Routing in Wireless Sensor Networks: Application on iFireControl

塞福(team leader), Zachary Lanman Imm ,杜乐，齐龙巴。

Abstract: this paper is about fire detection in building using a modified APTEEN routing protocol. Here we design a system called iFireControl which is a smart detection system for buildings, which is more water efficient than many current systems, while keeping its robustness.

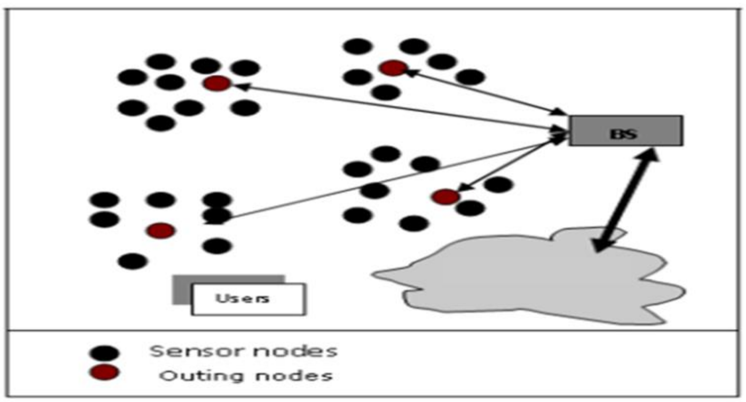
introduction

A Wireless Sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location.

The more modern networks are bi-directional, also enabling control of sensors activity.

The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; nowadays such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, Agriculture, Area Monitoring, Smart Home Monitoring, Seismic Monitoring etc.

Figure 1. A wireless sensor network structure.



Wireless Sensor Networks provide a bridge between the real physical and virtual worlds; allow the ability to observe the previously unobservable at a fine resolution over large spatio-temporal scales.

The WSN is built of “nodes” from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting.

A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth.The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.

Our paper is divided into three sections.in the first section we make a comparison of routing protocols suitable for fire detecting in buildings. the second part talk about the implementation of the iFireControl, the last section is about the simulation and the possible extensions of the iFireControl

Comparison of Routing protocols suitable for fire detecting in buildings

As said before our IFireControl system was aimed to deal with fire in buildings. To choose the most suitable routing protocol we considered several features of our case. First of all, fires in individual buildings happen very infrequently, which in terms of wireless sensor networks means that there is very small amount of events that will happen. Secondly, we need to be sure that all firefighting instances are notified on time to effectively cope with fire, which means that our sensor nodes must react immediately on happened fire and send information to the desired destinations. And the last, but not least feature is that we need always be sure that all sensor nodes are working correctly and have enough energy before batteries would be changed.

Considering features above firstly we thought about Rumor Routing protocol, because it performs best when the number of events is small. Rumor routing is a variation of directed diffusion and is mainly intended for applications where geographic routing is not feasible. In general, directed diffusion uses flooding to inject the query to the entire network when there is no geographic criterion to diffuse tasks. However, in some cases there is only a little amount of data requested from the nodes and thus the use of flooding is unnecessary. An alternative approach is to flood the events if the number of events is small and the number of queries is large. The key idea is to route the queries to the nodes that have observed a particular event rather than flooding the entire network to retrieve information about the occurring events. In order to flood events through the network, the rumor routing algorithm employs long-lived packets, called agents.

When a node detects an event, it adds such event to its local table, called events table, and generates an agent. Agents travel the network in order to propagate information about local events to distant nodes. When a node generates a query for an event, the nodes that know the route, may respond to the query by inspecting its event table. Agents also optimize the path to the event. If one agent comes across the path of another agent from the same or different event regions, it will record information about that path, after it could respond to the query if it has more optimal way to the event. Hence, there is no need to flood the whole network, which reduces the communication cost. On the other hand, rumor routing maintains only one path between source and destination as opposed to directed diffusion where data can be routed through multiple paths at low rates. This factor is very critical for our fire detecting system, because we must be sure that even if some node will break down, information will still reach the destination using another path. Rumor routing can achieve significant energy savings when compared to event flooding and can also handle node's failure. However, rumor routing is not quite suitable for time critical applications like our IfireControl system, because sensor nodes don't deliver information about event immediately. Considering this fact we rejected rumor routing and thought about TEEN (Threshold-sensitive Energy Efficient sensor Network protocol)

TEEN is hierarchical based routing protocol and was proposed for time-critical applications. In TEEN, sensor nodes sense the medium continuously, but the data transmission is done less frequently. A cluster head sensor sends its members a hard threshold, which is the threshold value of the sensed attribute and a soft threshold, which is a small change in the value of the sensed attribute that triggers the node to switch on its transmitter and transmit. Thus the hard threshold tries to reduce the number of transmissions by allowing the nodes to transmit only when the sensed attribute is in the range of interest. The soft threshold further reduces the number of transmissions that might have otherwise occurred when there is little or no change in the sensed attribute. A smaller value of the soft threshold gives a more accurate picture of the network, at the expense of increased energy consumption. Thus, the user can control the trade-off between energy efficiency and data accuracy. To save power cluster heads are changed periodically and are chosen among cluster members randomly. When cluster-heads are to change, new values for the above parameters are broadcast. The main drawback of this scheme is that, if the thresholds are not received, the nodes will never communicate, and the user will not get any data from the network at all. The nodes sense their environment continuously. The first time a parameter from the attribute set reaches its hard threshold value, the node switches its transmitter on and sends the sensed data. The sensed value is stored in an internal variable, called Sensed Value (SV). The nodes will transmit data in the current cluster period only when the following conditions are true: (1) The current value of the sensed attribute is greater than the hard threshold (2) The current value of the sensed attribute differs from SV by an amount equal to or greater than the soft threshold. The soft threshold can be varied. At every cluster change time, fresh parameters are broadcast and so, the user can change them as required. Important features of TEEN include its suitability for time critical sensing applications. Also, since message transmission consumes more energy than data sensing, so the energy consumption in this scheme is less than the proactive networks. TEEN protocol satisfies some of the requirements of our IFireControl system network, such as power consumption for small number of events and immediate response as for time critical applications. But still, TEEN doesn't have tools to periodically get information from sensor nodes to be sure that system is stable and all nodes are working. Considering this last feature we thought about APTEEN.

APTEEN is a hybrid protocol that changes the periodicity or threshold values used in the TEEN protocol according to the user needs and the type of the application. In APTEEN, the cluster-heads broadcasts the following parameters:

1. Attributes (A): this is a set of physical parameters which the user is interested in obtaining information about.
2. Thresholds: this parameter consists of the Hard Threshold (HT) and the Soft Threshold (ST).
3. Schedule: this is a TDMA schedule, assigning a slot to each node.
4. Count Time (CT): it is the maximum time period between two successive reports sent by a node.

The node senses the environment continuously, and only those nodes which sense a data value at or beyond the hard threshold transmit. Once a node senses a value beyond HT, it transmits data only when the value of that attribute changes by an amount equal to or greater than the ST. If a node does not send data for a time period equal to the count time, it is forced to sense and retransmit the data. A TDMA schedule is used and each node in the cluster is assigned a transmission slot. Hence, APTEEN uses a modified TDMA schedule to implement the hybrid network. The main features of the APTEEN scheme include the following. It combines both proactive and reactive policies. It offers a lot of flexibility by allowing the user to set the count-time interval (CT), and the threshold values for the energy consumption can be controlled by changing the count time as well as the threshold values. The main drawback of the scheme is the additional complexity required to implement the threshold functions and the count time. The main drawbacks of TEEN and APTEEN approaches are the overhead and complexity associated with forming clusters at multiple levels, the method of implementing threshold-based functions, and how to deal with attribute-based naming of queries.

Despite all drawbacks of APTEEN written above, still it is the most suitable routing protocol. Because it satisfies all three features of our IFireControl system network. APTEEN saves power by reducing the number of transmissions, there is threshold and sensor nodes transmit information about fire immediately, sensor nodes transmit information periodically so we can monitor the stability of the system.

<u>iFireControl implementation</u>

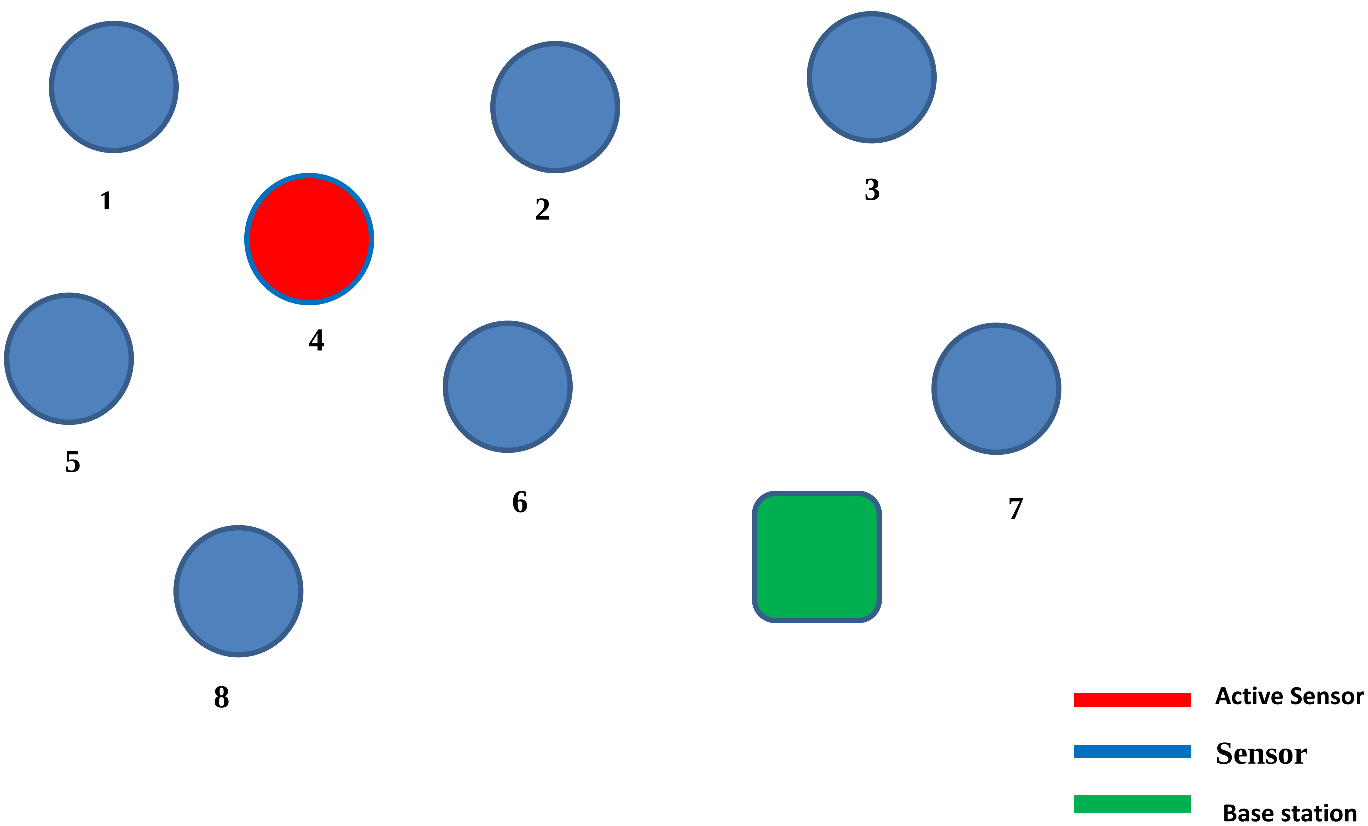


Figure 1: iFireControl sensors distribution sample

Suppose we have sensors distributed as shown in Figure 1. Each sensor is in a different room of a building. Let's assume fire broke out in one of the rooms, say room No 4, then the sensor No 4 which is in that room is said to be "Active". And we represent that sensor with a red disc. We also assume there is no other room on fire in the vicinity of eight sensors(rooms) shown on the above figure. The blue discs represent the sensors where no event(fire) is happening(non- active sensors). All the sensors are connected to a base station.

The iFireControl system is a fire detection system with specific requirements that are not always taken into consideration in nowadays. Usually system designers do not care about saving water for example .even though water is an abundant good on the globe, we know that it is also a limited resource that being affected with climate change. Therefore we decided to consider water as an important parameter as well as the robustness of the system and many other parameters that are listed on the system requirement below

iFireControl System requirements:

- Detect and localize network holes
  The system has to be able to function normally even when a sensor is not responding
- Save water
- Back up base station
  In case the one base station is burn out, the system should still be able to access the collected data and collected more from the sensors. We then keep a backup base station in a separate and remote place .
- Senses both smoke and fire
  avoid false alarm
- Alert fire department and maintenance service with information about the exact location of the fire(Eg: Building and room number , if fire is happening in a room).

Fire scenario

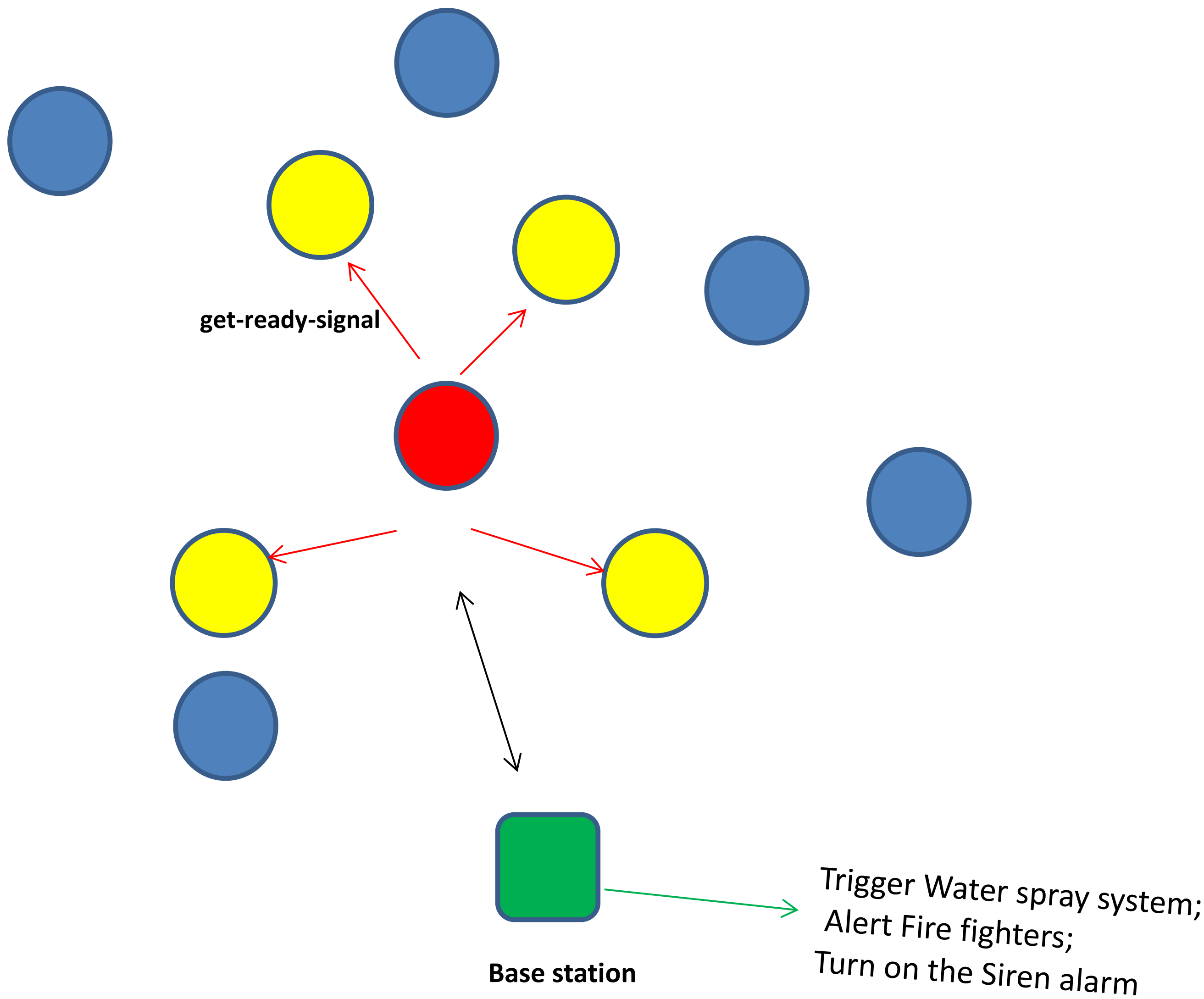


Figure 2: iFireControl event handling process

When fire occurs in a building where we have iFireControl installed, the fire will be put out immediately according to the process described on figure 2. On this figure we have a room on fire represented by the red disk, and the nearest room represented by the yellow disks. The active sensor sends a signal to the base station to report that fire is happening in the room where it's located. And at the same time the active sensor also sends a signal to its nearest sensors. This signal is called the "get-ready-signal". The get-ready-signal is to inform the nearest neighboring sensors that fire may break out in their respective rooms any moment from now. At this time the neighboring sensors are ready to report fire with a hard threshold that is smaller than the active sensor's hard threshold. The base station will trigger water and spray all over the corresponding room. The same base station will also turn on the siren alarm for eventual evacuation and alert the fire department with a signal containing information about the location of the burning room.

In the above scenario, unlike many fire detection systems, the iFireControl will only spray water inside the room where fire is happening, and eventually in the neighboring rooms. Since we are using APTEEN routing protocol in our system all the sensors in the cluster will be aware of the fire happening in one room. And they will also have the value of the "hard threshold" (temperature) that was sensed by the active sensor to report the fire. The neighboring sensors will report fire with hard threshold value that is lower than the active sensor's hard threshold value. Let **ATH** be the active node's hard threshold value and **NTH** be the neighboring node's hard threshold value .then

$$\mathbf{NTH = ATH - STH}$$

Where **STH** is the soft threshold. And the soft threshold is to be determined.

the APTEEN functioning here is a little different from the standard one whereby the soft threshold is to limit the number of transmissions or reports to the cluster head.in the standard APTEEN we would have NTH = ATH + STH. But knowing that the neighboring rooms of an active sensor are more likely to catch fire, the neighboring sensors will report fire if they sense a temperature close but less than the one that was sensed by the active node. This is a little modification we made on APTEEN to better satisfy our system.

iFireControl periodic check up

To make sure all the sensors are working properly and avoid holes in the system, iFireControl needs to perform a regular checkup. The checkup is made by the base station that sends status request to each cluster head, which in turn send the request to their cluster nodes and then send the reports back to the base station. If there is any broken node, the base station will inform the maintenance service that will fix or replace the sensor. Figure 3 is an illustration of the periodic check up.

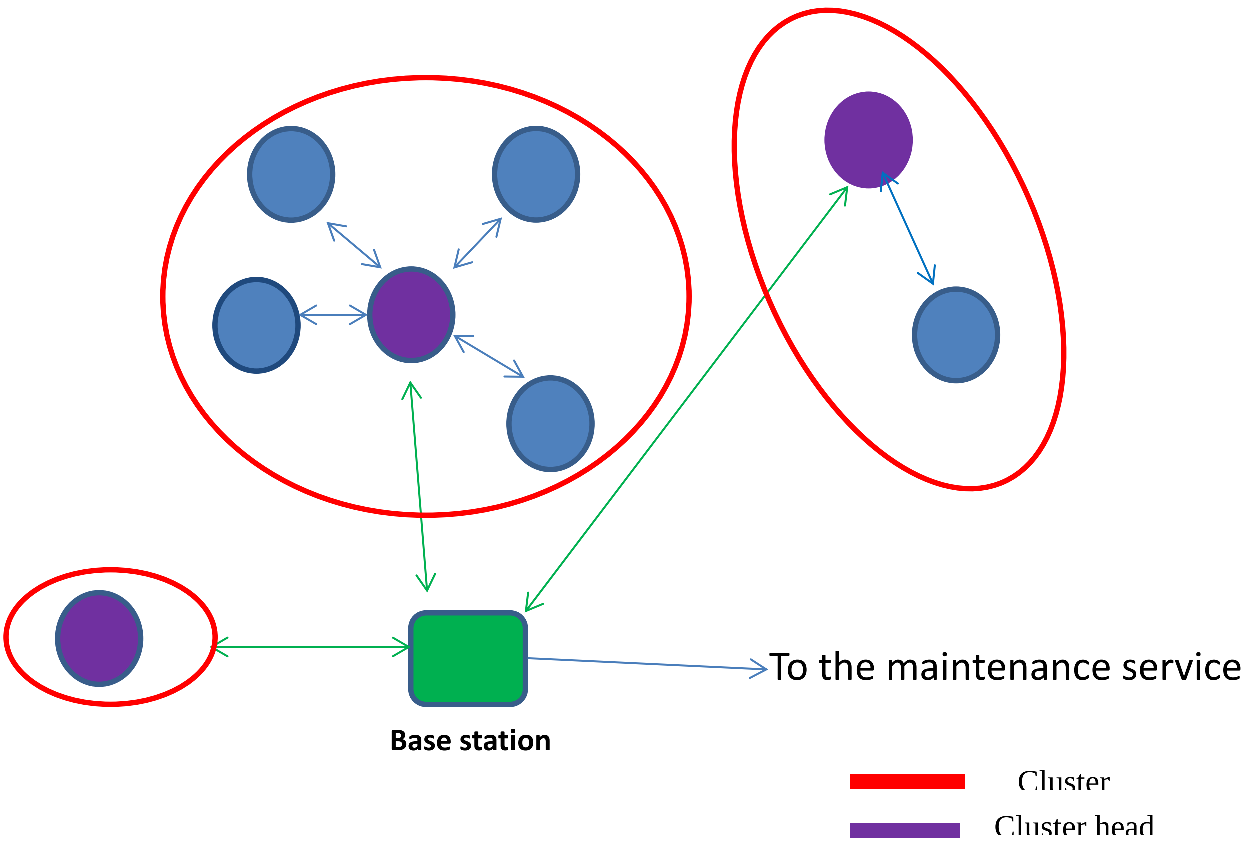


Figure 3: iFireControl system check up

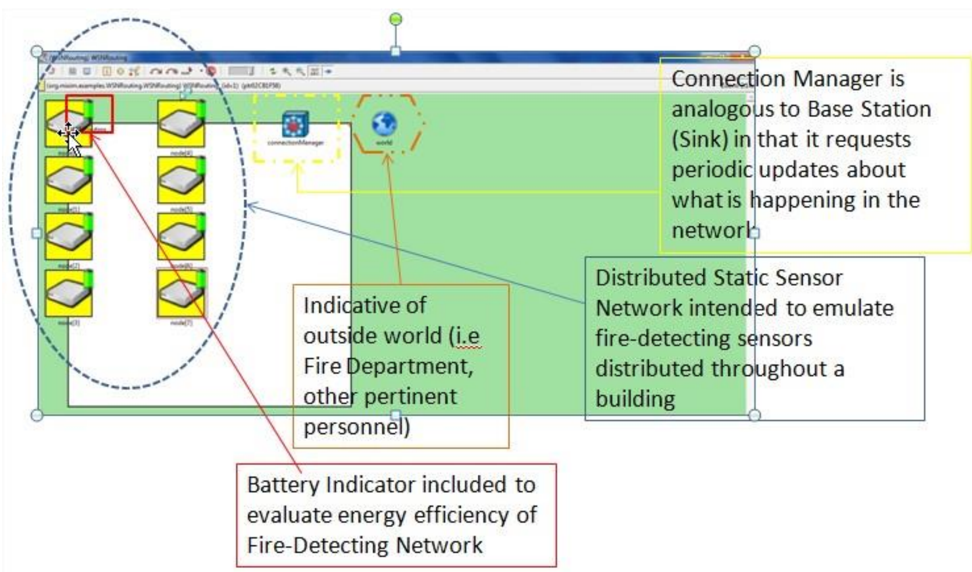
Simulation Details and Results

In order to test the effectiveness of the system discussed above we simulated the activity of a single cluster. In doing this test we hoped to verify that the system was stable at a cluster level, thus making it easy to scale in the future simply by adding more of the same clusters. When all was said and done, all of these clusters would then communicate with the Base station (sink) when an event occurred or periodically as specified by the APTEEN routing protocol. The simulation was performed in an environment known as Omnet ++. In order to better understand the testing that was performed, it is important to first understand the design of the testing environment.

Omnet ++ is a free simulation software designed to be used to simulate either wired or wireless networks. It is a discrete event simulation environment, which is perfect for our application, given that a fire is a discrete event. Omnet ++ also makes it easy to simulate a distributed sensor network as is required in our system. This is due to the fact that Omnet++ provides component architecture models that allow for the simple modeling of a sensor. These sensor models, which are programmed in C++, are then easily interconnected into a full network cluster using the high-level language, NED. Another key aspect of Omnet++, that made it an excellent choice for our simulation, is the GUI interface that it has. For the initial stages of simulation, this GUI interface made it easy to create the overarching framework of what we wanted. Then to further develop our simulation in order to more fully satisfy the requirements of our system, the C++ code was easily accessed in order to make the required changes. Alas, even though the Omnet++ platform provided an excellent base for our simulation, additional files packages were needed to fully implement our system. This requisite file package was known as MiXim.

The file package, MiXim, contained files created specifically for wireless sensor networks, as well as ad-hoc networks and vehicular networks. Given our wireless sensor network application, it was important to have files present that accurately simulate the effects of radio wave propogation, interference, radio transceiver power consumption (sensor power consumption in our case), and wireless MAC protocols (which are especially important for us because in the APTEEN protocol, the clusterhead uses TDMA to poll its subordinate nodes for status info).

Now that the specifics of the simulation platform as well as the requisite additional packages have been detailed, it is now appropriate to discuss the details of the simulation that was performed to test the effectiveness of our fire-detection system. To accurately simulate our system the nodes had to be stationary because in a building they would be in fixed locations in order to accurately monitor the outbreak of fire. In addition, the nodes would have to rotate who served as the clusterhead because this is a distinguishing feature of the APTEEN routing protocol. The power consumption of the sensor nodes also had to be tracked to effectively depict how our sensor network fire-detection system provided energy savings over currently used alternative fire-detection systems. Shown in Figure X is a screenshot of the fire-detection cluster layout.



<u>Figure X:</u> Fire-Detection System Layout with Comments

In order for the system above to be laid out as such, the C++ code underlying the model had to be modified. Shown below in Figure Y is an excerpt of the underlying C++ code with comments describing the modifications that were made to obtain the system seen above.

```
**.coreDebug = false
**.debug = false
**.playgroundSizeX = 250 m   #Establishes X-dimension size of Simulation environment
**.playgroundSizeY = 250 m   #Establishes Y-dimension size of Simulation environment
**.playgroundSizeZ = 0 m     #This line establishes that there is no depth in environment
**.numHosts = ${numHosts=8,25}   #The number of sensor nodes in the environment is set here

**.world.useTorus = false

**.connectionManager.sendDirect = true    # This line specifies initiates the use of the CM (Base Station)
**.connectionManager.pMax = 1.1mW       #Establishes max power that can be used sending to BS
**.connectionManager.sat = -100dBm     #Establishes sensitivity of the BS
**.connectionManager.alpha = 2.0    # The path loss exponent is set here
**.connectionManager.carrierFrequency = 2.4GHz  #Carrier frequency from Clusterhead to BS set here

**.node[*].nic.phy.usePropagationDelay = false
**.node[*].nic.phy.analogueModels = xmldoc("config.xml") #Configuration of network found in config.xml file
**.node[*].nic.phy.sensitivity = -100dBm  #Establishes sensitivity for the nodes in the network
**.node[*].nic.phy.maxTXPower = 1.1mW   #Max power of transmission established
**.node[*].nic.phy.initialRadioState = 0 #Initial radio state set zero
**.node[*].nic.phy.useThermalNoise = true #Thermal noise taken into account when analyzing system

**.mobilityType = "StationaryMobility"  #Mobility type is set to stationary due to nature of fire-detection
**.node[*].mobility.initFromDisplayString = false

**.mobility.speed = 2mps   #Speed irrelevant because stationary mode established above
**.node[*].mobility.initialZ = 0m  # All the nodes are located in the x-y plane

**.node[0].mobility.initialX = 0m  #Initial X-position of Node 0
**.node[0].mobility.initialY = 0m  #Initial Y-position of Node 0

**.node[1].mobility.initialX = 0m  #Initial X-position of Node 1
**.node[1].mobility.initialY = 50m #Initial Y-position of Node 1

**.node[2].mobility.initialX = 0m  #Initial X-position of Node 2
**.node[2].mobility.initialY = 100m #Initial Y-position of Node 2
```

Figure Y: C++ Code for Simulation (Includes Modifications Necessary for our System)

The full C++ code used to create the simulation can be found in the Appendix. Using the system shown above, the characteristic features of the APTEEN protocol, detailed as the most favorable for our application previously in this paper, can be shown. When simulated the network quickly establishes the physical layer and the data link layer (MAC) from its available libraries. Subsequently using both the Omnet++ and MiXiM packages in concert, the APTEEN protocol in the system's network layer can be approximated. This approximation includes the periodic inquiry for data from each of the nodes and well as the rotation of the clusterhead among the nodes.

(Add additional information about the simulation here. Don't know what to put though)

Extensions of the Fire-Detection System

Given that the fire-detecting system we have developed provides a number of benefits over the existing technologies that tackle the same problem, it is imperative that we ponder whether similar technologies can be implemented to provide even greater benefits. In the ensuing pages, we will develop a number of extensions to our fire-detection system. Some of them are very relatable to our technology while others are more abstract. The main binding characteristic among all of them is the APTEEN routing protocol, which provides an excellent framework to address infrequent, discrete environmental events.

The first extension of our system is to heating and air conditioning systems. With the price of energy rising and the focus of environmental groups growing ever stronger on the misuse of energy, there is no better time to implement a system which not only saves money, but the environment as well. As detailed throughout this report, our system uses a building-wide distribution of sensors to accurately and efficiently monitor whether fire has arisen in any room. Given that in many buildings as well as homes, central air-conditioning is used (meaning that there is one central air-conditioning unit connected to each respective room via duct work), it is possible to use augmented sensors to accurately judge whether people are present in a room and modulate air flow to that room accordingly. For example, if one of these sensors determined that a room had remained vacant for an extended period of time, it could send a message to the Base station, from which actuators in the ventilation system would be notified so that they could redirect air from that particular room.

This extension proves very feasible when you consider the attributes already present on the fire-detecting sensors. The temperature-sensitivity of the sensors could be used to monitor changes in the ambient room temperature that arise from human body heat. Through this method of detection, the system may even be able to make micro-climate adjustments to change air-conditioning levels depending on how many people occupy a room (as determined by the increased air temperature from numerous human bodies). To augment this temperature sensing ability, motion sensors could be added to these HVAC (Heating, Ventilation, Air-Conditioning) sensors in order to provide redundancy against people not changing the air temperature substantially as well as temperature variations arising from sources other than people (i.e drafts, open window, etc.). Overall this extension is very viable because given a little investment in the sensors themselves; the resulting savings due to efficient use of heating and cooling would easily offset this initial investment. This extension would also provide for a more enjoyable working climate for workers, students, etc. because the heating and cooling would be catered to their specific locations.

Another extension of fire-detecting system arises in the venue of warehouses. Warehouses are used by companies around the world to deal with supplies of goods that have yet to ship to their respective locations. While most warehouses are large, some have grown to a gargantuan scale. With these “mega” warehouses, it is challenging to monitor the whole place using manpower alone. Our fire-detection system provides an option for companies to more efficiently monitor their warehousing operation. If a fire arises in an area of the warehouse, the maintenance staff can be notified and the localized water spigots can be turned on. Through such a method, large scale fires can be efficiently avoided while excessive property damage can be avoided through localized use of sprinklers as opposed to warehouse-wide use of them.

A very important extension to our system arises due to the dwindling specter of smoking in the workplace and in school. In both these locations, cigarette smoke often is the catalyst that triggers a fire alarm to go off even though in reality no fire is present. These “false alarms” create unneeded disruptions to a worker’s day and can even cause significant reductions in worker productivity. This problem can be solved though using a system very much akin to that which we have already developed.

Using the fact that our fire-detection system tracks both the presence of smoke as well as the temperature variations that result due to the outbreak of fire, we can easily calibrate the system to disregard if someone is smoking near the sensing unit. This is done by recognizing that although a smoking individual creates smoke, the cigarette they are smoking does not create the requisite temperature variation as to elicit a response by the detection system. In terms of the APTEEN protocol, the hard threshold could possibly be triggered if smoking occurs close enough to the unit, but the soft threshold definitely would not be triggered because the smoking would not cause the air temperature to continue to increase, as is the case with fire. This extension is essentially solved using the fire-detection systems capabilities alone. To increase the robustness of the system additional measures could be taken to ensure cigarette smoke is differentiated from smoke caused by full-scale fire.

The last and most abstract extension of our fire-detecting system exists in the realm of nature. Given that our system is designed to deal with fires, which are infrequent events requiring immediate response, a similar system can be used in the case of wildlife tracking. Many nations and organizations around the world give considerable effort towards the tracking of various species, either to protect them or to learn more about the way these animals live. To this end, our system design provides an excellent means through which to do this. In order to more easily discuss the use of our system towards the detection of animals, we will assume that we are detecting the location of elephants. This is a quality assumption however given that elephants are endangered in many parts of the world.

To accommodate this seemingly far-fetched extension with no bearing whatsoever on fire detection, one simply needs to replace the event of fire with the event of extreme pressure. Through the distribution of static sensor nodes throughout a field, savannah, etc. one can monitor for extreme pressure, as would occur if an elephant stepped directly on the sensor. When this instance occurs, the surrounding nodes could be notified, as was the case with the fire detection, to further track the elephant based on vibrations, and thus determine its path through the terrain. Also using this system, the appropriate people (government officials, wildlife experts, conservationists, etc.) could be contacted because the location of the elephant is now known. Very minimal changes at most would have to be made to system we have already developed. Periodic monitoring of nodes ensures that all are operational and the rotational selection of the clusterheads ensures that nodes incur equal energy consumption.

Conclusion:

In this paper we designed a fire detection system that uses water in a smart way and that is invulnerable to broken nodes (holes).after comparing many routing protocols we found out that APTEEN was the most suitable for our system .But some modifications had to be made on the functioning of APTEEN to better satisfy iFireControl. iFireControl system can also be used to control the air conditioner of a building that uses a single air-conditioning system for all the rooms. we can save power and avoid cooling empty rooms .the iFireControl system can also be used for wild life monitoring to protect endangered species.

References:

1. A comparative study of wireless sensor networks and their routing protocols. ebnath Bhattacharyya 1, Tai-hoon Kim 1,* and Subhajit Pal 2

2. Akkaya, K.; Younis, M. A survey on routing protocols for wireless sensor networks. J. Ad Hoc Netw. 2005, 3, 325-349.

3. Lewis, F.L. Wireless Sensor Networks. Automation and Robotics Research Institute, The University of Texas at Arlington: Ft. Worth, Texas, USA, 2004; pp. 1-18.

4. Braginsky, D.; Estrin, D. Rumor routing algorithm for sensor networks. In Proceedings of the First Workshop on Sensor Networks and Applications (WSNA), Atlanta, GA, USA, October 2002.

5. Park, V.; Corson, S. A highly adaptive distributed routing algorithm for mobile wireless netowrks. In Proceedings of IEEE International Conference on Computer Communications, Kobe, Japan, April 7–12, 1997; Volume 3, pp. 1405-1413.

6. Embedded Everywhere _ A U.S. National Research Council Report, Eds. Estrin et al., National Academy Press 2001

7. Servicing & Maintenance of Fire Alarm Systems. CK Diong

8. Inspection, Testing, and Maintenance of Fire Alarm Systems—A Key to Life Safety.

dan finnegan, manager of industry affairs for siemens fire safety and securityDevices World Sdn. Bhd

9. The History of Fire Alarm Systems - Richard Bukowski, P.E., FSFPE, Wayne D. Moore, P.E., FSFPE

10. http://www.omnetpp.org/

11. www.wikipedia.org/wiki/Wireless sensor network

12. www.wikipedia.org/wiki/Wireless sensor network routing protocols

13. www.wikipedia.org/wiki/Wireless sensor network application