Anwar Hassen

Computer-based Positioning and Route Optimization System for Pedestrians

Helsinki Metropolia University of Applied Sciences
Bachelor of Engineering
Information Technology
Thesis
9 May 2011

Abstract

Author(s) Title Number of Pages Date	Anwar Hassen Computer-based positioning and route optimization system for pedestrians 43 pages + 4 appendices 5 May 2010
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Computer Networks and Security
Instructor(s)	Jarkko Vuori, Principal Lecturer Sami Isomäki, Project Manager

The goal of this project was to design a system that shows the distribution of wireless sensor nodes deployed for indoor navigation purpose. It was initiated by the demand for an optimized route a person should follow in order to reach a destination chosen by a service provider inside a building. The person was given a user device capable of displaying the paths she should follow.

The optimum path was calculated on a computer connected to a central node and the result was sent to the person requesting the service. A method for displaying and controlling the arrangement of wireless sensor nodes was designed. The distance between the fixed wirelesses nodes were given values that took traveling time into consideration. Distance was taken as a parameter in comparing and choosing the best path. The system was implemented independent of a particular service provider or a particular shortest path algorithm making it suitable for other applications with different requirements.

The result was a system capable of showing the distribution of wireless sensor nodes used for a navigation purpose. A mechanism that showed the paths used for traveling, and a method to change cost values that reflect both distance and traveling time.

Keywords	indoor navigation, shortest paths, wireless sensor networks

Contents

1 Introduction

The sophistication and usage of navigation systems has increased in recent years. Navigation systems have multiple applications ranging from guiding tourists to tracking the movement of objects of interest. Their main usage when used on roads is finding an optimum route to reach a destination when there are multiple alternative routes leading to the destination.

The navigation systems that are widely used have limitations when used for indoor navigations as the signals used are blocked by barriers such as walls and roofs. The signals that manage to pass may not be strong enough to be used to get the exact location of an entity. To solve this problem, already existing wireless networks deployed inside a building can be used.

This final year project was one part of an indoor navigation system that was carried out by Electria, an applied electronics research and development unit in Helsinki Metropolia University of Applied Sciences. The objective of this project is developing an application for managing and configuring wireless sensor nodes deployed inside a building, and finding an optimum route between two points of interest inside the building for which the navigation system is developed.

A service provider inside a building for which this system is implemented needs to guide a customer to a particular location inside the building. The customer is given a portable user device that displays only texts. The customer's location is tracked in real-time and the destination is provided by the service provider. This project also aims to show the distribution of wireless nodes inside the building for which the navigation is system is implemented. The nodes and the paths customers should follow need to be shown on a web browser.

2 Theoretical background

2.1 Wireless sensor networks

Wireless sensor networks (WSN) are widely used in monitoring environmental and physical parameters. They can make use of already existing radio options but the design of their system should take the limitation of the wireless sensor nodes used into consideration.

2.1.1 Characteristics

Monitoring an environment requires a large number of wireless nodes. These nodes typically are low cost, low power and have data processing and communication capabilities. For this purpose they contain microprocessors and radio transceivers. Being powered by battery, they are suitable for an environment that is hostile. Their computation power and storage capabilities are also very limited. [1,2]

WSNs are also self-configurable and any change in the communication network will not make the network unusable. As new nodes are added and others are removed due to failure or damage, the topology of the network changes frequently and unpredictably. [1,2] The addition of a node makes the WSN more stronger while the removal of a node has an opposite effect. A new added node will be part of the network without manual intervention making them self-organizing networks.

Addressing in WSNs is not global. [1,3] Since the number of nodes involved is large, it would be difficult to maintain the address scheme if each node is made identifiable globally. Nodes used for this project are numbered uniquely and their location name can also be used for administrative purposes. However all path calculations and internal communications refer to these unique numbers.

The data collected from each node should be aggregated and used by the application for which the WSN was built. For this reason most WSNs have a many-to-one traffic pattern where data from multiple nodes flows to a particular sink. [1,3] In this application data that is collected from each sensor node is sent to the node that is directly connected to the core system which may save the information for future use.

2.1.2 Sensors

Sensors, radio transceivers and microcontrollers are building components of a WSN. Sensors are devices used for interfacing between the analogue and digital world. Electrical devices process the converted signal and use actuators to convert an electrical signal into the physical world. [2,1] These physical world signals could be as simple as turning a led light on, which is understood by humans. Generally speaking sensors can be of an active or passive type. Active sensors interact with the environment and tend to consume more energy while the passive ones consume less as they just observe the environment. [3,23]

The sensitivity of the sensor is measured by the amount of change in the electrical signal for any change in the physical world. If for a very small physical phenomenon change there is a large change in the electrical signal, then the sensor is said to have a higher sensitivity. [2,2] Depending on the application for which the sensors are used, different sensitivity levels are required. Position sensors, for example, can give a real-time feedback on a moving object by detecting its motion, speed and direction. They convert this information in to an electrical signal that is processed by an electrical device such as a computer. [2,321]

2.1.3 Radio options

WSNs use different types of media for communication each having its own advantages and disadvantages. Radio waves are dominant in WSNs but Infrared, magnetism and acoustic are also widely used. [3,26]

IEEE 802.11 (WiFi)

WSNs implemented using the WiFi standard are used for applications with high bandwidth requirements. Their high power consumption is considered a drawback, as is the interference caused by the frequencies they use in the unlicensed band. [4,28] Since avoiding RF interference is not possible, some other measures should be taken. Studying the possible source of interference in the area for which the WSN is deployed is necessary. Strong signal coverage should be maintained and different channels that can avoid interfering of signals should be used. [5,307] As to the interference between Bluetooth and WiFi, it can be improved using power control, adaptive frequency hopping and packet fragmentation which is sending more but shorter transmission frames. [5,229] Unlike Bluetooth, the implementation of star and tree network topologies is possible with a data rate up to 54 Mbit/s for the 802.11a standard and a range that reaches 140 meters. [3,26]

Bluetooth

Bluetooth is used for short-range communication up to 10 m with a data rate reaching up to 3 Mbit/s, and all end-devices must communicate with a central host forming a star topology. [3,26] Bluetooth functions in the 2.4 GHz band with low cost and low power consumption. There are 79 channels of each 1 MHz for transmitting data and the device avoids interference by rapidly changing between these channels. Frequency hopping spread spectrum (FHSS), among others, is used to minimize interference. [5,28] Signal strength that affects transmission error rate and power consumption should be adjusted for optimum operation. [6,35]

RFID

Being used for identification of objects, radio-frequency identification (RFID) does not require a line of sight and can be used to transfer a higher data range at a faster data rate. It is used in product tracking and inventory systems, asset tracking, real time location systems (RTLS) and automatic vehicle identification, among others. The system contains readers which can be stationary or mobile and tags that are attached to objects that need to be identified. Information regarding the objects which are tagged is gathered by the reader if both are in the range. [7,20] One difference between RFID devices and wireless sensor nodes is that RFID devices identify objects whereas the nodes sense a given environmental parameter, collect data and transfer it for further processing. [1,11]

2.1.4 Applications

The low cost, delay and the applicability on all environments have made WSNs preferable over conventional wired network sensors. [1,3] The applications differ in what they detect. Some are motion sensors, some sense light and yet others sense temperature. A few of these applications are discussed in the following section, and the list is by no means exhaustive.

Environmental monitoring

When used for environmental monitoring different environmental parameters and conditions are monitored. [1,5] In an environment where humans cannot reach or cannot stay long enough to study its habitats, WSNs play a major role. For example, WSNs can be used to track the movement of certain animals or to learn seasonal variations seen in certain species of plants. In an environment like this WSNs are preferable since they do not require manual intervention, should some of the nodes fail. The WSN will re-organize itself making it suitable for any environment of interest. Air pollution monitoring, forest monitoring, green house monitoring and land slide detection are some other examples of environmental applications. [8]

The amount of air pollutants such as CO, CO2, NO2 and CH4, generated by vehicles and industry can be monitored using sensors. Comparing the measured values with expected levels different measures are taken. WSNs' use in forest fire detection is also significant. They can help fire fighters in determining when a fire starts and the way it is spreading. Temperature and humidity measurement can also be done using WSNs. when the values of parameter of interests drop below the expected, a notification message of different formats can be sent to those concerned. When used for land slide detection, WSNs can help predict their occurrences. Evacuations and preventative measures can be taken ahead of these occurrences. [8]

Industrial monitoring

Both manufacturing processes and conditions of the equipment used can be monitored using WSNs. Some processes span over a wide area which makes the monitoring and maintenance process expensive. Condition indicators are used with equipment and vehicles, which remove the necessity of scheduled maintenance, increasing up-time and reducing cost of service, as well as increasing customer satisfaction. [9,298]

The cost of maintaining and repairing is not avoidable throughout the life time of buildings. With proper ways of monitoring the cost of repairing can be reduced significantly. However these monitoring methods should not be expensive and the information they provide should be sent in real-time. WSNs could provide less expensive solutions as they do not require wiring and could simplify the maintenance of large number of buildings over a wide geographical area. Temperature and moisture are some of the parameters to be monitored in these types of applications.

Military applications

During wars getting information regarding enemy movements or locations of mines can be achieved using military sensor networks. [10,59] On the other hand each soldier can have a wireless sensor node creating a stronger network for communication than when each has to connect to a central location which consumes more energy. Each vehicle used during a war can also be made part of the network and feedback on its location can be collected in real time.

As to placement of nodes behind enemy lines, air-dropping is an alternative. Once the nodes organize themselves, they can start functioning as any other WSN would do. The same idea can be applied to border monitoring, which gives real-time feedback which otherwise would require a large number of personnel to cover thousands of kilometers. [3,17]

Multimedia applications

Multimedia applications generate too large an amount of data to be processed by a typical sensor node that relies on battery. A typical example of this application is Closed-circuit television (CCTV) networks implemented using WSN. [3,18] Programs using algorithms that require a large amount of resources pose the same problem. In such situations, processing the data and running the programs on a central node with sufficient power supply solves the problem to some degree. In this project, the shortest path was computed on a computer attached to the central node and the result was sent back to the node requiring the information.

2.2 Navigation

The sophistication and usage of navigation systems has increased in recent years. They are being used on roads, in the air and even on the oceans. The widespread use of smart phones with advanced computing capability has removed the need for a dedicated device for navigation. Though not widely used, indoor navigation systems are also emerging to cover the inside parts of buildings which are not normally reachable by the traditional navigation system due to barriers such as walls and roofs.

When moving towards a certain point that is not visible from the current position one takes known buildings, land marks or any known visible structure as a reference point. Without such reference points, the movement cannot be detected and is endless. One good example is the difficulties visually impaired people face when certain structures along their routes are changed. Unless they are notified about these changes, they will face considerable difficulties and accidents.

The same problem happens to travelers in the Sahara desert as the landscape changes every time. Every year many African immigrants die trying to cross the desert on their way to Europe. Though the excessive heat could be taken as one reason, it is not the only one. Since there are few signs to be taken as a reference, travelers rely on the landscape for navigation. In figure 1 the hills might be seen to be serving as reference points.

Figure 1. Changing reference points in the Sahara desert.
Reprinted from Nurie (2011) [17]

One may see the hills in figure 1 during the night and a totally different landscape in the morning. However the trees might help but only few of them are there and they will not stay there in all seasons. The other alternative could be for them to travel during the nights and use the stars for navigation .However many of them have neither the knowledge to use the stars for navigation nor the means to buy an expensive navigation device. Though they might find the north easily, it might not be good enough as dwellers of the desert do not live along a single route. For frequent travelers, it may be appealing to put black barrels every kilometer along the path and if they are lost, they just drive one kilometer back.

The barrels mentioned above are not applicable for boats sailing on the sea. Figure 2 shows a boat with no reference points.

Figure 2. A boat with no reference point.
Boat Owners of the United States (2011) [18]

In figure 2 one cannot see anything but the sky and the sea itself. No hills, trees or dwellers as were in the desert. The vastness of the water bodies does not allow putting barrels as signs. This is where a navigation system that uses altitude and longitude is useful.

The lack or invisibility of a reference point hinders an entity's movement towards its destination. Even with the presence of a reference point, an endless movement might result. This happens when the entity moves towards a moving object. In WSN applications, measuring the distance between two moving bodies is not as easy as when they are stationary.

2.2.1 GPS for indoor navigation

The distance between two entities can be measured in many different ways. When the distance between these entities is too large or one of them is unreachable by humans, time difference can be used. This principle is used in the global positioning system (GPS). In GPS, a method called triangulation is used. Since the speed of light is known, only the travel time of signals from a satellite to a receiver is required. Three satellites are used to find the exact location of a GPS receiver and a fourth one to increase the accuracy. [11,6]

There are at least six satellites in their orbit broadcasting signals that a GPS receiver uses to know its location. The user device needs to be away from any barriers such as buildings and mountains to receive these signals. [11,7] The passive GPS receiver has the capability of converting the raw data into position and speed. For the system to function properly it should be monitored and maintained. [11,8]

The usage of GPS indoors is limited as signals cannot penetrate concrete buildings. Also the signals that mange to pass may not be strong enough to be used by GPS receivers. However these weak signals may be used by applying some techniques. The receivers do not have the resources required to track these weak signals as the algorithms used are process-intensive and have high memory requirements. [12,277] WSNs are used for indoor navigation and the location of a receiver is determined with different techniques. One such technique is discussed in section 2.2.2.

2.2.2 Positioning

In any navigation system, knowing the exact locations of entities is desirable. When using WSN, both moving and fixed nodes should know their location in the area they are deployed. Configuring each node manually is neither scalable nor practical. [13,231] Knowing the position of a moving object is not easy and getting the exact location of a moving object referencing other moving objects is even harder. These situations are beyond the scope of this project and are not discussed further.

For a node inside a building, determining its location requires some reference points. Its distance from these reference points can be measured using different points. Using basic trigonometric functions the nodes coordinates in a two-or three-dimensional spaces, with reference to a given corner, can be determined. The main challenge is to get methods for measuring these distances.

The time of the arrival of signals can be used to measure the distance between two given nodes. Assuming a Signal’s speed under certain environmental conditions is known and the receiving node knows the time the transmission starts at the other end, the receiver can determine the distance from the time difference and speed. Since environmental factors can change, getting the right speed requires some adjustments. [6,236]

WSNs are made of nodes that monitor a given parameter and report to a central data collector unit. These nodes cooperate with each other in transferring the information. The nodes have the capability of processing the input data to a limited extent. A node contains a microcontroller, a transceiver, a sensor and a power supply and may contain more components. [1,7]

When nodes are battery-powered, the lifetime of their usefulness is very limited as replacing the batteries is required. Using environmental energy sources such as light vibration or temperature differences can increase the life time of the application. [1,8] However this advantage is not without any cost, and the power collected using these resources may not be sufficient to directly power the nodes. Also buffering may be required. [1,11]

2.3 Routing

When it comes to a navigation system, there are four main objectives that need to be addressed. The first one is knowing the current location of an entity. Without this information it is almost impossible to go further as there is no reference point to compare an entity's positions with. The second one is knowing the destination which refers to the positions the entity is supposed to reach. Having a variety of routes between the current position and the destination is the third objective, which could be a walking, driving, flying, sailing or any other route. The last one is for the entity to have a mechanism to make it able to choose the best route. [14] To achieve the first objective a wireless sensor node can be used, and the last two objectives require the knowledge of routing and routers. Both are discussed in the following paragraphs.

A router's two main tasks are forwarding packets and routing. Forwarding refers to the decision the router makes in choosing the exit line each arriving packet should follow. When making these decisions, the routing table is checked for the destination address. If the routing table does not contain any information as to where the destination address is located, the router broadcasts the packet when the source and destination addresses are in the same network. However if the source and destination addresses are in different networks, then the process continues in a recursive manner until the path to the destination is determined and the packet is sent to the chosen exit line. Otherwise the destination is declared unreachable and the packet is dropped. [15,350]

Routing on the other hand is concerned with populating and maintaining the routing table used for forwarding. The main parts of this process are routing algorithms and their associated data structures. [16,350]. Routers learn about networks not directly connected to them either manually or dynamically. Figure 3 shows how a router learns dynamically about networks that are not directly connected to it.

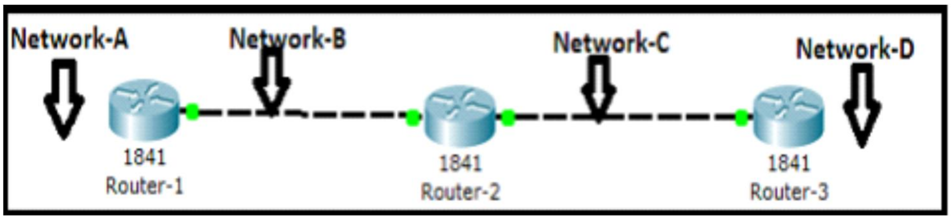

Figure 3. Routing in traditional networks.

In figure 3, Router-1 does not know how to reach networks C and D but it knows about networks A and B since they are directly connected to it. One way to teach Router-1 about these remote networks would be to manually configure the networks and the exit interfaces to reach them. The other method is for each router to exchange information regarding networks they know with each other.

When the dynamic method is used, Router-3 sends information about Network-D to Router-2. In the same manner Router-1 sends information regarding Network-A to Router-2. At this point Router-2 knows all networks in figure 3. However hosts on Network-A and D cannot reach each other. When Router-2 sends information regarding all networks it knows to Routers A and D, the network is said to be reached a converged state. Regardless of the number of routers between two given networks, using dynamic routing protocols it is possible to learn about remote networks. This makes them scalable and more desirable.

Static routes are used when the bandwidth over a link is too slow to be used for dynamic routing protocols as the update messages consume the bandwidth. Another situation is when a network has only one entry and exit point, also known as stub networks. The other reason for using a static route could be the ability of the router used to handle dynamic routing protocols. If it does not have the required CPU and memory resources to run dynamic routing protocols, static routing is recommended. However manual configuration of remote networks does not scale up since it is almost impossible to know all networks beyond the local network. If a path to a destination network fails, then a manual intervention is required. [16,22]

Routing in WSN

The data collected by the sensor nodes needs to be transferred to the application for which the WSN is deployed. Due to the nature of the environment they are functioning in and the energy limitations inherent in WSNs, an efficient routing protocol is required. [10,197] The frequency and number of transmissions to the central node impacts the energy consumption of the whole system. Aggregating the data collected from multiple sensor nodes and transferring the data to a sink, a node interfacing the WSN to other systems, at intervals or after a given threshold value is required. [10,199]

The distance of sensor nodes from the sink may vary significantly. Since the basic functionality of these nodes is sensing the environment and sending the data to the sink, there must be a way to transfer the data regardless of the nodes' proximity to the sink. Some of these nodes are close to the sink, some are at a considerable distance, and some even moving. There are different approaches to achieve the transfer of data from each node to the sink.

The simplest approach is for each node to send the data to the sink directly. Though this works for nodes close to the sink, the energy consumption of distant nodes and the interference levels of signals from these nodes are increased significantly. [10,199]

The problem with each node sending the data directly to the sink can be solved by using a multihop approach. The data reaches the sink via intermediate nodes. The reduced distance results in energy saving as well as reduction in interference. The frequency of transmission can be minimized by aggregating data and transmitting it when triggered or when a given threshold value is reached. [10,199]
The challenge in the multihop approach is choosing which intermediate node to use. The density of nodes used in WSNs does not allow broadcasting the data as the duplicated data will consume resources significantly. Only some of these nodes should be selected, and to this end routing algorithms that address the nature and limitations of WSNs should be used. [10,199] Figure 4 shows only specific nodes being used for data transfer.

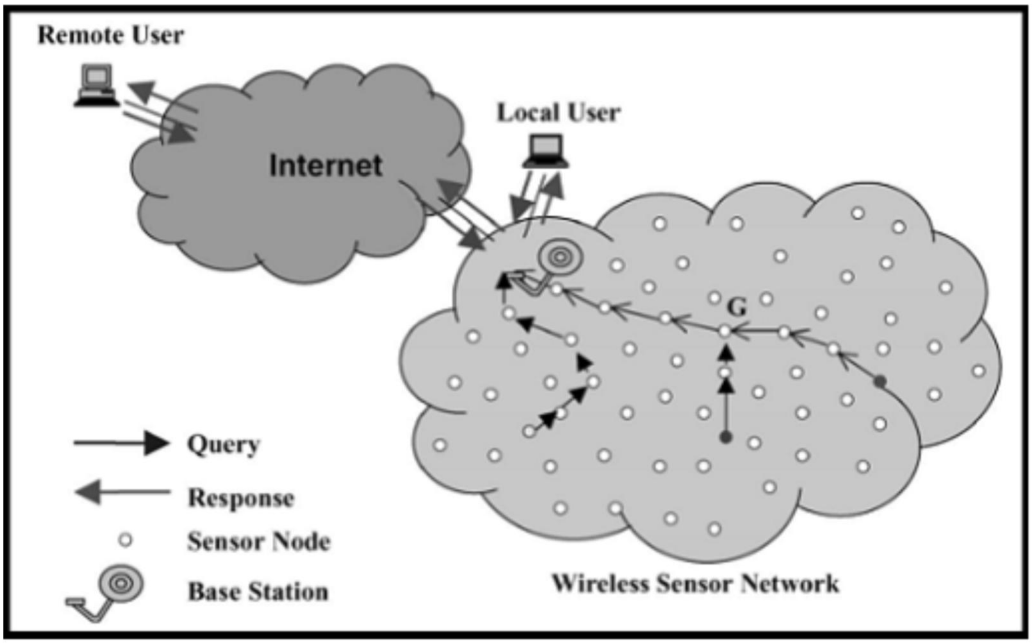

Figure 4. Multihop data and query forwarding.
Reprinted from Sohraby, Minoli, Znati (2007) [10,200]

Sensor nodes in figure 4 collect data from the environment and send it to the base station. Nodes do not send the data to every node in their vicinity and only certain nodes are selected along the path between the node and the base station. The intermediate nodes may aggregate data until a query is made by the base station at which point they will forward the data.

Routing algorithms suitable for WSNs can be compared based on parameters critical for the functionality of the system. For example, the Low-Energy Adaptive Hierarchy (LEACH) routing algorithm takes energy consumption, network lifetime and data aggregation into consideration. To this end it groups nodes in a set of clusters and chooses one of them as a head. [10,210]. Figure 5 shows nodes sending data to their cluster head.

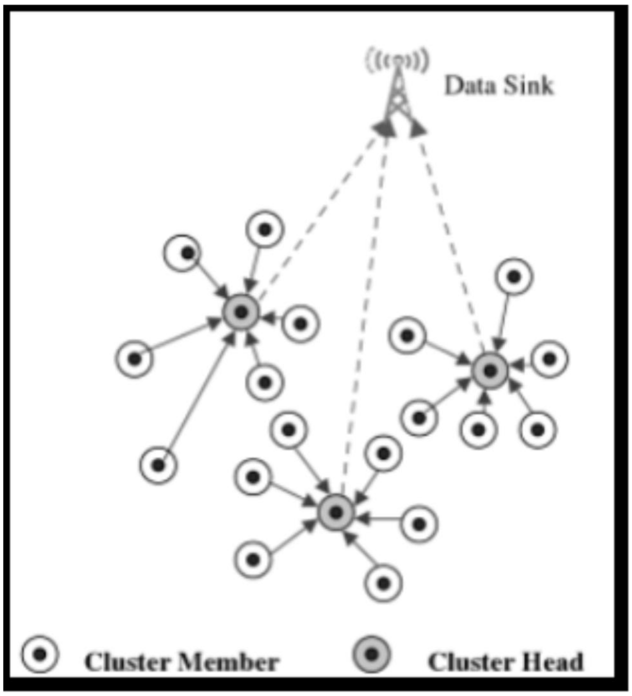

Figure 5. Leach network model.

Reprinted from Sohraby, Minoli, Znati (2007) [10,210]

What makes the arrangement in figure 5 different is that only the cluster heads are serving as intermediate nodes. The cluster heads may be grouped together to form other clusters of nodes from which a head is chosen in a similar manner. The process continues until the sink is reached directly by the heads which creates a hierarchical network structure. In all parts of the hierarchy, nodes other than the heads request to join the cluster. [10,212]

3 Components of the navigation system

This project was part of an indoor navigation system. The main project has Core, user device, Wireless sensor network and Mapping parts as well as an external system interfaced to the Core system. Due to time limitations the focus of this project was on the Mapping part. However to give a bigger picture other parts and the relationship between them is also discussed. In this paper a person means the user who holds the user device and, when referring to a particular device its identification number or name will be used. Figure 6 shows the above mentioned parts and their interconnections.

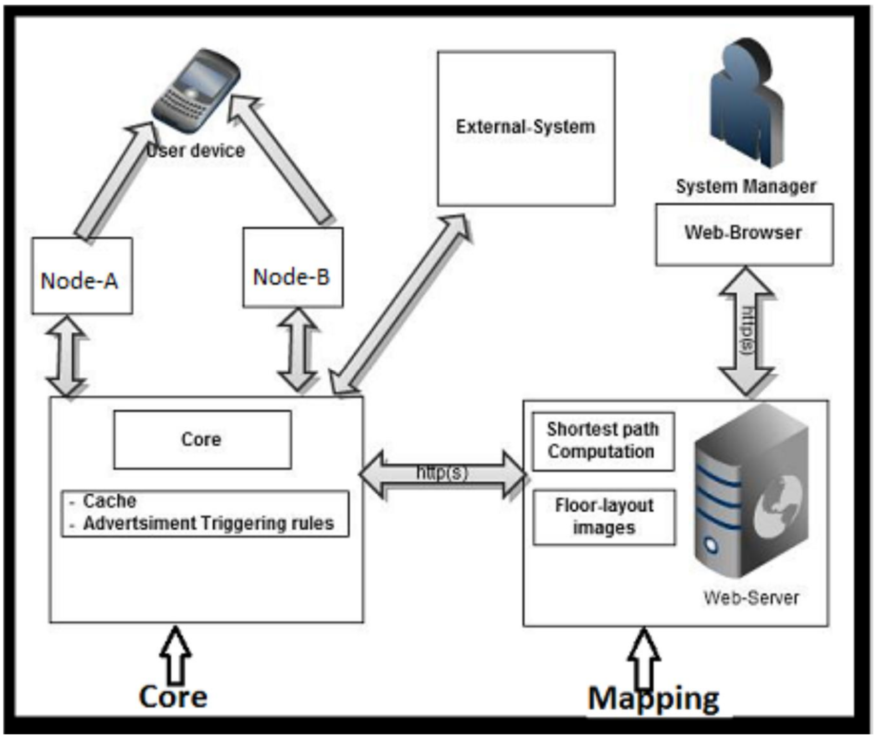

Figure 6. Components of the navigation system.

As figure 6 shows the main parts of the system are implemented independently and are run on different machines enabling the management and maintenance of each part to be done separately. Sections 3.1, 3.2 and 3.3 discuss these parts briefly.

3.1 Wireless nodes

Wireless nodes are placed on locations inside a building, so that the maximum possible area is covered by the network. These nodes are identified using unique numbers. These numbers are used for all computations and internal communications. However an appropriate name reflecting the locations of the node can be used and the person managing the system may refer to it with that name. However users should use only numbers.

The core part, which is connected to a central node, is aware of the distance between every node which makes drawing the topology of the network possible. The core part also contains a cache memory for storing the shortest paths received from the Mapping part and uses these values without the need to request the same information repeatedly. It also has rules for triggering advertisements sent to the person as she reaches different locations. These rules and other parts of the Core part are beyond the scope of this paper and are not discussed further.

The Core part is also connected to an external system which can be a service provider (SP) in a shopping mall, in an airport, in a train station, in a warehouse or any place where indoor navigation is required. The SP provides a portable user device for each person requesting its services and maintains a database to bind the person's information with the device's identification number or name. It also decides where the person's destination should be depending the service requested. Therefore, the main objective of the whole project is guiding the person to reach a destination chosen by the SP making it usable by multiple SPs at the same time.

3.2 User device

The portable device held by the person has a very limited memory and its display is of poor quality. Tracking its exact location and lowering its power consumption cannot be achieved simultaneously. The compromise is to exchange hello signals between a node and the device every 10 seconds. Should the person be expected to move faster than average, the hello interval should be made to be lower accordingly.

Since the display is not capable of showing map images, an alternative method is used. The user device gets instructions from the Core part via the wireless nodes. The user device can have connections to multiple nodes simultaneously making the instruction it receives unpredictable. Contradictory instruction may arrive or the same instruction may arrive at different instances of time from different nodes or even from the same node after reflection.

To avoid such problems only one node is chosen as a primary node for a particular person. When the person moves away from the chosen primary node, the signal strength is weakened. After reaching a given threshold value of the signal, a different node should be chosen as a primary node, a handover (HO) takes place. The number of HOs depends on the area covered by each node. The bigger the area covered by each node, the less HOs take place which simplifies the operation of the system.

When the Core part gets the destination from the SP and the path to be followed from the Mapping application, it sends this information to the person after which it keeps checking the person's movement. If the person is following the path as recommended, then there is no need for further communication except in the end to inform them that they have reached their intended destination. However, if the person moves away from the intended path, the Core will get the person's new location and will find the shortest path by asking the Mapping application or from its cache memory if a path for the current location and the original destination was asked before.

It is assumed that places inside the building reachable by the person are covered by the network, which eliminates the possibility of the person being lost from the tracking system. Therefore, the person will always be receive the right path for as long as the device is not turned off or the person has not left the building, in which case this system is not usable anymore.

3.3 Mapping system management

The locations of the wireless nodes used are shown on a web browser with which the person managing the Mapping system configures different parameters. There are two methods to achieve this goal. One is to construct the topology of the network using the known distances and some basic trigonometric functions. However, the actual distance between two nodes may not reflect the cost as the different paths having the same distance may take a different amount of time. The second method is to use manual configuration of paths and their corresponding costs.

The Mapping system manager defines the required paths for each service the SP provides and uploads floor layout images of areas for which the navigation system is implemented. Placement of nodes on the floor layout image depicts the actual locations of the nodes on the floor. These locations are assumed to be permanent, and changes are made whenever there are changes on the location of nodes on the ground.

The Mapping system manager uses a web interface to the Mapping application to add a new node, to relocate a node whenever changes are made, to remove a node when it is no longer needed or used, and to change the cost of the path between two given nodes. It is also possible to see the shortest path between two given points on the browser which is helpful for testing purposes. Maintaining the database used for this application and backing up the shortest paths computed to a file are also the tasks of the Mapping system manager. Each of the above mentioned tasks are discussed in chapter 4.

The Mapping application was tested on a 32-bit Windows system running Windows 7 professional with installed memory of 2.00 GB. Since making the application independent of any algorithm used to find optimum routes, cost values are stored into database. As it is a PHP-MySQL Web database application, configuration and set ups are done using a web browser. It works well on both Firefox and Internet explorer.

4 Mapping

4.1 Maps

Maps are graphic representation of the real world to a limited extent. Since the details in the cultural or physical world are too complex to be drawn on a two dimensional maps, some distortion occur. [20] One source of distortions is size. Maps are smaller than what they represent and this can be shown using scales as a ratio. Other sources of distortion are projection and symbolization. [21,5]

For this project, two alternatives creating maps that show the distribution of wireless nodes and other parameters of interest were tried. One was using MapServer, an open source GIS development environment, and the other using GD Graphics Library which requires less effort. The latter was chosen for the final implementation.

Layers

Depending on the application different layers can be shown on top of a base image. These layers may show country boundaries, airports, cities, road networks or water bodies. Thought it is possible to prepare layers to show the distribution of entities of interest, using readymade ones is the most efficient way of developing applications.
Different organizations prepare layers for public use. For example, The National Atlas of the United States of America prepares layers in different categories such as agriculture, Biology, Climate, Water and Transportation, among others. [21]

When a distribution of an entity of interest over a given area is needed to be shown on a map, a layer is prepared if a ready-made does not exist. There are different tools that can be used for this purpose. OpenEV is one these tools and was used for creating layers that show distribution of some parameters on top of a base image, show in figure 7. The layers could be made of point, line or polygon types. [19]

Without going into details of creating dynamic maps, figures 7 to 12 show how different layers are used to give meaningful information about a given area of interest. Figure 1 shows a floor's layout image inside a building for which the distribution of entities of interest is to be shown on.

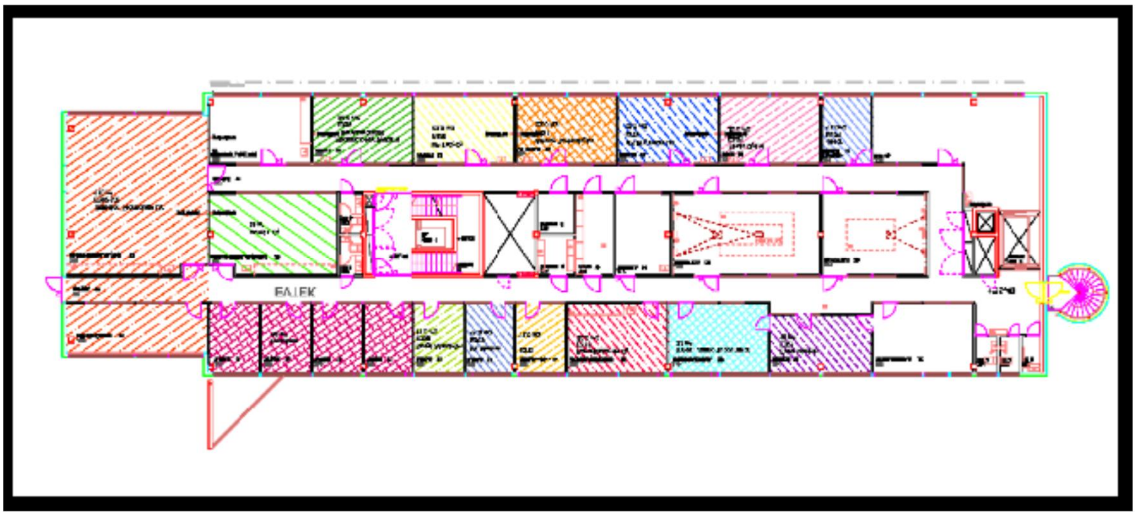

Figure 7. Base image for creating a map

There rooms, corridors, lifts and other entities in figure 7 are represented by standard symbols. However, the image itself cannot be used for creating a map for two reasons. One reason is that many people do not know these symbols and they do not have an appealing feature to be used for mapping purposes.

The other reason is that when some entities or locations are not to be shown for security reason. Showing safe houses or server rooms on the map may not be a good idea. In real world these entities could be military camps and nuclear plants. Showing these areas on a map may not be allowed by the government. The alternative method is to use an image that shows only required entities and hiding the rest as shown in figure 8.

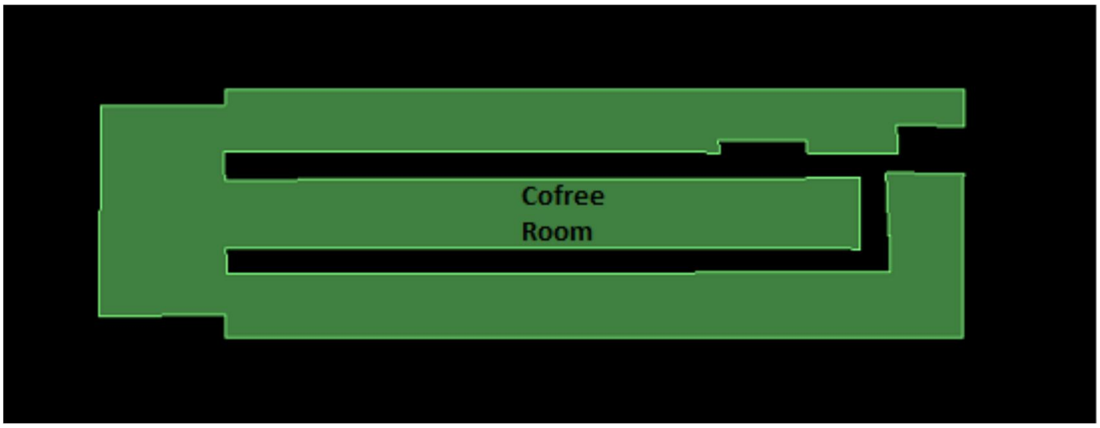

Figure 8. Hiding details of an area.

The details in figure 1 are not shown in figure 8. If, for example, only the coffee room is to be shown, that information is displayed on the appropriate location. Figure 9 shows the boundary lines between adjacent rooms and other entities leaving an empty space for the doors. It is also possible to prepare another layer showing only the doors in a shape chosen to represent a door.

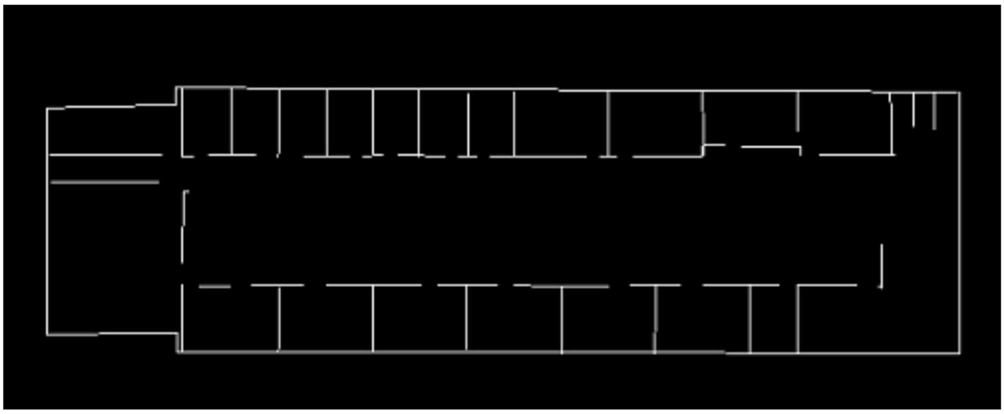

Figure 9. Boundary lines.

The boundary lines in figure 9 give a clear distinction between adjacent rooms and other entities. Without these lines the extent of a room, for example, cannot be known. The corridors can be shown as a walking path in a similar manner as in figure 10.

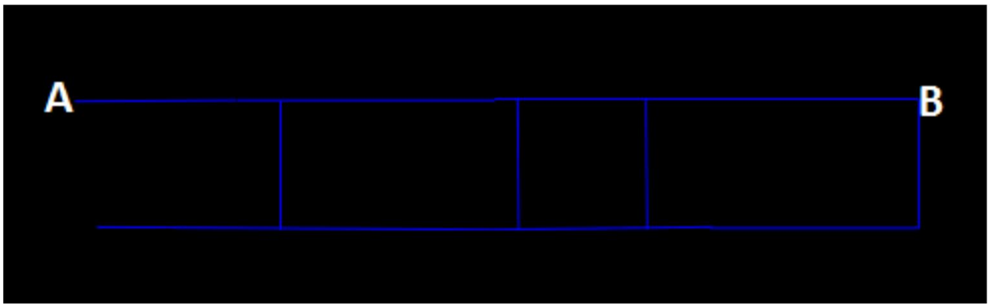

Figure 10. walking paths.

The paths shown in figure 10 drawn once but can be used with different spans. If only the path between point A and point B needs to be shown, then the whole layer is selected but only the part between these points is shown with a different color than the background on which the layer is added.

The layers created above do not change so often as does the layer showing employees in each office as shown in figure 11. An employee might change her office, leave the company making the office empty or a new employee might get hired. The information regarding each employee in an office can be made part of the layer dynamically.

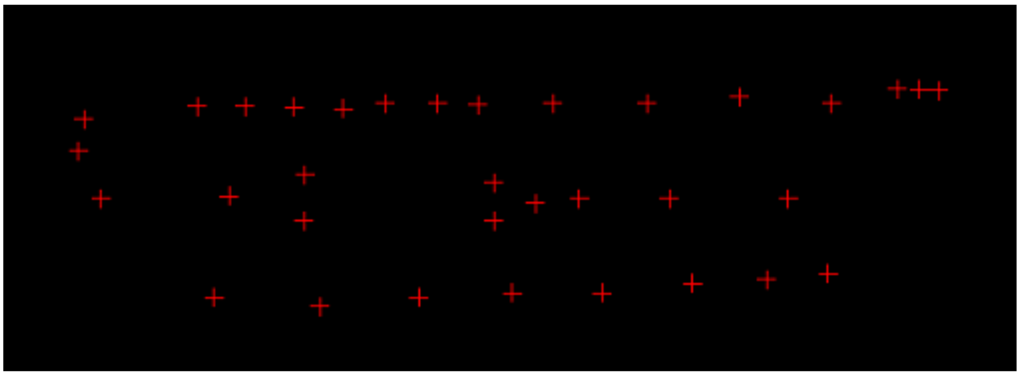

Figure 11. Employees in each office.

The cross signs in figure 11 can be changed with names of employees or images, if the map created is big enough to hold images. The information regarding the occupants of the offices should be updated and need to be shown on the layers relaying on this information.

Similar to the layers above, more layers can be created to show the distribution of parameters of interest. One or more layers are shown on top of a base image to give meaningful information. Since details of the floor plan image of figure 7 are required to be hidden, figure 11 can server as a pseudo base image. Figure 12 shows all layers created above drawn together.

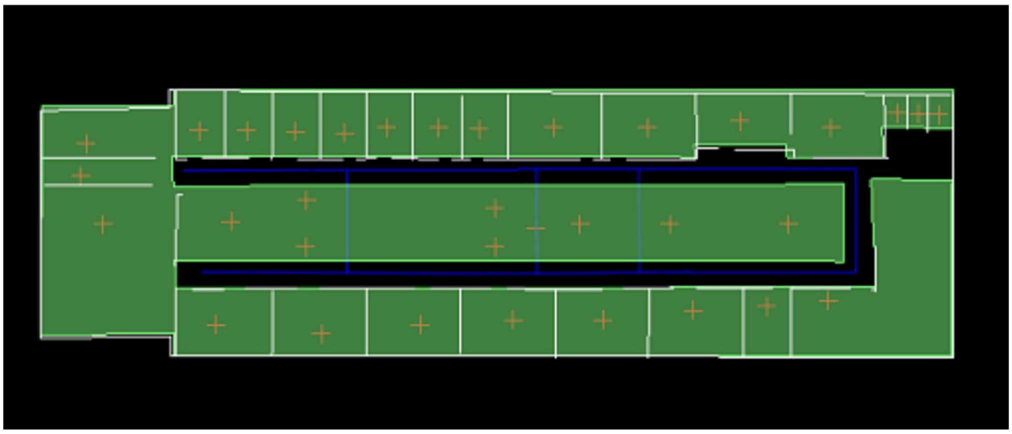

Figure 12. Layer combined to give meaningful information.

All layers in figure 12 are not necessarily shown at once. Depending on the requirement only few of the layers can be shown. Even when showing a layer, every part of it may not be shown as discussed on the path layer case. If the distribution of the parameters of interest does not change as with rooms, lifts or any physical structure inside a building then using layers to create a map is desirable and scalable.

Using the methods above, it is possible to create a layer that shows the distribution of wireless nodes used in this application. However, the location of the nodes change from time to time which requires creating a layer that reflects the change. Creating layers to show all possible arrangements of nodes is not practical as the number of layers increase much more than the increase in the number of nodes.

The alternative used for creating the Mapping system does not require the creation of multiple layers. The floor's plan image for a building is used as a base and the distribution of nodes is drawn manually using the GD library. This implementation makes the management of the system much simpler and does not require sophisticated knowledge. The details of the system implementation and configuration are discussed in section 4.3.

4.2 Using the GD Graphics Library

The GD Graphics Library is used for creating and manipulating images on the fly. Depending on the version of GD installed handling different images formats such as gif, jpeg, png and xpm is possible. The Freetype 1.x, FreeType 2 and T1lib are font libraries used for enhancing the GD.[1]Enabling GD-support requires configuring PHP. The details on the requirements and installations can be found on the PHP manual page.

This section provides a brief description of Image functions used for developing this application. The full list of Image functions can be found in appendix 1. Figure 13 shows two points connected by a line. The points represent wireless nodes used for navigation purpose while the line represents the path that exist between two given nodes.

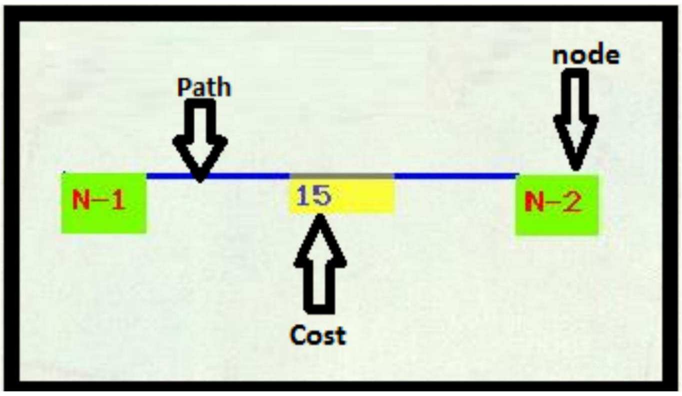

Figure 13. Mapping building blocks.

The function *'resource imagecreate(int $width, int $height)'* ' creates a blank image of height and width specified by the parameters. 'resource' is an image identifier representing the created image. The two green rectangles and the yellow one drawn mid-way between them need a text to be written on top of them. The background color is also changeable.

The background image for the rectangles created by can be changed using the function '*imagecolorallocate (image , red, green , blue)' .* The 'image' parameter refers to the identifier discussed above while the red, green and blue parameters contain integer values of their corresponding color. The function returns a color represented by the color values. In a similar manner the color of the text to be written on top of these rectangles is set.

The function 'imagestring (im , font , x , y , text , color)' on the other hand is used to draw a string horizontally at the given coordinates by indicated by x and y. The Image identifier on which the string is drawn, the font size and its color are also passed as parameters. The function returns true on success.

The function 'imagefilledpolygon (image , points , number_of_points ,color)' is used for creating a filled polygon. The 'image' is an identifier on which the polygon is to be drawn. 'points' is array containing the coordinates of the polygons, and a minimum of three such points are required. The line showing the shortest path between two given nodes is made to be thicker than ordinary paths is shown in a different color. The function takes color of the polygon as a parameter and return true on success.

'imagecopymerge(dst_im, src_im, dst_x, dst_y, src_x, src_y, wid, hei ,pct)' is used to Copy and merge part of an image and takes some parameters.. The 'dest_im' is an identifier for the floor plan image on which the wireless nodes are shown. The 'src_im' is an identifier for the image created to show the nodes. The coordinates for the source and destination image are also passed as parameters. The same format is used to draw the paths between nodes on the same destination image. The function returns true on success.

The above discussed functions take the coordinates of a node and its cost from other nodes as parameters. There are utility functions that provide coordinate and cost values that are saved to a database. When the coordinates of a node or its cost are changed, the utility functions update the database accordingly. When a node is added or removed using a web browser these functions add and remove entries in the database respectively.

The reason for saving the coordinates and cost values into a database instead of other data structures is to make the application independent of any particular programming language or a particular algorithm used for finding the shortest path between two given nodes. Though, in this application, the Dijkstra algorithm is used to find the shortest path between two given nodes, any other algorithm can also be used. The coordinates and cost values from the database can be copied to the data structures used by the algorithm.

There are more functions to the above list. There are also a considerable number of utility functions used when developing the Mapping application. Discussing the details of these implementations defies the idea of making this application generic, independent of any particular programming languages or shortest path algorithms. However, section 4.5 show the use interface for managing the Mapping system and the output required by the core system to guide users.

4.3 Configuration

Sections 4.2.1 to 4.2.7 describe the steps in changing parameter regarding a wireless node used for the navigation purpose. These parameters include the location of a node and its cost from other nodes. The cost takes both distance and traveling time into consideration. Removing and relocating a node is also shown.

4.3.1 Adding a node

By default there are 10 wireless nodes that are not visible on a web browser. The drop- down list in figure 14 shows these ten nodes plus one additional node with an identification number that is one greater than the maximum available number of nodes, 11 at beginning. When a node is chosen and added, a corresponding entry is added to the POSITIONS table. The difference between adding and removing a node is that adding is an endless process while removing ceases when the number of available nodes is reached. Figure 14 also shows a situation in which all existing 13 nodes are added and the drop-down list shows only the next node to be added, which in this case is node number 14.

Figure 14. Adding a new node.

To complete the addition process in figure 14, the new node's location should be chosen by clicking in the floor layout diagram. The location's (x,y) coordinate values are the X and Y values of the POSITIONS table entry for the new node.

The last step in the addition process is adding an entry to the CONFIG table. As was described in section 4.3.6, an additional row and column is added. Taking figure 14 as an example, row 14 and column 14 are added. All values of the new entry are set to the maximum integer value the local machine can hold. This makes the new added node to be at infinity from each of the already existing nodes, and no path is drawn.

4.3.2 Relocating a node

To change the location of a wireless node selecting it from the drop-down list as shown in figure 15 is required. The drop-down list shows only wireless nodes that are visible regardless of whether they are connected with other nodes or not.

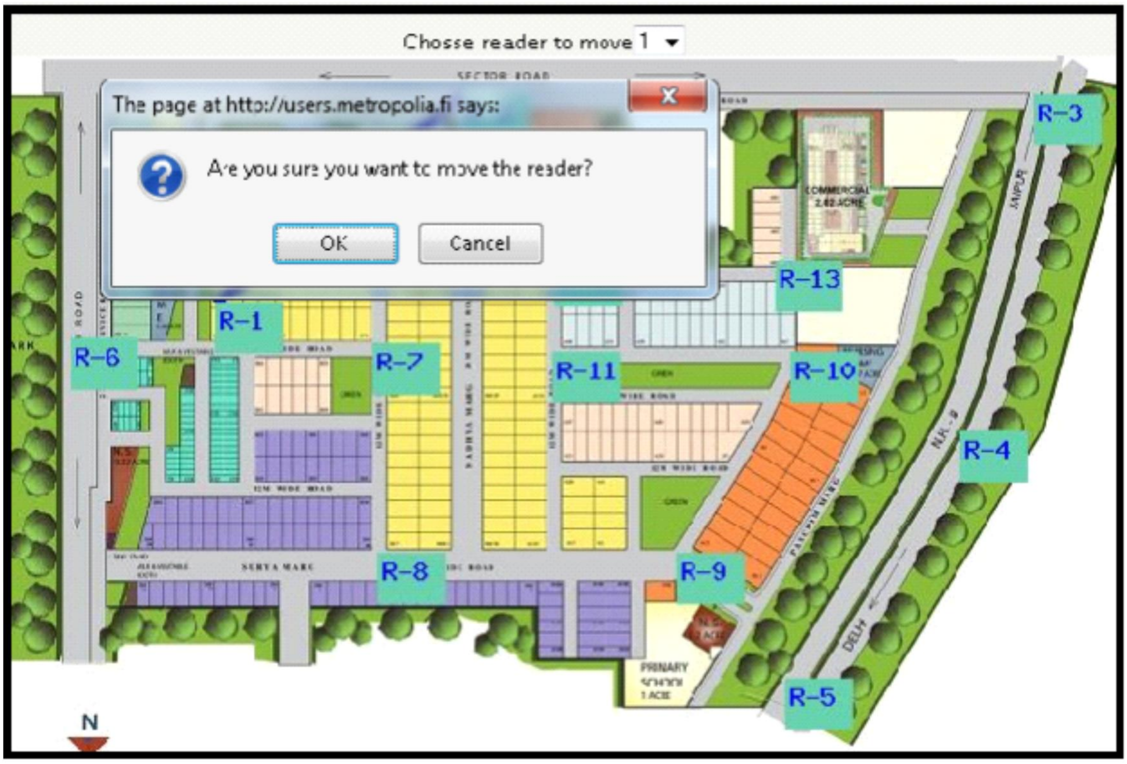

Figure 15. Relocating an existing node.

As shown in figure 15 relocation requires a confirmation before committing the changes. The new location should reflect the actual arrangement on the ground. The POSITION table is updated to show the new position of the node. If what is on the Mapping system does not reflect what is on the ground, then a non-optimal route is selected.

4.3.3 Removing a node

A node can be removed without affecting the functionality of the Mapping system. As show in figure 16, the node to be removed is chosen from the drop-down list. The list shows all nodes that are visible, which could be connected to other nodes or exist as a standalone node.

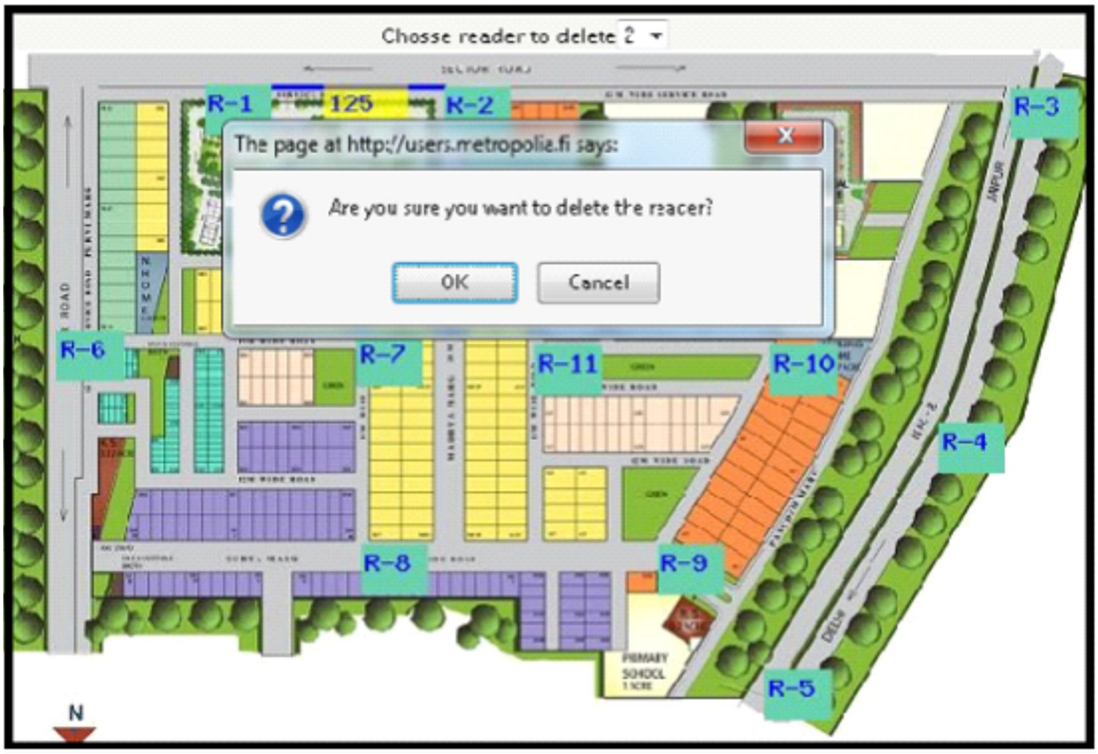

Figure 16. Removing a node.

As shown in figure 16, removing also requires a confirmation. Upon getting the confirmation the following actions take place. The node's entry on the POSITIONS table is removed completely making it invisible on the browser. If the node has any connection with other nodes, these connections are broken, and it is made to be a standalone node by updating the CONFIG table. After that the node is considered to be at infinity with other existing nodes.

After a node is removed, as mentioned above, it is not possible to see it on the drop-down list. Only those that are not deleted are shown. If and when all nodes are removed from the Mapping system, then the drop-down list does not show anything and POSITION table is emptied.

4.3.4 Changing the cost

There are two ways of configuring the cost between two wireless nodes. One option is to make the wireless nodes measure the distance between each other and take the result as a cost. Though this is the easiest way, it has its limitations. When two wireless nodes are on the opposite side of a barrier, the distance between them does not reflect the actual path the person follows. Another situation is when the path the person follows is expected to take more time than is expected for a given distance. This could happen around cashier or security check points. In these cases manually setting the appropriate cost for each path is important, and the shortest path is calculated taking these values as an input.

Figure 17 shows how to set the cost between two nodes. The nodes can be selected from the drop-down list and the cost can be either (-1) or any positive value less the maximum integer number the local machine can hold.

Figure 17. Changing cost between wireless nodes.

As figure 17 shows, when a new node is added, it is assumed to be at infinity with any of the already existing nodes. This state can be changed by providing an appropriate value. After choosing two nodes, if (-1) is provided as a cost, the two nodes are set to be at infinity from each other and no path is drawn between them. When a positive number less than or equal to the maximum integer value the local machine can hold is provided, a path is drawn between the two nodes and the corresponding CONFIG table entry is updated.

For N number of nodes there exists a table with N columns and N rows depicting an NxN matrix. The distance of a node with itself is always considered to be zero. If a different value is provided, the input is ignored and all entries of the diagonal matrix are set to zero. Figure 18 show when the cost between nodes 1 and 2 is set to 125.

Figure 18. Changed cost between two nodes.

As seen in figure 18, a path is drawn between nodes 1 and 2 and the cost is shown at midpoint between them with a different background color. In a similar manner, the rest of the nodes are connected as necessary. If a node stops functioning or the current cost does not reflect what is on the ground, then its cost from all other nodes should be changed to reflect these situations.

4.3.5 Displaying the shortest path

In finding the shortest path among multiple routes, each route's cost is taken into consideration. Cost in this particular application refers to distance. Figure 19 shows how the shortest path is selected among multiple routes.

Figure 19. Choosing the shortest path.

As shown in figure 19, to reach node 6 there are two alternatives. One route is via node 2, node 3, node 4 and node 5 with a total distance of 23 units. The other option is to take the direct path whose cost is 25. The shortest path is drawn with the deep blue color while the alternatives are drawn with the light blue color. Should the two alternatives tie, the number of nodes will be taken into consideration to break the tie. In figure 19, had the distance between nodes 5 and 6 been set to 3, the direct path from node 1 node 6 would have been chosen as it has less number of nodes despite both routes having the same cost.

As can be seen in figure 19, the cost set between two points may not reflect the actual distance on the ground. Some paths can be crowded or there might be a queue. The direct path between nodes 1 and 6 represents such a case. Since the same path may server a different number of people at different times of the day, the cost should always be adjusted to reflect the traffic.

4.3.6 Cost and node location data

All information relevant for this application is stored in a database. As shown in Figure 20, the location of every node is saved in the POSITIONS table. The distance of a node from every other node is also saved in the CONFIG table.

Node	x	y
1	180	31
6	180	334
3	601	239
4	290	239
2	599	34
5	289	331

Node	to_r1	to_r2	to_r3
1	0	10	78
2	10	0	77
3	78	77	0

Figure 20. a) POSTIONS table b) CONFIG table

The nodes column in figure 15 a) stands for each node that is visible on the browser while the x and y values represent that particular node's X and Y coordinates on the browser, respectively. The reference point for all nodes' positions is the left top corner of the floor layout image used as a base image on the browser.

The node column in figure 15 b) represents each node while the rest of the columns stand for the distance of the node from the other nodes. Taking the first row as an example, the following points can be observed. Number 1 stands for node 1, to_r1 stands for node 1’s distance from itself. The distance of a node from itself is always considered to be zero. The column to_r2 stands for Node 1's distance from node 2 which can be any positive value less than the maximum integer value the local machine can hold. In a similar manner to_r3, to_r4... to_r_n are Node 1's distances from all nodes starting from the third node to the n^{th} one. This results in an NxN matrix where N is the number of nodes available. These entries are taken as an input to the shortest path calculation. However sorting the table as shown in figure 20 is not required.

The distance between two given nodes is assumed to be the same. If the distance between node 1 and node 2 is 100, so is the distance between node 2 and node 1. For this reason, half of the entries in CONFIG table can be ignored.

4.3.7 Shortest paths

Since the application is for guiding customers of a SP, the person should be able to get the shortest path information using the portable user device she is given. The device has a small memory size and can display only texts and numbers. For this reason, the information sent to the person should be in the simplest possible format.

There are numbered signs inside the building for which this project is being implemented. Figure 21 shows numbered signs in the Helsinki Railway Station. These signs are already deployed for other purposes. By just putting the wireless nodes next to them they can be made part of the navigation system.

Figure 21. Signs for navigation

In places where such signs do not exist, new ones are added and each node is numbered the same as the signs next to them. If, for example, the person is required to go from point 4 to 10 and must pass through points 6 and 9, then 4-6-9-10 is sent from the core and is shown on the display. The user then follows the path using the signs shown in figure 21 as reference points. The usage of numbers instead of names makes the application independent of any specific language and usable by any person.

If the person is not following what was sent earlier, then a new calculation is made according the person's new position. To reduce power consumption, the communication between the person's handheld device and wireless nodes is done once in every 10 seconds. If the person moves too fast to be tracked, then the computed paths will not be useful.

This Mapping system interacts with the core system which handles the communication with the device held by the shopper. The main task of this application is to respond to the core system's request for the shortest path between the person's current location and the destination asked by the SP.

The core system can get his information in three ways. The first option is to use a TCP connection to request the information. This application listens on TCP port 10,000. If there is another application using the same port, then a different one can be chosen. The second option is using pre-computed values as shown in figure 22.

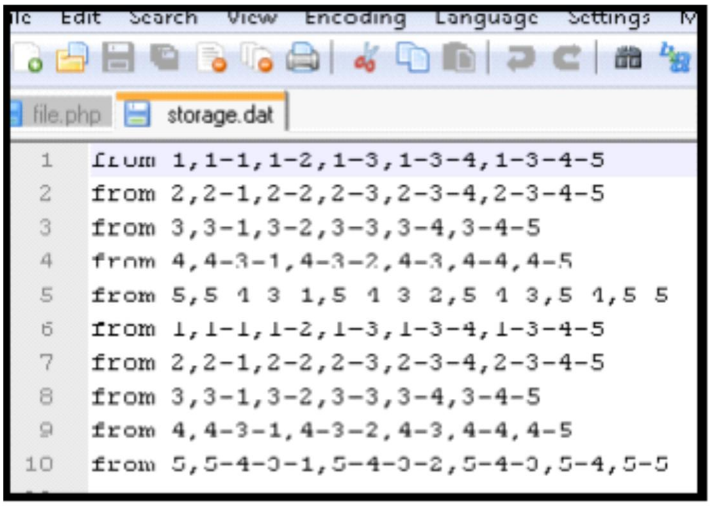

Figure 22. Pre-computed values saved to a file.

In figure 22 all shortest paths from every node to every other node are selected and saved in the “storage.dat” file. For example, from node 4 one should take path 4-3 to reach point 1 or point 2. In a similar way, the shortest path from each node to every other node is saved and there is no need for making calculations, unless there is a change to the arrangement of wireless nodes on the ground. If so, appropriate changes should be made, as discussed earlier. The last option is to save these results in a database.

The three options discussed above give redundancy and make the system reliable and predictable. The last two options discussed here do not require a direct communication with this Mapping system. The results of computation are saved to a file or a database and the core system is getting these values and sends this information to the person if required.

5 Conclusion

The objective of this project was developing an application for managing and configuring the placement of wireless sensor nodes deployed inside a building, and finding an optimum route between two points of interest inside the building for which the navigation system is deployed. It also aimed to develop a mechanism to show the distribution of the nodes and the paths used for traveling inside the building.

The key outcome of this project was production of the mapping application for configuring and controlling wireless nodes deployed inside a building for which the navigation system was implemented. The distribution of these nodes and the paths customers should follow were shown on a web browser. A system manager for the mapping application was able to add new nodes, to remove and relocate them. The manager was also able to change the cost between two given nodes taking traveling time into consideration.

Information regarding the optimum paths was sent to the core system, which would send it to the person holding the device provided using the service provider. To avoid re-computations, calculated paths were saved to a database as well as to a backup file. However, when the source, the destination or both points were not known, calculation to get the optimum route was done.

The limitation on the handheld user device used has limited the output of the system to be of only simple formats, and showing graphical maps was not possible. The limitation on power source has also made communication between the core system and the user device at an interval not suitable to show the persons current location in real time.

This project can be enhanced using better techniques to create dynamic maps that reflect changes on the ground in real. Using user devices capable of displaying images, it would be possible to convey graphical information to the users requesting the navigation service.

References

1. Jun Zheng, Abbas Jamalipour: Wirelss Sendor Networks: A networking perspective. Canada: John Wiley & Sons; 2009.

2. Jon S. Wilson: Sensor Technology Handbook. Canada: Elsevier Inc; 2005.

3. Stephen Beeby, Neil White: Energy Harvesting for Autonomous Systems. MA: Artech House; 2010.

4. Terrance J. Dishongn, Michael McGrath: Wireless sensor networks for healthcare applications. MA: Artech House; 2010.

5. Craig Wright: The IT Regulatory and Standards Compliance Handbook: How to survive an information systems audit and assessments. MA: Syngress Publishing, Inc., Elsevier, Inc.; 2008.

6. Alber S. Huang, Larry Rudolph: Bluetooth: Essentials for programmers. NY: Cambridge University Press; 2007.

7. Amin Rida, Li Yang, Manos Tentzeris: RFID-Enabled Sensor Design and Applications. MA: Artech House; 2010.

8. Wireless sensor network. [Online]. URL: http://en.wikipedia.org/wiki/Wireless_sensor_network. Accessed 1 May 2011.

9. Feng Zhao, Leonidas Guibas: Wireless sensor networks: an information processing approach. San Francisco: Morgan Kaufmann Publishers; 2004.

10. Kazem Sohraby, Daniel Minoli, Taieb Znati: Wireless sensor Networks: Technology, Protocols, and Applications. New Jersey: Jhon Wiley and Sons, Inc.; 2007.

11. Stephen W. Hinch: Outdoor Navigation with GPS. CA: Wilderness Press; 2007.

12. G S Rao: Global Navigation satellite systems: With essentials of satellite communications. New Delhi: Tata McGraw-Hill; 2010.

13. Holger Karl, Andreas Willig: Protocols and Architectures for Wireless Sensor Networks. England: John Wiley & Sons, Ltd; 2005.

14. Net Objectives. The Essential Ingredients of a Navigation System. [Online]. URL: http://www.netobjectives.com/blogs/essential-ingredients-of-navigation-system. Accessed 24 April 2011.

15. Andrew S. Tanenbaum: Computer Networks. New Jersey: Prentice Hall PTR; 2003.

16. Cisco Systems, Inc.: Implementing Cisco IP Routing (ROUTE): Foundation learning guide. Indianapolis: Cisco Press; 2010.

17. Nurie. Undomestic Goddess Tales. [Online].
URL: http://nuriespace.blogspot.com/2010/05/as-dry-as-sahara-desert.html.
Accessed 8 May 2011.

18. Boat Owners of the United States. Cruising logs. [Online].
URL: http://www.boatus.com/cruising/littlegidding/200310-22.asp.
Accessed 8 May 2011.

19. OpenEV Online Help Pages. [online]. URL:
http://openev.sourceforge.net/app/openevmain.html.
Accessed 17 May 2011.

20. Maps, Remote Sensing, and GIS . [online]. URL:
http://www.physicalgeography.net/fundamentals/2a.html.
Accessed 17 May 2011.

21. National Atlas of the United States: Map Layers . [online]. URL:
http://www.nationalatlas.gov/maplayers.html.
Accessed 17 May 2011.

22. Mark Monmonier: How to lie with maps. England: The University of Chicago press; 1996.

23. GD and image functions: PHP manual page. [online].
URL: http://php.net/manual/en/ref.image.php.
Accessed 17 May 2011.

Appendices

Appendix 1: GD and image functions

gd_info — Retrieve information about the currently installed GD library
getimagesize — Get the size of an image
image_type_to_extension — Get file extension for image type
image_type_to_mime_type — Get Mime-Type for image-type returned by getimagesize, exif_read_data, exif_thumbnail, exif_imagetype
image2wbmp — Output image to browser or file
imagealphablending — Set the blending mode for an image
imageantialias — Should antialias functions be used or not
imagearc — Draws an arc
imagechar — Draw a character horizontally
imagecharup — Draw a character vertically
imagecolorallocate — Allocate a color for an image
imagecolorallocatealpha — Allocate a color for an image
imagecolorat — Get the index of the color of a pixel
imagecolorclosest — Get the index of the closest color to the specified color
imagecolorclosestalpha — Get the index of the closest color to the specified color + alpha
imagecolorclosesthwb — Get the index of the color which has the hue, white and blackness
imagecolordeallocate — De-allocate a color for an image
imagecolorexact — Get the index of the specified color
imagecolorexactalpha — Get the index of the specified color + alpha
imagecolormatch — Makes the colors of the palette version of an image more closely match the true color version
imagecolorresolve — Get the index of the specified color or its closest possible alternative

imagecolorresolvealpha — Get the index of the specified color + alpha or its closest possible alternative
imagecolorset — Set the color for the specified palette index
imagecolorsforindex — Get the colors for an index
imagecolorstotal — Find out the number of colors in an image's palette
imagecolortransparent — Define a color as transparent
imageconvolution — Apply a 3x3 convolution matrix, using coefficient and offset
imagecopy — Copy part of an image
imagecopymerge — Copy and merge part of an image
imagecopymergegray — Copy and merge part of an image with gray scale
imagecopyresampled — Copy and resize part of an image with resampling
imagecopyresized — Copy and resize part of an image
imagecreate — Create a new palette based image
imagecreatefromgd2 — Create a new image from GD2 file or URL
imagecreatefromgd2part — Create a new image from a given part of GD2 file or URL
imagecreatefromgd — Create a new image from GD file or URL
imagecreatefromgif — Create a new image from file or URL
imagecreatefromjpeg — Create a new image from file or URL
imagecreatefrompng — Create a new image from file or URL
imagecreatefromstring — Create a new image from the image stream in the string
imagecreatefromwbmp — Create a new image from file or URL
imagecreatefromxbm — Create a new image from file or URL
imagecreatefromxpm — Create a new image from file or URL
imagecreatetruecolor — Create a new true color image
imagedashedline — Draw a dashed line
imagedestroy — Destroy an image
imageellipse — Draw an ellipse
imagefill — Flood fill
imagefilledarc — Draw a partial arc and fill it
imagefilledellipse — Draw a filled ellipse
imagefilledpolygon — Draw a filled polygon
imagefilledrectangle — Draw a filled rectangle

imagefilltoborder — Flood fill to specific color

imagefilter — Applies a filter to an image

imagefontheight — Get font height

imagefontwidth — Get font width

imageftbbox — Give the bounding box of a text using fonts via freetype2

imagefttext — Write text to the image using fonts using FreeType 2

imagegammacorrect — Apply a gamma correction to a GD image

imagegd2 — Output GD2 image to browser or file

imagegd — Output GD image to browser or file

imagegif — Output image to browser or file

imagegrabscreen — Captures the whole screen

imagegrabwindow — Captures a window

imageinterlace — Enable or disable interlace

imageistruecolor — Finds whether an image is a truecolor image

imagejpeg — Output image to browser or file

imagelayereffect — Set the alpha blending flag to use the bundled libgd layering effects

imageline — Draw a line

imageloadfont — Load a new font

imagepalettecopy — Copy the palette from one image to another

imagepng — Output a PNG image to either the browser or a file

imagepolygon — Draws a polygon

imagepsbbox — Give the bounding box of a text rectangle using PostScript Type1 fonts

imagepsencodefont — Change the character encoding vector of a font

imagepsextendfont — Extend or condense a font

imagepsfreefont — Free memory used by a PostScript Type 1 font

imagepsloadfont — Load a PostScript Type 1 font from file

imagepsslantfont — Slant a font

imagepstext — Draws a text over an image using PostScript Type1 fonts

imagerectangle — Draw a rectangle

imagerotate — Rotate an image with a given angle

imagesavealpha — Set the flag to save full alpha channel information (as opposed to single-color transparency) when saving PNG images
imagesetbrush — Set the brush image for line drawing
imagesetpixel — Set a single pixel
imagesetstyle — Set the style for line drawing
imagesetthickness — Set the thickness for line drawing
imagesettile — Set the tile image for filling
imagestring — Draw a string horizontally
imagestringup — Draw a string vertically
imagesx — Get image width
imagesy — Get image height
imagetruecolortopalette — Convert a true color image to a palette image
imagettfbbox — Give the bounding box of a text using TrueType fonts
imagettftext — Write text to the image using TrueType fonts
imagetypes — Return the image types supported by this PHP build
imagewbmp — Output image to browser or file
imagexbm — Output XBM image to browser or file
iptcembed — Embeds binary IPTC data into a JPEG image
iptcparse — Parse a binary IPTC block into single tags.
jpeg2wbmp — Convert JPEG image file to WBMP image file
png2wbmp — Convert PNG image file to WBMP image file

Reprinted from PHP Manual page (2011) [23]