Real-Time Communications and Sensor Networks

Richard Arps, Robert Forester, Jungwoo Lee, Hui Cao

Abstract – In this paper, we will discuss multiple proposed protocols and technologies to aid in the new area of sensor networks. These proposed systems provide functionality including QoS, routing, power conservation, and event detection. These new systems must be designed and tested as existing protocol suites will not be sufficient for this new area of networking. Sensor networks, by their nature, are ad-hoc and highly mobile with demands on deadlines and reliable transmission. Sensor networks demand lightweight, low overhead protocols that conserve their limited battery life and provide good end to end service from any point in the ad-hoc network. Here we will go through a number of promising technologies that have been designed to meet these rigorous requirements.

1. Introduction

Wireless sensor networks are composed of small sensor devices, each of which are equipped with wireless communication interfaces. These devices are usually deployed in very large numbers. The distances between nodes are typically in the order of meters, and the network density is sometimes as high as tens of nodes per square meter. Sensor nodes are responsible for detection of events, observation of environments, and relaying of third party messages. Generally, information is gathered at sinks, which are more powerful nodes that are responsible for higher level processing and decision making.

While each node is capable of communicating with other sensor nodes, the nodes are generally limited in power, memory, and computational ability. As a result, only small pieces of data can be transmitted at one instant and over short communication ranges.

An area which contains many sensor nodes performing the same work is called a sensor field. Sensor fields can be scarcely or densely populated. Typically, sensory nodes sense the environment, collect data, and route data to the sink.

Sensor networks are a special class of ad-hoc mobile networks. The key issues relating to the design of sensor networks are:
- data centric paradigm
- location based
- real time
- scalability
- fault tolerance
- resource constraints

Data centric paradigm

In traditional networks, addressing is used to route data. The data is communicated between two systems through a route which contains addressed nodes. In contrast, sensor networks are data-centric. Data from various sources, sensing the same event in the same environment, must be aggregated. This does not give specific importance to any individual sensor node. Messages are not sent to individual nodes but rather to locations or areas based on data content.

Location based

Sensor networks deal with physical environments and hence must work with physical locations rather than logical IDs. In fact, sensor networks do not have permanent IDs assigned to them. One of the reasons sensor networks do not have permanent IDs is the large volume of sensor networks existing and the constraints on their memory and processing capabilities. As a result, when a query is given, it must refer to a location rather than an ID.

Real Time

Sensor network showing event notification traveling to sinks.
Since sensor networks are deployed in environments dealing with real world events, there is a necessity for real-time guarantees for communication and data transfer. For example, a sensor network may be deployed around a military camp to detect intruders. If we cannot guarantee that the troops will be alerted to the intrusion within a certain time of detection, it may be have deadly consequences.

**Scalability**

Sensor networks may be very dense and may span a large area. As a result, protocols must be designed to keep control overhead to a minimum, and maintain as little global state information as possible. If each node has to maintain global state information, we would obviously run into problems since each node has limited memory.

**Fault tolerance**

Sensor nodes are highly susceptible to failures. Many types of failures are possible, including environmental noise or obstacles, physical destruction, power depletion, and more. As a result, communication protocols must be designed such that the sensor network performs properly in the event of failure of many nodes. As in the example given previously, imagine a sensor network deployed to detect intruders. As time passes, some nodes may run out of power, others may be broken from previous intruders stepping on them, and some others may have been damaged from the weather. Even in the event that many nodes fail, we still want to ensure that our system will function as expected.

**Hardware constraints**

Given that sensor nodes are meant to be deployed in extremely large numbers for a low cost, the hardware is placed under tough constraints. The typical components in a sensor node are a sensing unit, processing unit, transceiver unit, power unit, and occasionally a mobilizer. The sensing unit interacts with the environment and converts the perceived data into digital signals. Also, the sensing unit should be simple and energy efficient. The processing unit manages the functions responsible for communicating with other sensor nodes. This unit typically has very limited processing power due to the cost and power constraints. The transceiver unit is responsible for sending/receiving data. The power unit is critical as it largely determines the lifetime of a sensor node. All hardware and protocols must be designed with power efficiency as a key constraint. Other optional components are mobilizers, localization hardware, and power generators.

**Resource constraints**

Sensor networks use small batteries to operate and radio transmissions take up most of this power. Hence power conservation is the key issue in sensor networks, especially at the communication layer. Energy efficient routes can be found based on available power in the nodes or the energy required for transmission in the links along the routes. An energy efficient route can be selected in a number of ways: by selecting a route which has the maximum available power among all nodes, by selecting the route that consumes minimum energy to transmit data from the node to the destination in the route, by selecting a minimum hop route, or by selecting a route in which the minimum power is larger than the minimum powers of the other routes.

### 2. Sensor Network Communication Architecture

The sensor nodes are usually scattered in a sensor field. Each of these scattered sensor nodes has the capabilities to collect data and route data back to the sink. Data are routed back to the sink by a multihop infrastructureless architecture through the sink. The sink may communicate with the task manager node via Internet or satellite. The design of the sensor network is influenced by many factors, including fault tolerance, scalability, production costs, operating environment, sensor network topology, hardware constraints, transmission media, and power consumption.

![Diagram of protocol stack](image)

**Protocol Stack**

The protocol stack combines power and routing awareness, integrates data with networking protocols, communicates power efficiently through the wireless medium, and promotes cooperative efforts of sensor nodes. The protocol stack consists of the physical layer, data link layer, network layer, transport layer, application layer, power management plane, mobility management plane, and task management plane. The physical layer addresses the needs of simple but robust modulation, transmission, and receiving techniques. Since the environment is noisy and sensor nodes can be mobile, the medium access control (MAC) protocol must be power-aware and able to minimize collision with neighbors’ broadcasts. The power
management plane manages how a sensor node uses its power. The mobility management plane detects and register the movement of sensor nodes, so a route back to the user is always maintained, and the sensor nodes can keep track of who their neighbor sensor nodes are. The task management plane balances and schedules the sensing tasks given to a specific region.

The physical layer is responsible for frequency selection, carrier frequency generation, signal detection, modulation, and data encryption. Thus far, the 915 MHz industrial, scientific, and medical (ISM) band has been widely suggested for sensor networks.

The data link layer is responsible for the multiplexing of data streams, data frame detection, medium access and error control. It ensures reliable point-to-point and point-to-multipoint connections in a communication network. The MAC protocol in a wireless multihop self organizing sensor network must achieve two goals. The first is the creation of the network infrastructure. Since thousands of sensor nodes are densely scattered in a sensor field, the MAC scheme must establish communication links for data transfer. This forms the basic infrastructure needed for wireless communication hop by hop and gives the sensor network self-organizing ability. The second objective is to fairly and efficiently share communication resources between sensor nodes. Another important function of the data link layer is the error control of transmission data. Two important modes of error control in communication networks are forward error correction (FEC) and automatic repeat request (ARQ). The usefulness of ARQ in multihop sensor network environments is limited by the additional retransmission energy cost and overhead. On the other hand, the decoding complexity is greater in FEC since error correction capabilities need to be built in. Considering this, simple error control codes with low-complexity encoding and decoding might present the best solutions for sensor networks.

Sensor nodes are scattered densely in a field either close to or inside the phenomenon. Special multihop wireless routing protocols between the sensor nodes and the sink node are needed. Traditional ad hoc routing techniques do not usually fit the requirements of the sensor networks. The networking layer of sensor networks is usually designed according to the following principles:

- Power efficiency is always an important consideration.
- Sensor networks are mostly data-centric.
- Data aggregation is useful only when it does not hinder the collaborative effort of the sensor nodes.
- An ideal sensor network has attribute-based addressing and location awareness.

Transport layer is especially needed when the system is planned to be accessed through the Internet or other external networks. However, to the best of our knowledge there has been no attempt thus far to propose a scheme or discuss the issues related to the transport layer of a sensor network in literature. TCP with its current transmission window mechanisms does match the extreme characteristics of the sensor network environment. An approach such as TCP splitting may be needed to make sensor networks interact with other networks such as the Internet. In this approach, TCP connections are ended at sink nodes, and a special transport layer protocol can handle the communications between the sink node and sensor nodes. As a result, communication between the user and the sink node is by UDP or TCP via the Internet or satellite; on the other hand, communication between the sink and sensor nodes may be purely by UDP-type protocols, because each sensor node has limited memory.

Application layer protocols for sensor networks remain a largely unexplored region. In this survey, we examine three possible application layer protocols: Sensor Management Management Protocol (SMP), Task Assignment and Data Advertisement Protocol (TADAP), and Sensor Query and Data Dissemination Protocol (SQDDP), needed for sensor networks based on the proposed schemes related to the other layers and sensor network application areas. All of these application layer protocols are open research issues.

Routing

Routing in a sensor network must be approached differently than routing in a standard network. Considerations for power, mobility, and network congestion must be taken into account. Here I will discuss a routing protocol researched at the Los Alamos National
Laboratory called Parametric Probabilistic Sensor Network Routing.

The basic principal of this routing mechanism is that of partial flooding. The probability of a node to retransmit a packet goes up if the packet has gotten closer to the destination, and goes down if it has gotten farther away.

With $k$ as the percentage of the distance to the base gained or lost we use the following calculations to get the next probability of retransmission.

$$P_{R_i} = \begin{cases} (1 + k) P_{R_{i-1}} & \text{closer to destination} \\ (1 - k) P_{R_{i-1}} & \text{further from destination} \\ P_{R_{i-1}} & \text{same or indeterminate} \end{cases}$$

These calculations provide a simple method of calculating a probability that accurately reflects the change in the packet’s location.

Below is a graph of retransmission probability in a two dimensional field with the initial transmission at (0,0) and destination at (1,0)

This method lends itself to networks whose topology is constantly changing and want to avoid continuous recalculation of route schemes. There is no requirement for knowledge of other sensors in the network, simply a distance from the base station, most likely calculated by GPS. This protocol also helps spread out the load of those sensors closest to the base station. In a typical routing setup, one node may be designated as the next hop for many others and may drain its battery quickly retransmitting packets for these other nodes. With this protocol, each node within range of the previous node has equal probability of performing the retransmission.

The downside to this mechanism is that it results in wasted life of limited power sensors that retransmit the packet either after it has already been received by the destination, or retransmit the packet in the wrong direction.

RAP is a communication architecture used for various sized sensor networks. RAP provides features to enhance the communications in a real time, large area, ad-hoc network compared with other protocols.

3. RAP

Another consideration in a sensor network needs to be traffic management. Here I will discuss the proposed protocol suite RAP. RAP is a lightweight protocol suite that aims to reduce missed deadlines in packet transmission.

One major part of RAP is Velocity Monotonic Scheduling (VMS). The data in sensor networks is, by the nature of sensor networks, needed in real time. The percentage of late packets, using traditional methods, goes up as the distance from the base station goes up. It is necessary requirement to have a protocol in use in the network that provides the lowest percentage of late packets possible. The VMS protocol illustrates one way of achieving lower percentages than can be given by using preexisting protocols. Unlike using deadlines, where a packet is given a hard deadline when it enters the network, when a packet enters the network in a RAP system, it is given a requested velocity. A packet’s velocity determines how fast it needs to move through the network to make its deadline. That being determined, if the packet is able to move through the network at this velocity, it will make its deadline.

Each node in the network knows its goal time for forwarding a packet by using its velocity. The velocity is a unit of time that gives, after subtracting the transmission and propagation time for that hop, the maximum time goal for the specific packet to remain at that node.

$$\text{Time in queue}=\text{velocity}-(\text{transmission time} + \text{propagation time})$$

In current ad-hoc networks the first come first served method for packet scheduling is used. This method is inappropriate for real time communication networks. If the FCFS method was used then no priority would be given to packets that need to travel longer distances. These packets have the same amount of time to travel, but many more hops to make as compared with packets originating from nodes closer to the base station. Because of this, they should be forwarded sooner to reduce the percentage of late packets in the system.

In a system using RAP the transmitting node knows the distance to the base station. The node also has knowledge of the deadline, either predefined or defined by the urgency of the data. With this information in hand, it can calculate the velocity. The velocity calculation can be done one of two ways: static or dynamic.

For the static calculation, the velocity of the packet does not change over its trip. The node would set the packet’s velocity to:

$$V=\frac{\text{distance\,(origin, dest)}}{\text{deadline}}$$

In current ad-hoc networks the first come first served method for packet scheduling is used. This method is inappropriate for real time communication networks. If the FCFS method was used then no priority would be given to packets that need to travel longer distances. These packets have the same amount of time to travel, but many more hops to make as compared with packets originating from nodes closer to the base station. Because of this, they should be forwarded sooner to reduce the percentage of late packets in the system.

In a system using RAP the transmitting node knows the distance to the base station. The node also has knowledge of the deadline, either predefined or defined by the urgency of the data. With this information in hand, it can calculate the velocity. The velocity calculation can be done one of two ways: static or dynamic.

For the static calculation, the velocity of the packet does not change over its trip. The node would set the packet’s velocity to:
This calculation gives a constant, higher velocity for packets farther from the destination. The calculation does have its drawbacks though. If there is little traffic on the network a packet will be forwarded very quickly. This will result in an unnecessarily high velocity on later hops in the network. The opposite situation also gives rise to problems. When a packet encounters a delay in the network its velocity is unrealistically low. Dynamic calculations can help in these situations.  

For the dynamic calculation, the velocity of the packet changes at each hop. Each node sets the packet’s velocity to:

\[ V = \frac{\text{distance(current, dest)}}{\text{(deadline-time elapsed)}} \]

This calculation gives a better indication at each hop as to the urgency of the forwarding for each packet. In this method if a packet is delayed and does not make its velocity, the velocity will be increased at the next hop. Similarly, if the packet encounters very little congestion early on, its velocity is reduced to reflect a lower urgency to forward. There is, however, an increased overhead for the packet transmission using this scheme. And with sensor networks, this extra calculation, using finite resources, is always something that should be kept in mind.

The queuing system for VMS based transmission also deserves some consideration. A packet should be inserted into the outgoing queue in an appropriate position using its calculated velocity. In a single queue system, this idea would result in a traversal of the queue for each insertion. A simpler solution would be to implement several FIFO queues, each with a designated velocity range. Using this method a packet could simply be dropped into the appropriate queue and transmission would take place from the highest velocity queue that is not empty.

These priority queuing systems do have the chance of causing starvation for packets with a lower velocity. There are two apparent ways to deal with this problem. First, a recalculation of priority, which is unlikely due to limited resources, or a periodic clearing of the queues of packets that have missed their deadlines.

Simulations have shown that RAP can reduce the deadline miss ratio from 90% to 17.9% for packets originating from nodes at a great distance from the base station.

The graphs shown below are from a test done at the University of Virginia Department of Computer Science. Four methods of testing were done. The first was FCFS, the second was a hard deadline system, the third was static velocity, and the fourth was dynamic velocity. The first graph shows the overall miss ratio of packets for all nodes in the network. The second graph shows the miss ratio of packets originating from the far corners of the network.

VMS is not enough to guarantee success in a sensor network setup. Each radio is competing for the same space and proper prioritization needs to be done to allow higher priority traffic through.

Another feature of RAP provides a mechanism to remedy this while introducing minimal overhead. To accomplish this, the initial wait time is set to a random number between 0 and MAX times the priority. This gives nodes with a higher priority, corresponding to a lower priority number, a statistically lower wait time.

\[ \text{Wait} = \text{Base_Wait} \times \text{Priority} \]

The backoff mechanism, rather than doubling the max wait time, increases the backoff in accordance with the priority of the packet.

These mechanisms give higher priority packets a greater chance to make their deadlines and further reduce the miss ratio for the entire network.

4. SPEED

SPEED is a real-time communication protocol for sensor networks. It provides 3 types of real-time communication services, namely, real-time unicast, real-time area-multicast and real-time area-anycast.
In real-time unicast, data is sent to a specific sensor node. This is the event where data is sent to the base station from a sensor node. In real-time area-multicast, data is sent to an area in the sensor network. This is effectively when the base station queries an area. In this scenario, multiple nodes are expected to respond. In real-time area-unicast, the destination is any one node in an area. The base station may query an area, but since the data sensed by all nodes in an area may be redundant, it is only necessary for one node to respond.

SPEED supports soft real-time based on feedback control and stateless algorithms. SPEED utilizes geographic location to make localized routing decisions. SPEED also handles congestion and provides soft real-time which location based protocols do not.

SPEED avoids the expensive (in terms of both power and delay) route discovery broadcasts in reactive routing algorithms. SPEED only maintains information about its immediate neighbors. This information is exchanged between neighbors with a periodic beacon exchange. It does not maintain a routing table or per-destination states, which means that its memory requirements are minimal.

Although SPEED provides soft real-time guarantees, it does not use any information related to deadlines. Instead it provides real-time guarantees by providing a uniform packet delivery speed across the sensor network so that the end-to-end delay of the packet is proportional to the distance between the source and destination.

SPEED does not require specialized MAC support and can work with existing best effort MAC protocols due the feedback control scheme that it employs. All distributed operations in SPEED are highly localized meaning that any action invoked by a node will only directly affect the local neighborhood of nodes.

Reactive routing protocols typically suffer from substantial delays when routes become congested. This is often due to route rediscovery. In SPEED, backpressure re-routing is used to re-route packets around congest links with minimal control over head and delay.

As mentioned before, all actions invoked by individual nodes in a sensor network utilizing SPEED will not affect the rest of the network. SPEED ensures that all operations are completely localized. This also helps to avoid broadcast storms which can lead to network meltdown.

A modification to the aforementioned SPEED protocol is the MMSPEED protocol. This protocol, as specified by researchers at The Ohio State University, Electrical and Computer Engineering college, is very similar to the VMS process in RAP. Instead of velocity, speed is used, but the formulas and concepts are the same, with some added features in the MMSPEED protocol suite. These will be now be discussed.

In addition to the speed priority system, this protocol has the ability to exploit the inherent nature of sensor networks to have redundant, non-equal paths to the destination. With traditional routing protocols these longer paths would normally be passed up, being used only after the nodes on the shortest path have expired. With the MMSPEED protocol, these routes can be used to alleviate the strain on the nodes in the shortest path, assuming that a route through these nodes can still satisfy the required speed. To accomplish the task of this multi-path forwarding technique, each node keeps track of several nodes that are closer than itself to the destination along with their recent packet loss rates and backpressure. Each node can then calculate if the nodes in its forwarding table are capable of meeting the speed requirements of incoming packets. With this information the packets can be spread out evenly among the resulting paths.

The system also has the ability to forward a packet on multiple paths simultaneously with a routing calculation similar to that of the previously discussed probabilistic system. Each node calculates the probability of the packet to reach the destination from itself and judges its retransmission probability upon that.

5. Event Detection

Data Service Middleware (DSWare)

DSWare layer exists between the application layer and the network layer. This middleware provides data service abstractions. In this architecture, routing is separated from both DSWare and the network layer since the group management and scheduling components in DSWare can be used to enhance the power-awareness and real-time-awareness of routing protocols.

Software Architecture in Sensor Networks
Framework of DSWare

Event Detection Services

Some of the key concepts of the event detection services are event hierarchy, confidence, and time semantics, and implementation issues.

A. Event Hierarchy

An event is an activity that can be monitored or detected in the environment and is of interest to the application. We group events into two different types: atomic events and compound events. An atomic event refers to an event that can be determined merely based on an observation of a sensor. A compound event can not be detected directly from observations; instead, it must be inferred from detections of other atomic or compound events.

B. Confidence, Confidence Function and Phase

The confidence is the return value of the confidence function specified in event registration. An event with a confidence higher than 1.0 is regarded as confirmed, i.e., the sensor network is highly confident that the event actually occurred. A confidence function specifies the relationships among sub-events of a compound event with other factors that affect the decision such as relative importance, sensing reliability, historic data, statistical model, fitness of a known pattern and proximity of detections. In each phase, there is a set of events that are likely to occur with meaningful context, while other events are less likely to occur. Using phase in this manner not only saves power in monitoring and event detection, but also increases the reliability of event detection.

C. Real-Time Semantics

Each sub-event has an absolute validity interval (avi) associated with it. The avi depicts the temporal consistency between the environment and its observed measurement. When an event consists of more than one sub-event, the time an aggregating node should wait for the arrivals of all these sub-events becomes an important issue. The delay of a sub-event’s detection varies according to sensors’ sampling period and communication delay. We should preserve a time window to allow all possible reports of sub-events to arrive to the aggregating node. After an event is detected, it should be sent to the registrants before the reporting deadline.

D. Registration and Cancellation

To register an event of interest, an application submits a request in the following SQL-like statement:

```
INSERT INTO EVENT_LIST
(EVENT_ID, RANGE_TYPE, DETECTING_RANGE, 
SUBEVENT_SET, REGISTRANT_SET, REPORT_DEADLINE, 
DETECTION_DURATION [, SPATIAL_RESOLUTION ] 
[, ACTIONS])
VALUES ()
```

The Subevent_Set defines a set of sub-events and their timing constraints.

```
Subevent_Set {Time_window, 
Phase_set, 
Confidence_function, 
Min_confidence, 
(subevent_1, avi1), 
((subevent_2, avi2),...)}
```

Registered events can be cancelled even before the Detection_duration is terminated by submitting a cancellation request. Event cancellation is similar to event detection. The difference is that it only needs to specify the event’s id instead of describing an event’s criteria.

Evaluation of Real-time Event Detection Services

The simulator simulates the detection of an Explosion (E) event that consists of a high temperature atomic event (T), a light atomic event (L) and an acoustic atomic event (A). To evaluate our event service, we use communication cost, reaction time and total number of missing reports as the performance metrics. For comparison, we choose a baseline which works as follows: Once a sensor detects an atomic event, it will directly send the atomic event report to the registrant. The registrant will use the same mechanism of event service to decide whether there is a compound event happening.

A. Performance in Reduction of Communication

Event detection scheme can process and aggregate data and thus reducing unnecessary communication without sacrificing real-time constraints. As a result, event detection can save a lot of energy since the communication cost dominates the energy consumption in sensor networks.

B. Performance in Reaction Time

The reaction time of the baseline increases rapidly from 15.1 seconds to 21.6 seconds. The reason is that all sensors will directly send atomic event reports to the registrant, which causes severe traffic congestion in the network. As a
result, the registrant has to wait for longer time to get the atomic event reports to do analysis. Obviously the baseline is not suitable for real-time applications.

**C. Performance in Completeness**

The number of missing reports for event service is very low, around 1 or 2, while the number using the baseline reaches 4. Because there are only 100 nodes in the experiment, which are uniformly divided into 16 groups, there may not be enough sensors to cover the range of explosions.

**D. Impact of Node Density**

According to the results of 400 nodes experiment, we can see that if the node density is very low, there will be missing reports. However, if the node density is high, there will be a lot of energy consumed by communication and the registrant may not be able to get the detected report in time. It is clear that there is a tradeoff between communication cost, reaction time and the number of missing reports for our event service.

**6. Power Management**

**Background and related work**

In one of the earliest deployed sensor networks, researchers at Berkeley and Intel Berkeley Research Lab put the network in sleep mode and used wake-up messages to inform the whole network (800 nodes) to enter the working mode when the event of interest happened. To implement the wake-up, time-based protocols appeared. A sensor node keeps awake form 100ms in every cycle of 4 seconds. This results in a duty cycle of 2.5%.

In some sensor networks, a fraction of the network nodes operate in full power mode or have a much higher duty cycle than other nodes in the network. These nodes, called sentries, provide a sensing coverage or communication back-bone as required by the application. If the application requires a full-time monitoring of possible events of interest, the sentry service selects a number of nodes as sentries to form a monitoring of possible events of interest, the sentry service selects a number of nodes as sentries to form a monitoring network, which provides the required sensing coverage. The sentry service saves energy by selecting as few sentries as possible. Also, sentries send out power management packets to wake up non-sentries when the events of interest happen.

**Basic radio-triggered power management and basic radio-triggered circuits**

**A. Basic radio-triggered power management**

The radio-triggered power management scheme aims to avoid the useless wake-up periods. In this scheme, nodes stay asleep after they decide to go to sleep. They do not wake up until a special radio signal is sent by another node. The special radio signal wakes up the sleeping node almost instantly. Consequently, we save all the energy spent in previous wake-up and listen intervals. There are four main requirements for radio-triggered power management.

- The node should wake up almost instantly when it receives a wake-up packet. Here instantly means that it should be ready to work after the wake-up packet has finishing arriving.

- The node should use approximately the same amount of energy in sleep mode as in power management protocols without radio-triggered support.

- The node should not wake up when the event of interest does not happen.

- The node should not miss wake-up calls – when the event of interest happens, the node should not keep sleeping and thus miss the event.

**B. Design of the basic radio-triggered circuit**

The radio-triggered circuit has two essential tasks. First, it needs to collect energy from radio signals, which are electromagnetic waves. Second it needs to distinguish trigger signals from other radio signals, thus meeting the requirement of not waking up for false positives. In the design of the basic radio-triggered circuit, both tasks are fulfilled by the antenna. If a certain radio frequency is present and the antenna is activated, the electrical energy on the antenna can be used to power other parts of the circuit, and generate an output voltage signal.

**Basic radio-triggered circuit**

**Performance evaluation of the basic radio-triggered circuit**

**A. Evaluation of the output voltage**

We mentioned that the output voltage, $V_{\text{out}}$, will be more than 0.6V. In this subsection, we use the SPICE simulator to evaluate the accurate value for $V_{\text{out}}$. SPICE is a circuit level simulator developed by Berkeley. It uses numerical methods to analyze the circuit and can trace the changes in the circuit against time. The voltage drop on $R_1$ represents the output voltage $V_{\text{out}}$. The circuit simulation shows that $V_{\text{out}}$ is 0.62V. This supports the evaluation of the effectiveness of the basic radio-triggered circuit.
B. Evaluation of the potential power saving

Tracking application system based on Berkeley Mica 2 motes. There are totally 1000 nodes randomly deployed. There are 10 events to be detected every day, and each event lasts 2 minutes. The each network node uses two 1600mAh AA batteries. The average voltage of the AA batteries during the lifespan of the network is 1.45V. The network nodes have two working modes- wake-up mode and sleep mode. The wake-up mode is the usual working mode with all the functional units ready to work, and the average wake-up mode current is 20mA. In sleep mode a node shuts down all its components except the memory, interrupt handler, and the timer. The sleep mode current is 100μA. When a network node changes from the sleep mode to the wake-up mode, there is a surge current on the average of 30mA for 5ms. The always-on scheme is a network without power management- all the nodes are on throughout the lifespan of the network. The rotation scheme is a wake-up/sleep schedule based scheme much like many current power management protocols. And the radio-triggered scheme uses the basic radio-triggered power management. We assume that the application requires the entire network to wake up within 4 seconds when an event occurs in order to provide more accurate tracking. It is assumed that power management message exchange in the network takes 200ms. Comparing the result with always-on and rotation-based power management schemes, it is found that the radio-triggered scheme saves 98% of the energy used in the always-on scheme, and saves over 70% of the energy used in the rotation-based scheme. Consequently, the lifespan increases from 3.3 days (always-on) or 49.5 days (rotation-based) to 178 days (radio-triggered).

7. Conclusion

The flexibility, fault tolerance, high sensing fidelity, low cost, and rapid deployment characteristics of sensor networks create many new and exciting application areas for remote sensing. In the future, this wide range of application areas will make sensor networks an integral part of our lives. However, realization of sensor networks needs to satisfy the constraints introduced by factors such as fault tolerance, scalability, cost, hardware, topology change, environment, and power consumption. Since these constraints are highly stringent and specific for sensor networks, new wireless ad hoc networking techniques are required. Many researchers are currently engaged in developing the technologies needed for different layers of the sensor networks protocol stack.

The RAP protocol suite lends itself to any size range of sensor networks while still providing good QoS to all nodes. This protocol, as discussed in the cited paper, does not fully detail some of the systems that could help a sensor network. Things such as multiple route paths to save battery life on shortest path nodes, and other routing techniques were not addressed in this paper, but a good proposal was offered by SPEED and MMSPEED.

SPEED and MMSPEED provide good routing and reliability systems for sensor networks. Their low overhead and high mobility capable protocols offer a good option for implementation in a sensor network that demands real time and QoS requirements.

A sensor network should be able to provide the abstraction of data services to applications. DSWare is a flexible middleware designed to hide unattractive characteristics of sensor networks including the unreliability of individual sensing and communication, complexity and necessity of group coordination, and large volume of dynamic data distributed all over the networks, to present a more general data service interface to applications. Event detection is one of the services that is most widely used in sensor network applications. Instead of providing only simple detection of atomic events, middleware architecture is developed that accommodates the data semantics of real-life compound events and tolerates the uncertainty and unreliability in sensor networks.

Radio-triggered hardware is presented and explored its applications in power management. By extracting energy from the radio signals, the radio-triggered hardware provides wake-up signals to the network node without using internal power supply. If adequate antennas are used, this wake-up mechanism does not respond to normal data communication signals and thus does not prematurely wake up the network node.

References

Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci “A Survey on Sensor Networks”, IEEE Communications Magazine August 2002


