quickly exceed the cost-benefit ratio of these ad hoc applications. Even in new construction, each wired network port and outlet has a cost that must be justified. Thus we see low-power operation, energy harvesting, and wireless as necessities even in relatively wired environments. However, there is also an opportunity to leverage these sparsely available infrastructural resources for the benefit of the entire network.

We briefly review hardware that can be used to deploy workplace sensor network applications, followed by a detailed description of applications: conference room monitoring (Section 3) and visitor guidance (Section 4), and several applications briefly (Section 5). We then conclude by summarizing our experiences and identifying reusable components in these examples.

2. Hardware for Workplace Sensor Network Deployment

Four types of hardware platforms with heterogeneous capabilities are commonly used in the deployment of workplace sensor network applications: sensor nodes, display nodes, gateway nodes, and handheld nodes. These hardware platforms are tailored for sensing, human interaction with the sensor network, and interfacing the sensor network with workplace networks, and so they provide a mix of processing power and input/output capabilities. Each of the hardware building blocks described in this section should be viewed as representatives for a class of devices. Table 1 provides a comparative description of these devices.

2.1. Sensor nodes

A mote is a generic sensor node platform that integrates sensing, computation, and communication. Motes are typically low-cost, small, battery powered devices that are designed to allow large-scale deployment of sensors in an environment. An example of the Berkeley mote (Figure 1) that is commonly used in sensor network research and applications is the Mica-2. The Mica-2 mote is constructed using off-the-shelf components and includes an I/O connector to provide a

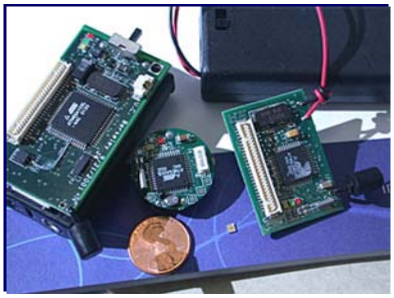


Figure 1: Mica2, Mica2 dot and Rene motes

Table 1: Comparison of various hardware platforms for use in wireless sensor networks

Node Type	Sample "Name" & Size	Typical Application Sensors	Radio Bandwidth (Kbps)	MIPS FLASH RAM	Typical Active Energy (mW)	Typical Sleep Energy (uW)	Typical Duty Cycle (%)
Generic Sensing Platform	Mote 1-10cm^3	General Purpose Sensing and Communications Relay	<100 kb/s	< 10 < 0.5 Mb < 10 Kb	3V * 10-15mA	3V*10uA	1-2%
High Bandwidth Sensing	Imote 1-10cm^3	Rich Sensing (Video, Acoustic, and Vibration)	~ 500 kb/s	< 50 < 10 Mb < 128 Kb	3V*60mA	3V*100uA	5-10%
Gateway	Stargate >10cm^3	High Bandwidth Sensing and Communications Aggregation. Gateway node.	> 500 kb/s – 10 Mb/s	> 100	3V*200mA	3V*10mA	> 50%

stackable platform for effective integration with sensors and alternative communication boards for experimentation. The Mica-2 optimizes power consumption, cost, and size, and is designed primarily to handle limited amounts of data from simple sensors and is not suitable for many sensor network applications that require collection of high bandwidth data, such as vibration, sound, or vision. The Intel Mote (Imote) increases processing capacity to provide an example of a device that can be used to sense more bandwidth intensive data and perform robust in-network communication. Many of the workplace applications described in this chapter use these motes.

2.2. Display nodes

Many workplace applications, including several described in this chapter, require simple user interactions at various points within the network, thus we must augment a basic sensor with simple, human-oriented input/output capabilities. The *button box* node (Figure 2) includes a Mica-2 mote and is powered by two AAA batteries. It provides a simple interface that includes two buttons for input and three LEDs and a buzzer for output.

While the button box is useful in many applications, a richer interface is sometimes required. The LCD display node (Figure 3) is a small, low-power wrist-watch form factor node designed to enable limited human interaction with a sensor network. This device consists of a Mica-2 mote integrated with an LCD capable of showing text and simple graphics and four control buttons. These buttons may be used to trigger the

(a) (b)

Figure 2: An external (a) and internal (b) view of the *button box* node.

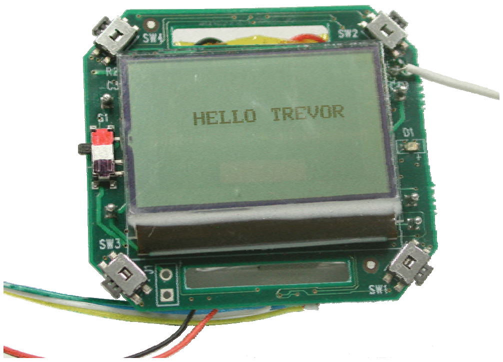


Figure 3: LCD display node.

Figure 4: Canby compact flash mote.

node to a wake up from deep sleep and also allow user text input. These devices provide an easy and inexpensive method to allow ubiquitous display and user interaction of information in the workplace.

2.3. Handheld nodes

While suitable for limited human interaction, handheld computing devices such as PDAs and laptops can provide sophisticated user interfaces and data analysis tools to humans. The Canby (Figure 4) is a compact flash card form factor Mica-2 mote that allows handhelds and laptops to easily interact with the sensor network. These devices may be used as part of a field tool. An example field tool TASK [4] allows the handheld to be used to query devices in the proximity of the human by sending out "ping" messages to which nearby nodes respond. In some workplace applications, a GUI on handheld nodes has been used to provide information from sensors that are in the human's proximity.

2.4. Gateway nodes

Previously described sensor nodes minimize cost and size by eliminating support traditional networks such as Ethernet or 802.11. Gateway nodes are devices capable of bridging communication between sensor nodes and higher-end wireless or wired networks. Gateway nodes often have more computing capabilities than sensor nodes as well as access to line power. An example of a gateway node is the Stargate platform (Figure 5), which includes a 400 MHz Intel XScale™ architecture-based processor, tens of megabytes of RAM and up to gigabytes of persistent storage. It is capable of interfacing directly to either a Mica2 or an Intel mote device and can bridge the data from the low-power sensor network to traditional networks including 802.11, Ethernet, and wide area networks. Additionally, the processing and memory provisions on the Stargate node allow it to act as a web interface to a sensor network. Sensor readings can be stored in a local database and queried over the web. Additionally, the same web interface can be used to actuate or manage the sensor network. As described later in this chapter, Stargates may also be used to create hierarchical networks that provide performance enhancements to reduce sensor network energy consumption and extend the lifetime of battery-powered nodes.

Figure 5: An Xscale(TM) architecture based gateway node

3. Conference Room Application

In many modern office complexes, closed-wall offices have been replaced with high-density cubicles to inspire an atmosphere of open collaboration and accessibility among employees. However, the lack of private offices makes it difficult for employees to hold impromptu meetings. Discussions in or near their cubicles often disturb other employees who are trying to work nearby. Buildings include cafés and other common areas, but these areas tend to be noisy and distracting and are not good candidates for important or private discussions.

Modern buildings have conference rooms for meetings, but these rooms may be reserved days or weeks in advance, and it is often not realistic to reserve a room with little or no notice. However, it is common for meetings to be shorter than the entire reservation time or to be entirely be cancelled without canceling the room reservation. Thus, it is often possible for employees to find an empty room for an impromptu meeting through exhaustive search.

Many conference rooms are equipped with motion detectors that are used to turn off the lights when the room is not in use to save electricity. In most cases these motion sensors are hard-wired to the light switch in a given room and are not accessible from outside the room. As a driving application behind the sensor network research project, we have networked these motion detectors in one building at Intel using multi-hop network protocols implemented on a collection of motes giving employees access to room usage status and allowing them to find empty rooms from their handheld, mobile, and desktop computing devices from anywhere and at any time

3.1. Architecture and Operation

The system consists of a network of sensors deployed in and around conference rooms. In-room sensors are connected to motion detectors (Figure 6 (a)), which monitor room occupancy status. A gateway node receives the sensor data which is aggregated

(a) (b)

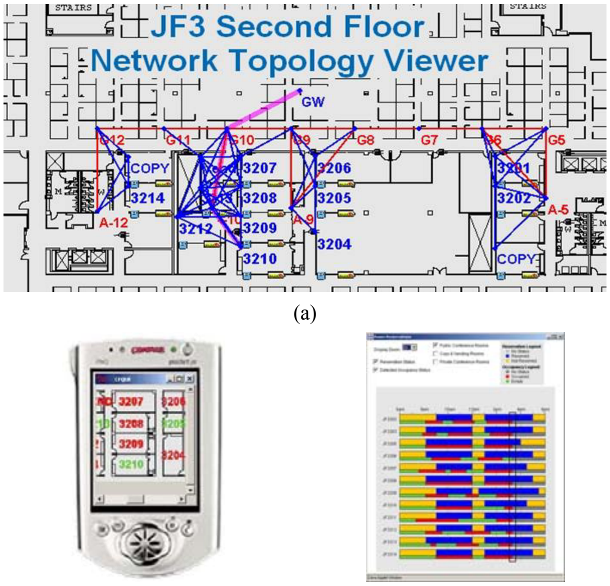
Figure 6: Conference room (a) motion sensor node and (b) reservation status indicator.

and stored to provide status information to desktop users over the web. Figure 7 (a) shows a screen shot from a web application that provides live occupancy information for rooms on a given building floor. Users of this application can avoid searching for a conference room and walk directly to an empty room. We have also connected PDAs to the sensor network, allowing mobile users to obtain the status of nearby conference rooms directly (Figure 7 (b)). Occupancy information is also available via status nodes at the end of the aisles that indicate the presence of an empty room in that aisle.

In addition to providing live occupancy data, motion detector data may also be compiled over time for future analysis. Figure 7 (c) illustrates an application that compares gathered room usage statistics with data from the online reservation system. Such data allows the automatic identification of individuals who consistently reschedule meetings without canceling room reservations and allows facility analysts to analyze building usage patterns. Typical conference room usage patterns can aid the design of new buildings. These usage scenarios require real-world information to be gathered, processed, stored and made available in a ubiquitous manner. In addition to these applications, the infrastructure has been reused to gather temperature and battery voltage/current usage at each node.

This application also makes room reservation status, normally available through an Outlook® reservation service, available at each room. Status nodes at the entrance to each conference room indicate current/future reservation status (Figure 6 (b)). This status information is pushed from the gateway to individual nodes, using the reverse path generated by the data collection tree. To save battery power these nodes can turn on when a user presses the status button, prolonging the life of the node.

In the currently deployed network, one of three LEDs are lit to specify whether the room is reserved or unreserved, or if the status is not known (typically only if the network is down). One of two buttons can be pressed to determine the reservation status in the current half hour and the next half hour. Because users typically use the device at the top or bottom of the hour and it can be difficult for the user to know the exact time, a flashing light indicates status for the first half of the hour, while a solid light indicates the status in the second half of the hour.



(a)

(b)

(c)

Figure 7: Applications. (a) Web page showing live conference room occupancy and network topology in the building. (b) PDA application showing status of rooms in the vicinity of the mobile user. (c) Occupancy history application comparing actual usage data to room reservations.

Clearly the current interface is not very rich, resulting in an overly complex user interface. Future versions of this node will include an inexpensive display to indicate current time and room ownership. This service helps resolve room reservation conflicts that are currently settled by phone calls to conference room administrators.

3.2. Application Challenges

Power was the most significant challenge we faced when installing these applications. To simplify the deployment of our application and to avoid installation of new wires, we used battery powered devices, allowing an ad hoc network deployment at minimal cost. The power issue is not limited to the conference room application. In our analysis of building- and factory-monitoring applications, eliminating both power and network wiring at the sensors results in a significant cost savings and improved return on investment.

These deployment and maintenance challenges motivate the design of our protocols. While we leverage power outlets available throughout the building, many of the sensors and devices are battery powered. One of our primary objectives is to take advantage of wall-powered nodes to yield energy savings for battery powered nodes. However, all battery powered nodes will not be within one hop of wall-powered nodes. A topology control protocol is required to leverage heterogeneity and a synchronization protocol is required to allow battery powered nodes to sleep yet still communicate [2]. Such battery conservation techniques can extend network lifetime to meet practical requirements. Building maintenance practices already require replacement of other consumables, such as light bulbs, at intervals of six months or a year.

3.3. Communication Protocols

Network protocols for the conference room application can be divided into two parts: sensing and actuation. The goal of the sensing task is to deliver the occupancy status of each conference room to the web server. We use a single-destination version of the DSDV protocol [6] to create a many-to-one data delivery tree to a sink node connected to the web server. An end-to-end reliability metric, which sums the normalized log of the link success rates, is used to select paths with greatest chances of delivering data to the sink, as described in [10]. Each sensor periodically sends a packet to the sink containing the node number (one byte), the room number (two bytes), and the room's occupancy status (one bit). Nodes along the delivery path append their own node number and occupancy status. For each packet received, the web server obtains the room number to node number mapping for the originating node and the occupancy status for several rooms.

Actuation in this application provides room reservation information to nodes outside of each conference room. Since reservations change infrequently, an entire day's worth of reservations for each room are pushed every few hours. To enable one-to-many communication, the server uses the list of forwarding nodes (essentially a traceroute) in each incoming occupancy status packets to track the topology of the data delivery tree. The server is an ideal candidate for this function as it is not memory constrained. Reservation status packets generated by the server can then be source routed by reversing the path from the data delivery tree. Current reservation packets consist of a reservation status bitmap and a timestamp with half-hour granularity that corresponds to the start of the bitmap. In a future version of the application, the reservation status packets will also include the name of reservation owners to help users to resolve room conflicts. At the start of each half hour, a packet containing the current timestamp is flooded into the network to provide time synchronization. Latency of a few seconds or less provides sufficient synchronization for the reservation status user interface.

While the conference room application could utilize a flat multi-hop network, Ethernet and 802.11 connectivity are common in conference rooms and can be used to create an overlay network. By tunneling sensor network packets across the IP-based infrastructure backbone, the sensor network can utilize a highly-reliable, high-bandwidth communication channel. To create the backbone overlay, we deployed Stargate nodes in several conference rooms on each floor and one at the sink. Each Stargate is attached to a mote, and includes software to receive sensor network packets from the mote and tunnel them to the sink node. Because route update packets from the sink node flow across both the sensor and overlay networks, the Stargate-enabled motes will be able to advertise a favorable routing metric (indicating a highly reliable path is present), causing nodes to form clusters around each Stargate. Data from nodes in these clusters flows across the sensor network to the Stargate node and then across the backbone network to the sink. Use of the overlay network reduces the depth of the data delivery tree, thereby increasing network reliability and decreasing the amount of energy that nodes must spend forwarding packets.

4. Follow-Me Application

Navigating an unknown place can be difficult. While signs may guide the way, and computer kiosks may provide room numbers and maps, neither provides active assistance to visitors as they move through a building.

The *Follow-Me* application is an active visitor guidance system designed to address this problem. Sensor nodes are deployed around a building on walls, one at each office doorway. Nodes blink their lights to indicate a path, guiding a visitor with a "breadcrumb trail" to their destination.

Although we describe this problem in the context of an office application, follow-me represents a class of applications where sensors are deployed to assist navigation. Other examples include marking paths in buildings damaged by earthquake or fire, and underground exploration. The key innovation demonstrated in follow-me is the *deployment-order* approach to topology configuration.

4.1. Hardware

Sensor nodes in this application are "button boxes". Figure 8 shows a possible deployment scheme, with 85 button boxes deployed in an office building. Node deployment is based on two general guidelines: There should be one node at each office doorway, and the distance between two adjacent nodes should not be too large. The later means we need to place additional nodes along hallways with few doors, such that visitor can follow lights easily. A touch-screen display at the entrance allows visitors to select a destination.

We are in the process of completing deployment at ISI. As of April 2004, our current deployment is smaller, with eight button boxes covering one long hallway at half the desired density, and with two button boxes with labeled buttons substituting for the touch-screen display.

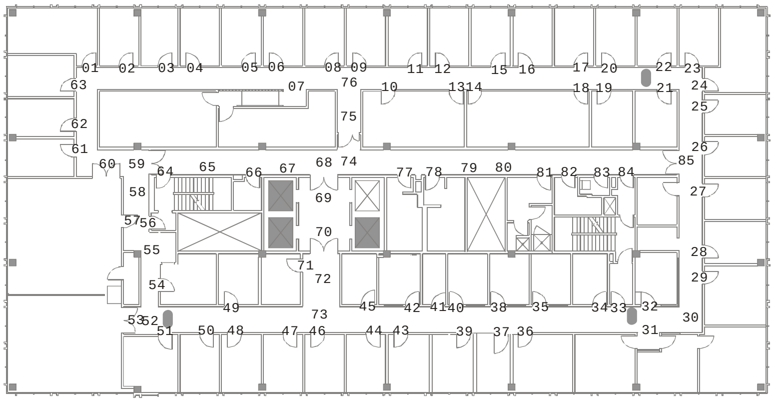


Figure 8: "Follow-Me" deployment example

4.2. Protocols

The follow-me application must guide visitors along appropriate paths. While network routing algorithms specialize in path finding, they are not directly applicable to guiding humans who are constrained by physical walls and prefer to follow adjacent nodes. Traditional routing algorithms select the shortest path based on radio connectivity, selecting paths through physical walls and skipping physically intermediate nodes when possible. Even a strictly geographic routing algorithm will cut corners and pass through walls if it shortens the physical path.

Thus the main technical challenge in follow-me is determining the *logical topology* that connects nodes as a human would walk, as opposed to the radio or physical topologies. Figure 9 compares radio and logical topologies for the same follow-me deployment shown in Figure 8.

It is not easy to capture logical topology because it is defined by human constraints such as walls and doors that are not visible to sensor nodes. We next describe *deployment order* in Section 4.2.1, our algorithm that captures logical topology. It is present when a network is first configured, allowing construction of complex topologies with minimal human interaction. We then consider how we can build follow-me by layering a simple routing algorithm over this logical topology.

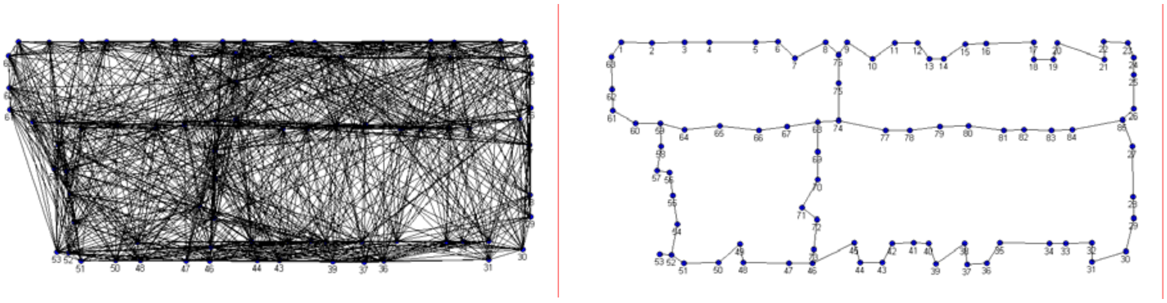


Figure 9: Comparison between radio connectivity graph (left) and logical topology (right)

4.2.1. Deployment order

As described earlier, each node in the follow-me application needs to be configured properly with its logical location, which is a set of physical neighbors. We would like nodes to configure themselves automatically. With localization hardware, it is possible to derive logical locations from physical coordinates. However we developed a new method for two reasons: First, the system must be easy to deploy and have low cost. This constraint requires methods that work without specialized localization infrastructure or specialized hardware. Second, the system must work well with building-like topologies: long, linear segments, parallel hallways, and moderate density.

We developed the deployment order method for logical location configuration. For many instances, sensor nodes are deployed sequentially. If two nodes are deployed (switched on) one after another within a short time, we can assume they are closest neighbors to each other. Links between these closest neighbors can create a linear path (not necessarily straight). If nodes detect and remember this path, it can be used later to guide visitors. Some other mechanisms are needed to handle “intersections”. One method is to manually interact with sensor nodes to add and remove links. The algorithm on each node can be described by simple state machines. We will discuss both linear paths and intersections below.

Linear Paths

To create a linear path, a newly deployed node communicates with previously deployed nodes to determine which one was deployed immediately prior to the deployment of itself. We can use the following state machine on each node. (**Figure 10**)

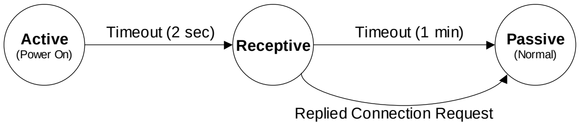


Figure 10: Initial state diagram of deployment order method

Below is a description about the three states in the figure:

Active: This is the state after a node is switched on. Nodes in this state send out connection request packets to look for neighbors.

Receptive: Nodes in this state will reply to connection request packets and establish links.

Passive: Nodes in this state will not be involved in link operations. This is the state for normal operation.

A node is in active state right after its deployment (switched on), and it sends out connection request packets. Every node in receptive state replies with a connection reply packet and adds the newly deployed node to its neighbor list. The newly deployed node also adds all replying nodes to its neighbor list as it receives their connection reply packets.

For example, after the first node is switched on, it won't find any neighbor and will go to receptive state. When the second node is switched on, it will find the first node. The first and the second node will link to each other. The first node will go to passive state and the second node will go to receptive state. Similarly, the third node will link with the second node, and so on.

In other words, only nodes in receptive state will accept link request packets and link to newly deployed nodes. A node will leave the receptive state after accepting a link request packet. We call this procedure the concept of “receptiveness.”

Intersection Handling

We can see that linear paths can be automatically configured while nodes are being deployed. Beyond linear path, we need to handle cases where nodes have more than two neighbors, which we call “intersections.” Assuming the majority of links belong to linear paths, we can still use the process for linear path most of the time, plus additional steps for intersections.

In our implementation, we use a button on each sensor node to toggle node states. When the node is in passive state, pressing the button will bring the node to active state. When the node is in receptive state, pressing the button will bring the node to passive state. The updated state diagram is shown in Figure 11.

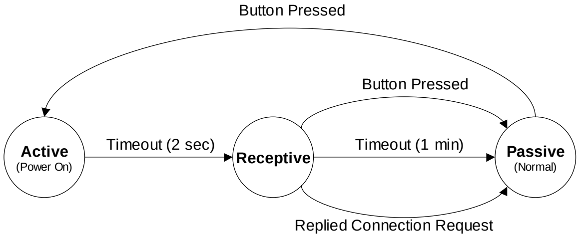


Figure 11: State diagram of deployment method – updated

With the ability to change state, we can add arbitrary connections by making one node active and another node receptive. An example of an intersection is shown in Figure 12. In this figure solid circles indicate nodes in passive state; empty circles indicate nodes in receptive state; and shadow filled circles indicate nodes in active state.

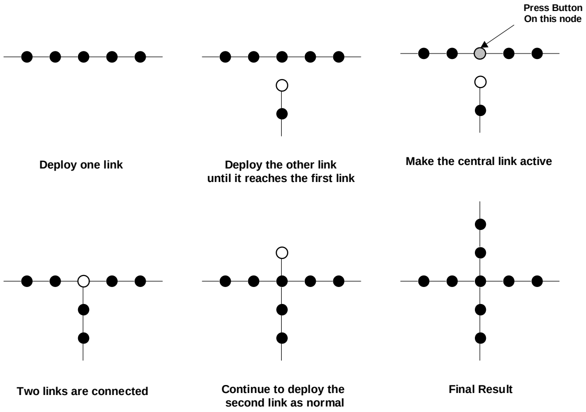


Figure 12: Intersection handling of deployment order method

Reconfiguration and Maintenance

The ability to tolerate failed nodes is an important feature for almost all sensor network applications. For the deployment order method, 1-hop node failures can be fixed by the following process:

When a node detects a failed neighbor, it will try to skip this neighbor and link to the neighbor's neighbors directly. This is done by broadcasting a “link fix” packet containing the ID of the failed neighbor. Only neighbors of the failed node will respond with their own IDs. New links between the sending node and responding nodes will be established, and old links to the failed node will be removed.

During normal operation, inadvertent configuration changes can occur if someone accidentally presses buttons on button boxes. To prevent this situation, buttons need to be locked after the node configuration process is complete. At the same time network managers may still want to change configurations periodically for network maintenance. We designed a lock mechanism that utilizes a “key” node to fulfill these goals. More details about the full system can be found elsewhere [8].

4.2.2. Routing

Given the logical topology it is relatively easy to build the follow-me application. We use a simple minimum-distance routing algorithm over the logical topology to determine the best path to guide a visitor between two points.

Our current implementation uses flooding to find forward paths and gradient style routing for reverse paths. This routing combination is very similar to directed diffusion [4]. We could also employ DSDV-style routing algorithms as described in Section 3.3, provided they operate on the logical topology.

When a visitor arrives at the lobby and selects a destination from a touch screen, the network finds the path as described above, flooding and establishing previous-hop gradients. The destination node gathers routes and selects the best one based on the desired metric. Unlike diffusion or DSDV where latency or energy is the metric of choice, we use physical distance traveled as the metric. (Our current implementation assumes all nodes are equidistant and so hop count is equivalent to physical distance; we are in the process of relaxing this approximation.)

While we use routing in the logical topology for follow-me, a general routing service is applicable to many tasks. For example, we monitor our network from a central point using this same routing algorithm. In this case our routing algorithm uses radio connectivity rather than logical topology.

4.3. User interaction

A common and effective approach of designing sensor network applications is to keep sensor nodes simple, and rely on the collaborative behavior of the whole network to achieve complex functions. Unlike systems with a keyboard and a screen, simple devices such as LEDs and buttons are more frequently used, and sensor nodes are spatially distributed in the target environment. The user interface part of the follow-me application shares the same idea.

As described earlier, a touch screen driven by a gateway node can be used for visitors to choose their destinations, and synchronized blinking patterns across the network are used to show paths to visitors. These blinking patterns should create a visual effect of moving light dots or lines, communicating both path and direction information to visitors in an intuitive way.

An interesting user interaction problem is how to guide multiple visitors at the same time. A possible solution is to show several paths simultaneously using different colors and/or blinking patterns. To reduce chances of mixing paths among visitors, we can limit blinking to nodes within visible ranges of visitors, provided there is a method to sense visitors' locations.

5. Other Applications

In addition to these two applications, several other in-building applications are being considered at Intel, ISI, and elsewhere.

The "voting app" was developed to provide feedback from an audience to a speaker, without requiring installation of a wired input device at each seat in an auditorium. Audience members are given a button box (described in Section 2). Using the buttons, the audience can respond to a question from the speaker or suggest the speaker speed up or slow down. The LEDs are used to indicate the user's current vote. In the future, other types of sensors may be employed to detect voting box motion and orientation, allowing richer audience participation. Votes are delivered to the speaker's laptop using the same single-destination version of DSDV used in the conference room application, which forms a data delivery tree. While data collection could include vote aggregation at each branch of the tree, we chose to deliver each individual vote, which reduces the impact of packet loss on the outcome of a vote. At regular intervals, each node sends a packet containing the originating node's identity and vote, and each forwarding node appends its identity and vote. This application was deployed at several Intel CTO keynote presentations, primarily as a demonstration of sensor networking technology. As such, network topology was also presented, using the route taken by each packet to dynamically identify the data delivery tree. Like the conference room application, this application can also utilize a hierarchical topology to increase the scale and reliability of the network.

At ISI we are exploring a security application that exploits multiple classes of sensors to balance privacy and security [7]. We place a video camera in our building lobby, a public area with a security guard where visitors have little expectation of privacy. We augment this sensor with motion detectors throughout the building in hallways and at office doors. Motion detectors can sense and timestamp the presence of an individual, but it cannot tell who they are or capture their photograph, thus there is no direct way to observe office occupants. If there has been a theft or security violation, we map the path of an individual from the site of the violation (perhaps a particular office) through a time-related series of motion-sensor detections, back to the lobby and ideally a photograph of the thief. This application seeks to balance privacy and security, allowing investigation of problems, but avoids pervasive cameras and explicit search to extract information.

Labscape [1] was developed by the University of Washington and Intel Research Seattle lab. Labscape is a smart environment that combines sensing and traditional ubiquitous computing to improve work flow in a cell biology laboratory. Labscape's focus on creating a real world application for its users has provided insights on design approaches, evaluation methods, and implementation challenges [1]. Labscape provides workflow automation in two phases: Experiment preparation and execution. In the preparation phase, Labscape allows researchers to plan the experiment using graphical flow charts. These flow charts guide the execution of the experiment and are place-holders to document results. During the execution of an experiment, users may log each step and annotate it with experimental results. Data logging during the execution phase is simplified using bar-code scanners and RFIDs to identify physical objects. The system also allows users to link pictures, diagrams, and hand written notes. Unlike traditional lab work that requires separate document and record book logging, Labscape seamlessly integrates the documentation step into experiment execution phase. From the Labscape project, researchers learned that building systems that integrate user interaction and sensing is extremely difficult; integrating sensing is easy but presenting the data to the user is much harder.

The set of applications that have been explored in this chapter are those that use workplace sensing to improve day-to-day activities. Additionally, wireless sensor networks can play a significant role in improving traditional building automation, control, and maintenance. Large buildings are subject to micro climates, which are dealt with today on a manual basis based on complaints from users of building. Wireless sensing provides the opportunity for fine grain temperature and humidity monitoring and control. Monitoring vibration from heaters, coolers, pumps, and motors using wireless sensors enables pro-active maintenance to predict and prevent failures. Likewise, sensing temperatures inside switch boxes allows early detection of shorts and prevention of fires and failures. The combination of traditional applications and new applications enables a world of revolutionary uses of wireless sensors.

6. Reusable tools and techniques

The applications described in the preceding sections suggest the applicability of sensor networks to office environments. More important than these specific applications are the tools and techniques that are reusable in similar applications. Three such areas stand out: routing, leveraging existing infrastructure, and exploiting simple external interactions.

All of the applications described require multi-hop wireless routing. The details of the protocols vary, with DSDV or a simple diffusion-like protocol, and shortest-path, minimal loss, or logical topology as the primary routing metric. But despite the rich connectivity from widely available 802.11 and Ethernet in the office environment, the power and flexibility advantages of a lightweight, sensor-net-specific multi-hop routing protocol are important.

This observation does not imply that we should ignore existing infrastructure. In fact, our second observation is that there is a great advantage of exploiting that infrastructure where possible. Examples of this include using overlay routing to improve reliability in the conference room application, exploiting a web interface or handheld GUI in the conference room application, and employing a PC-based a touch screen in the Follow-Me application.

Finally, complementing this use of graphical interfaces, we observe that very simple interactions *can* be used successfully. The conference room application illustrates this observation in the contexts of sensing and user interaction: sensing the flashing LED on an existing motion detector greatly simplifies the problem of detecting if a room is occupied, while simple “busy” or “free” status of the room can be indicated by red and green lights on the LED Box. Three examples in Follow-Me are the use of deployment order information to collect logical information, the use of a simple state machine to patch together complex topologies, and the use of flashing lights plus the node’s physical position to guide visitors to their destination.

7. Conclusions

The workplace provides a rich and challenging environment for applications of sensor networks. This chapter has described two such applications, conference room scheduling and visitor guidance, in detail, and additional applications in less detail. From these applications common hardware and software themes are apparent. Unlike outdoor applications, hardware and software designed for workplace applications emphasizes user interaction and use of existing infrastructure.

While the specific workplace applications described above are of interest to specific users and industries, the techniques developed to implement and deploy these applications are generally useful to enable a broad class of workplace applications. Routing algorithms, overlay networks, and easy configuration are specific technologies that all make sensor networks applicable to in-building applications. A key challenge to the sensor network community is to demonstrate that workplace applications can provide a return on investment through ease of use, ease of management, and workplace productivity.

References

1. Arnstein, L., Borriello, G., Consolvo, S., Hung, C., Su, J., "Labscape: A Smart Environment for the Cell Biology Laboratory," *IEEE Pervasive Computing Magazine*, vol. 1, no. 3, New York, NY, July-September 2002, pp. 13-21.
2. Conner, W.S., Chhabra, J., Yarvis, D. M., Krishnamurthy, L., "Experimental Evaluation of Synchronization and Topology Control for In-Building Sensor Network Applications," *Proceedings of the Second ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '03)*, San Diego, CA, September 2003, pp. 38-49.
3. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., and Pister, K. "System architecture directions for networked sensors," *Proceedings of ACM ASPLOS 2000*, Cambridge, MA, November 2000, pp. 93-104.
4. Intanagonwiwat, C., Govindan, R., and Estrin, D., "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCOM '00)*, Boston, MA, August 2000, pp. 56-67.
5. TASK TinyOS Tooklit http://berkeley.intel-research.net/task/
6. Perkins, E.C. and Bhagwat, P., "Highly dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Proceedings of the Conference on Communications Architectures, Protocols and Applications*, London, UK, August 1994, pp. 234-244.
7. Rajgarhia, A., Stann, F., and Heidemann, J., "Privacy-Sensitive Monitoring With a Mix of IR Sensors and Cameras," Technical Report ISI-TR-582, USC/Information Sciences Institute, November 2003. (Submitted for review.)
8. Wang, X., Silva, F., and Heidemann, J., "Infrastructureless Location Aware Configuration for Sensor Networks," Technical Report ISI-TR-2004-589, USC/Information Sciences Institute, May 2004. (Submitted for review.)
9. Weiser, M. "The Computer for the 21st Century", *Scientific American*, Vol. 265 No. 3, September 1991, pp. 94-104.
10. Yarvis, M.D., Conner, W. S., Krishnamurthy, L., Chhabra, J., Elliott, B., and Mainwaring, A., "Real-World Experiences with an Interactive Ad Hoc Sensor Network," *Proceedings of the International Workshop on Ad Hoc Networking (IWAHN 2002)*, Vancouver, British Columbia, Canada, August 2002, pp. 143-151.

.