A Context-Aware Mobility Indoor Positioning System

Eugene Ferry, Computing Department, Letterkenny Institute of Technology, Letterkenny, Ireland

John O'Raw, Computing Department, Letterkenny Institute of Technology, Letterkenny, Ireland

Kevin Curran, School of Computing and Intelligent Systems, Ulster University, Northern Ireland

ABSTRACT

The need for location based services has dramatically increased within the past few years, especially with the popularity and capability of mobile device such as smart phones and tablets. The limitation of GPS for indoor positioning has seen an increase of indoor positioning based on Wireless Local Area Network 802.11. The authors demonstrate here a real world application of determining one's location with the Cisco Context-Aware Mobility which provides a Real Time Location System solution based on Wi-Fi. They detail their implementation of an Android application which communicates with the Cisco Context-Aware Mobility system to visually display the location of the mobile device. The application was tested in a production environment and limitations in the production environment along with the diagnostic capabilities of the Context-Aware Mobility were identified. The authors found that to obtain optimal accuracy, a device must be detected by four or more Access points so a recommended distribution for an indoor positioning system built on the Cisco context-aware mobility framework is for an Access Point to be placed every 12 – 20 linear meters.

Keywords: *Context-Awareness, Indoor Positioning, Indoor Tracking, Mobile, Wireless Networks*

1. INTRODUCTION

Over the last few years the popularity of Real Time Location Systems (RTLS) has increased. This increase has created a demand for indoor positioning systems (Hossain et al., 2007). Wi-Fi based RTLS can be developed using a number of different techniques to measure Wi-Fi signals to determine one's location. The level of accuracy provided by these techniques can vary depending on the chosen environment. The availability of indoor positioning systems not only provides the location of the user and its surrounding environment but it also provides businesses with the opportunity to advertise or promote their business, their latest deals and much more. The majority of enterprises, universities and shopping centres deploy WLAN as a means of communications and services, hence the reason Wi-Fi is used in RTLS. The fact that most mobile devices now

DOI: 10.4018/IJARAS.2015010102

contain Wi-Fi radios also has a major factor in choosing Wi-Fi. The most popular positioning system used today is GPS, however GPS does not work well indoors as a clear line of sight to the satellites is not always attainable due to obstructions (Ching et al., 2010). This has led to the introduction of indoor positioning based on WLAN 802.11. According to (Hossain et al., 2007) indoor location systems should be cost-effective, involve a short training phase and provide a high level of accuracy. The ever increasing popularity of smartphones and tablets has also made these RTLS more accessible to the general public.

There have been many proposed solutions to better improve the accuracies of some of these techniques. One proposed solution involved applying different formula in order to account for multipath interference and so forth. Another allowed the user to inform the RTLS of the accuracy of its location positioning by stating if the proposed location is in fact correct or not. One of these solutions is Cisco's Context-Aware Mobility. The Context-Aware Mobility provides real time positioning by measuring Wi-Fi signal and incorporating location calculations to provide a high level of accuracy. Each solution has advantages and disadvantages. This paper looks to ascertain if the use of the Cisco Context-Aware Mobility is a viable solution. By deploying it on an Android device, this research will determine if the Cisco Context-Aware Mobility is capable of providing a high accuracy RTLS suitable for distribution and that will complement the needs for SMEs, large enterprises, universities and shopping centres.

Following initial research, an Android application was developed using the Cisco Context-Aware API which made use of the Mobility Service Engine (MSE) located within the department. This application investigated the accuracy and usability of the Cisco Context-Aware Mobility and assessed its capability as a RTLS. A hypothesis of this research was deemed to be: Cisco's Context-Aware Mobility is not capable of providing a high accuracy RTLS. To evaluate this hypothesis, a number of tests were carried out on the Android application within the production network environment.

2. RELATED WORK

With the popularity of Wi-Fi increasing dramatically over the last few years, so has the need and use of Real Time Location Systems (RTLS). The availability of WLAN Access Points (APs) deployed almost everywhere including universities, hotels and shopping centres to name but a few, allows RTLS to calculate one's location using a number of different techniques and approaches. These approaches differ in terms of the measurement techniques used to sense and measure the position of the device. There are also a number of factors that influence the technique that RTLS choose to implement such as cost, accuracy, performance, and environment. The following section expands on some of the techniques used.

2.1. Cell of Origin

Cell of Origin (COO) or Nearest Cell is one of the simplest forms of location tracking. Although simple to implement, this technique makes no attempt to resolve the exact position of the mobile device other than the cell that it has been located in and therefore lacks accuracy (Cisco, 2010d). This is mainly due to the large size of the cells (Retscher and Fu, 2010). The most basic of cell networks contain transmitters that transmit evenly in all directions creating a circle. Due to the fact that circles do not tessellate well, the network cells are usually calculated as hexagons (Jagoe, 2002). This technique is used due to its relative ease of implementation which negates the need for complex algorithms thus providing fast positioning performance. The accuracy of this technique is based on the density of the tower, with a higher density proving better accuracy due to the coverage area being smaller (Ching et al., 2010). However, most towers have low densities which mean large coverage areas that only provide accuracies of hundreds of meters and over. Like all positioning techniques it does have its disadvantages. One of its main downfalls is its lack of accuracy and due to various reasons mobile devices can be associated with far away cells, even though there are closer cells at hand. This incorrect association to cells can really affect the accuracy in multi-story buildings where floor to floor cell overlap occur (Cisco, 2008b). There are procedures that can be taken to increase the accuracy of the nearest cell technique, such as using the Received Signal Strength Indication (RSSI) provided by the cells and noting the highest signal strength. The mobile device is then associated with the cell that records the highest signal strength thus increasing the possibility of selecting the correct nearest cell. This technique is good for low cost, good performance and non-critical tracking, but it would not be suitable for users requiring finer accuracy. COO can provide accuracies of 100m up to 1km for urban areas and 35km for rural areas providing a response time between 2 to 5 seconds (Jagoe, 2002), (Kos, Grgic and Kitarovic, 2007). Indoor APs can provide a far greater density than that of a mobile cell tower, hence, the COO can provide greater accuracy than indoor GPS (Ching et al., 2010). According to (Jagoe, 2002) COO was the most common method used by positioning systems in 2001.

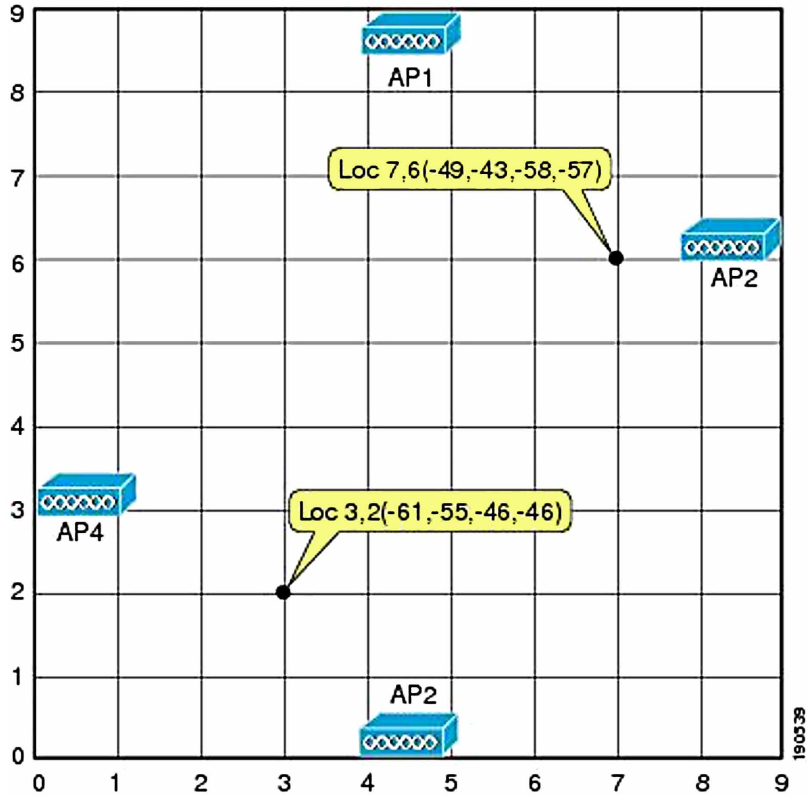
2.2. Lateration

2.2.1. Time of Arrival

Time of Arrival (ToA) is based on the time of arrival of a signal sent from a mobile device to two or more sensors (Kos, Grgic and Kitarovic, 2007; Llombart, Ciurana and Barcelo-Arroyo, 2008). "The ToA technique requires very precise knowledge of the transmission start time(s), and must ensure that all receiving sensors as well as the mobile device are accurately synchronized with a precise time source" (Cisco, 2008b). ToA determines the distance between the mobile device and the receiving sensors by calculating the time it takes for the signal to travel between them (Kos, Grgic and Kitarovic, 2007). The calculated distance is used as a radius to create a circular plot around each respective receiving sensor, with the mobile device believed to be placed somewhere along each plot (Llombart, Ciurana and Barcelo-Arroyo, 2008). The location of the mobile device is pinpointed where the plots intersect as shown in Figure 1 or the intersecting area. Cases in which three sensors are used are referred as ToA tri-lateration and cases in which four or more sensors are used are referred to as ToA multi-lateration. ToA can also be constructed in 3D using spheres instead of circular plots, which is the technique incorporated by Global Positioning Systems (GPS).

(Liu and Yang, 2011; Llombart, Ciurana and Barcelo-Arroyo, 2008) State that RF trilateration requires at least three APs to calculate distances and determine location. However, they also state that high positioning accuracy using trilateration can be difficult to achieve due to multipath interference and the difficulty of obtaining detailed coordinates of all the APs. The difficulty in obtaining accurate coordinates of APs in buildings can be due to building structures, network services and mobile services including attenuation, multipath, occlusion and reflection (Liu and Yang, 2011; Brown and Dunn, 2011). ToA systems are flexible, non-complex and are highly adaptive to changing environments (Llombart, Ciurana and Barcelo-Arroyo, 2008). According to (Yu et al., 2012) ToA is more accurate than Received Signal Strength (RSS) techniques and more practical than Time Difference of Arrival (TDoA) and Angle of Arrival (AoA) which are discussed later. However, due to the need for precise time synchronization very small errors in time can result in very large positioning errors. According to (Llombart, Ciurana and Barcelo-

Figure 1. Calibration phase (Cisco, 2006)

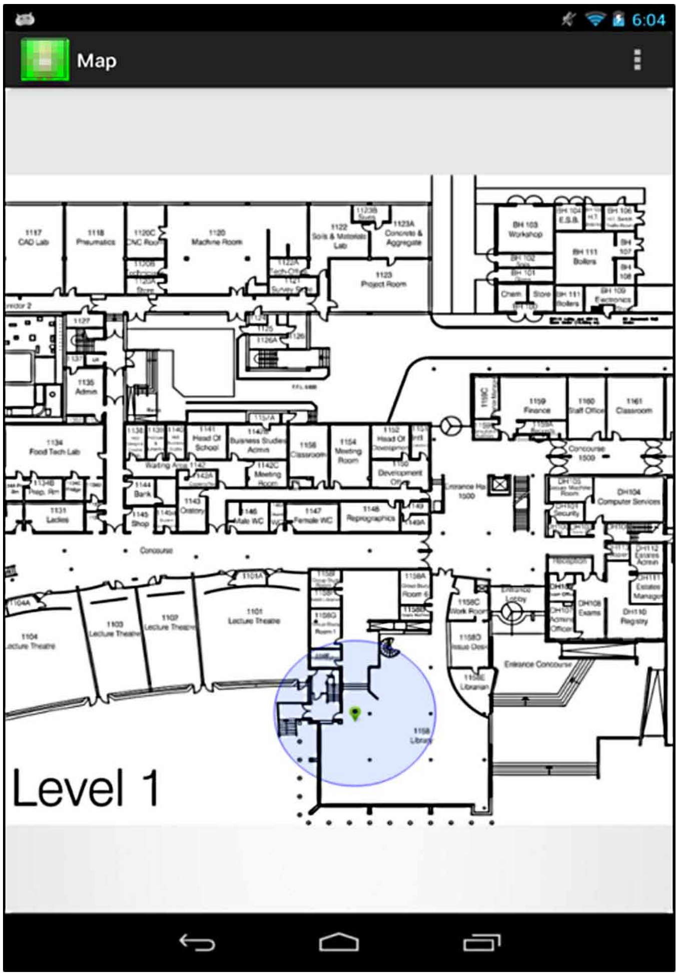


Arroyo, 2008) a time inaccuracy of 1 microsecond can lead to a distance error of up to 300m. ToA based methods suffer when utilised by a large number of people due to the traffic generated by the system. This is due to the ToA- based system generating traffic when calculating distances (Llombart, Ciurana and Barcelo-Arroyo, 2008). Using the ToA technique accuracies of up to 1 to 2 meters can be achieved on a WLAN network when an investment in dedicated hardware (synchronized clocks) is made (Ciurana et al., 2010). However, (Ciurana et al., 2010) suggests a software only approach by using the CPU clock of the mobile device to measure the Round Trip Time (RTT) that is required to calculate distances.

2.2.2. Time Difference of Arrival

Time Difference of Arrival (TDoA) is based on the time of arrival of a signal sent from a mobile device to three or more sensors (Morgan, 2009; Abdul-Latif, Sheperd and Pennock, 2007). However, unlike ToA, TDoA only requires the receiving sensors to have synchronized time sources and not the mobile device. This means that TDoA does not require knowledge of the transmission start time (Cisco, 2010c). TDoA is calculated using hyperbolic tri-lateration which requires at least three synchronized receiving sensors. As shown in Figure 2 the time taken for a mobile device to transmit a signal to receiving sensor A and receiving sensor B is recorded. The time difference of the transmission is calculated by subtracting the two times. Therefore, if

Figure 2. Android app map activity



sensor A receives the transmission before sensor B then sensor A is closer. Estimated TDoAs are converted into distance measurements which result in a set of nonlinear hyperbolic curves; where the curves intersect indicate the position of the mobile device (Elkamchouchi and Mofeed, 2005).

TDoA systems can obtain greater accuracy with greater bandwidth and transmit power. It also states that indoor TDoA and ToA systems are best suited to large, open buildings that contain high ceilings and little obstructions allowing adequate clearance between the ceiling and building contents. TDoA techniques have poor accuracy due to technical limitations and propagation environments, but its poor accuracy is mainly due to Non Line of Sight (NLOS). NLOS is when a direct line of sight between the mobile device and the base station (AP) is obstructed by obstacles (Kim, Lee and Park, 2008). NLOS propagation error results in the transmitted signal taking longer to arrive at the receiver as the signal is reflected and diffracted which results in a greater travelling distance (Kim, Lee and Park, 2008). A solution to the NLOS propagation error is to apply TDOA error modelling at each base station (Kim, Lee and Park, 2008). It concluded that using the error modelling algorithm the NLOS error can be mitigated by 45 to 77%.

2.2.3. Received Signal Strength Indication

Received Signal Strength Indication (RSSI) is another technique which uses either a mobile device or a receiving sensor to measure the signal strength of the transmission in order to determine location. Typically in an RSSI environment, a location server will know the locations of all the receiving sensors (measurements devices). The receiving sensors measure the signal strength from a transmission sent from a mobile device and forward it on to a calculation engine. The calculation engine then locates the mobile device using triangulation or other known techniques taking into account the transmitter output power, cable losses and antenna gains (Cisco, 2010c).

It is also assumed in direct line of sight situations, that if two signals are received by the receiving sensor from two mobile devices, then the mobile device with the lowest signal strength is the furthest away. In most situations a direct line of sight is not obtainable as environments such as warehouses and offices contain many obstructing objects and reflective surfaces (Cisco, 2010c). This means that the direction of the signal sent from the mobile device to the receiver is not direct. Instead it is reflected off a surface such as a wall or ceiling, meaning that the signal has to travel further thus producing lower signal strength when detected by the receiver (Cisco, 2010c). This multipath interference can impact on the accuracy of the positioning system as it believes that the mobile device is further away than it actually is. The RSSI technique is an attractive one as it does not require any specialised hardware at either the mobile device or the receiver, making it a cost effective choice especially for those of 802.11 WLAN systems wanting to offer location systems. The pure RSSI technique does not implement measures to account for attenuation and multipath, hence resulting in loss of accuracy which leads to unacceptable results. Therefore, it is best suited to controlled environments where a direct line of sight is attainable (Cisco, 2010c).

2.3. Angulation

Angle of Arrival (AoA) or Direction of Arrival (DoA) is a technique in which the location of the mobile device is located by calculating the angle of incidence of the signals received by the receiving sensors. In order to measure the angles an antenna array is required (Niculescu and Nath, 2003; Abdul-Latif, Sheperd and Pennock, 2007; Drane, Mac Naughton and Scott, 1998). The location of the mobile device is determined by the intersecting lines in which the signal is received by at least two receiving sensors (Abdul-Latif, Sheperd and Pennock, 2007). Improved accuracy can be achieved from three or more receiving sensors. In situations where a clear line of sight is attainable between the mobile device and the receiving sensors, mechanical antennas at the receivers adjust to point to the highest signal strength received. These antennas can be used to calculate the direction in which the signal is received (angle of incidence) determining the lines of bearing (Cisco, 2008b). It is where these lines intersect that the position of the mobile device is located.

AoA techniques have previously been used in the cellular industry in order to determine the location of mobile phone users in cases of emergency. This was achieved using numerous towers to calculate the angle of arrival of the signal and by converting this information to longitude and latitude coordinates. However, the AoA technique struggles for accuracy when a clear line of sight is not attainable, for instance, when challenged by multipath interference such as reflection from surrounding objects (Wong, Klukas and Messier, 2008). According to (Abdul-Latif, Sheperd and Pennock, 2007) a small error in angle measurement can result in a large error in positioning. Increasing distances between the mobile device and the AP results in accuracy degradation. However, within an indoor WLAN environment the APs are close to the mobile device so accuracy is not affected (Wong, Klukas and Messier, 2008). (Wong, Klukas and Messier, 2008) tested two algorithms Maximum likelihood (ML) and Space-Alternating Expectation-Maximization (SAGE) to determine the performance of their system in which it concluded that SAGE was not suitable for determining the AoA of a mobile device using indoor APs. However, it states that using the ML algorithm has the potential of determining position with less than 2m accuracy.

2.4. Location Patterning

Location Patterning or Fingerprinting is a technique where no specialised hardware is required by either the mobile device or the receiving sensors allowing it to be fully implemented in software (Chang, Rashidzadeh and Ahmadi, 2010), (Lui et al., 2011). The Location Patterning technique makes assumptions that each mobile device location will contain a unique Radio Frequency (RF) signature and also that each building or floor will contain unique propagation characteristics meaning no two buildings or floors should be the same in terms of RF pattern recognition (Cisco, 2006). Although most commonly Location Patterning is based on RSSI it can be implemented to include ToA, TDoA and AoA techniques. The Location Patterning technique consists of two main phases: Calibration phase and Operation phase.

2.4.1. Calibration Phase

The calibration phase involves walking around the chosen environment with a mobile device and using the receiving sensors (APs) to measure the signal strength of the mobile device to accumulate the required data (Del Mundo et al., 2011; Chang, Rashidzadeh and Ahmadi, 2009).

In order to acquire this data, a set of reference points are placed on a map of the environment to precisely mark where the data should be recorded. At each marker an array of RSS values associated with the mobile device are recorded in a database along with the coordinates of the reference point (Shin et al., 2010). This is usually referred to as a radio map. The array size of the RSS values is determined by the number of receiving sensors detected by the mobile device. Figure 1 shows a simple representation of the calibration phase.

According to (Cisco, 2006) the recorded signal strength of a mobile device will vary and due to this, multiple samples are required at each reference point during the calibration phase. A method used to simplify this procedure is to record a mean value of all the measurements taken at each specific location for a specific mobile device, meaning that the array of RSS values becomes an array of mean RSS values.

2.4.2. Operation Phase

The operation phase consists of the receiving sensors forwarding the signal strength received from the mobile device to a calculation engine. The calculation engine then performs complex algorithms with the use of the radio map, allowing it to retrieve the location of the mobile device (Koo and Cha, 2012). There are two ways to compare fingerprints: deterministic and probabilistic. Deterministic calculates the difference between the current reference fingerprint and all those recorded during the calibration phase. The fingerprint in the radio map with the smallest difference from the reference fingerprint is selected as the mobile devices position (Ching et al., 2010). The probabilistic approach calculates the probability that the user is at a specific reference point. This method requires more sampling at each reference point during the calibration phase (Ching et al., 2010). During the operation phase, the system calculates the probability of the mobile device being at each reference point in the radio map. The reference point with the highest probability is selected as the mobile device's position (Meng et al., 2011).

The most common algorithms used include K-Nearest Neighbour (k-NN), Naive Bayes Classifier (NBC) or Support Vector Machine (SVM). A comparison carried out by (Del Mundo et al., 2011) found that k-NN had an accuracy of 0.7294 within 1.96 meters, NBC achieved an accuracy of 0.8173 within 1.96 meters and SVM achieved an accuracy of 0.8267 within 1.96 meters. It concluded that SVM outperformed NBC in terms of accuracy and found that k-NN was the least accurate of the three. Location Patterning techniques perform well in indoor environments and provide good accuracy when the mobile device is detected by at least three receiving sensors (Cisco, 2006). In order to increase the accuracy, the mobile device must be detected by six to ten receiving sensors which in turn provides an accuracy of up to five meters. A benefit of using Location Patterning techniques is that it can be used with existing infrastructure such as 802.11 WLANs, and it also accounts for attenuation and multipath interference during the calibration phase. According to (Del Mundo et al., 2011) Fingerprinting techniques perform better than that of cell based (Cell of Origin), lateration (ToA, TDoA) and angulation (AoA) techniques in terms of positioning accuracies.

However, a disadvantage to this technique is the need for a high number of APs in tight spacing. Another drawback to this technique is the calibration phase. The radio maps used tend to be very specific to the area in which they were created for regardless of the structure of the buildings. The chances of identical buildings having the same radio maps are very low which in turn provide little re-use of the radio maps. As mentioned earlier the results of the calibration phase are likely to vary, especially overtime as changes occur that affect RF propagation such as changing environments and so forth. (Ching et al., 2010). This requires re-calibration in order to achieve a consistence level of accuracy, with complete re-calibration required twice annually in some environments (Ching et al., 2010; Meng et al., 2011). There have been many suggestions to improve the accuracy of the Location Patterning technique. Some of these look at improving the calibration phase by allowing the end user to update the radio map by correcting the actual position (Gallagher et al., 2010), (Koo and Cha, 2012) where (Kim, Chon and Cha, 2012) uses an autonomous calibration phase. Others look to combine a number of techniques such as GPS, RSS, accelerometers and digital compasses to help improve accuracies (Chun et al., 2011; Chon and Cha, 2011; Kim, Shin and Cha, 2012).

2.5. RF Fingerprinting

RF Fingerprinting uses the Received RSSI approach to provide indoor location performance which only Location Patterning techniques were capable of. It works in a very similar way to Location Patterning, but with more speed, efficiency and improved accuracy. "RF Fingerprinting significantly enhances RSS lateration by using RF propagation models developed from radio propagation data gathered directly from the target environment or environments very similar to it" (Cisco, 2008a). The calibration phase for RF Fingerprinting is the same as that in Location Patterning in which the coordinates of the reference point is stored along with the mobile devices RSSI from three or more APs. The accumulated calibration data is processed and used to build an RF propagation model which calculates path loss and shadow fading standard deviation to account for propagation discrepancies in the current environment. RF Fingerprinting also allows the use of models created during the calibration phase or pre-packaged RF propagation models.

According to Cisco, the RF Fingerprinting technique performs better than other techniques such as triangulation or RSS lateration as these techniques make no attempt to account for attenuation or any environmental considerations. It also states that RF Fingerprinting applies 'statistical analysis techniques' on the calibration data allowing it to rule out unconvincing possibilities and further improve location accuracy (Cisco, 2008a). RF Fingerprinting does not require the same calibration effort as Location Patterning nor does it require re-calibrating as often as Location Patterning. Unlike Location Patterning, RF Fingerprinting can use the same RF model in similar environments or it can use pre-packaged RF models which allow for swift deployment.

3. CISCO CONTEXT AWARE MOBILITY

Most Context-Aware Mobility solutions used today are Wi-Fi based due to the deployment of WLAN by the majority of enterprises and also that most mobile devices contain Wi-Fi radios. Context-Aware Mobility solutions enable enterprises to retrieve and use contextual information regarding mobile assets to help optimise and change business processes. Information can be collected on any mobile assets involved in the business process, including that of devices, people or products. Cisco's Context-Aware Mobility Service is used to capture such information and more, such as location, temperature, availability and applications used, whilst also providing the ability to integrate this information with other systems to better enhance business functionality. Cisco's Context-Aware API allows users to create customised context-aware programs that interact with Mobility Service. The following section expands on the architecture and functionality of Cisco's Context-Aware Mobility API.

The Context-Aware Mobility Service can be used by either the 3300 Series Mobility Service Engine (MSE) or the 2700 Series Wireless Location Appliance which in turn communicate with the Wireless Control System (WCS). The WCS is a WLAN management tool that provides the ability to manage network controllers through a web based user interface. It allows users to configure and monitor controllers and access points. It also provides the ability to edit building floor plans such as changing the types of walls, locations of APs and also provide heat maps of the selected floors (Cisco, 2011). The MSE communicates with WLAN Controllers (WLC), the WCS and the Location Server. "*MSE is the integration point for applications through well-defined APIs that help in Asset tracking for location.*" (Cisco, 2012b). It allows integration between Cisco technologies and their partners and it is used to gather statistics and data of clients of WLC and WCS. Applications such as the WCS or partner application wanting to use MSE services can do so by communicating through APIs (Context-Aware API) using XML/SOAP. The Mobility Service Engine provides real time location tracking and historical trends of rogue devices, Wi-Fi clients and RFID tags allowing business to gather a better understanding of client movement and habits.

The Context-Aware Mobility Service runs on the MSE and can be combined with other mobility services to provide even more functionality. It can also be deployed across multiple MSE providing a scalable network design (Cisco, 2010a). In order to track the location of a mobile device, the MSE running the Context-Aware Mobility Service gathers information from all of its controllers and their associated APs in the chosen environment. As the Context-Aware Mobility Service runs on the MSE instead of a single controller, it can aggregate all of the AP measurements from numerous controllers (Cisco, 2010a). This also ensures that the location calculations can be performed at rapid speeds (a few seconds), which is necessary for active location tracking information to be of any use to its consumers. The Cisco 3310 Mobility Service Engine can cater for up to 2,000 clients and tags combined and the Cisco 3350 Mobility Service Engine can cater for up to 25,000 clients and tags combined. It supports Wi-Fi 802.11 a/b/g/n networks.

The Context-Aware Mobility API provides customers and partners the ability to build and use customised applications that interface with the MSE. The Context-Aware Mobility API communicates with the MSE through SOAP (Simple Object Access Protocol)/XML (Extensible Markup Language) over HTTP/S (Hypertext Transfer Protocol/Secure). This involves a request being sent from the client to the server to get or set information and a response being sent from the server to the client in response to the request. It can also involve notifications being sent from the server to the client asynchronously when a specified event is detected by the server (Cisco, 2010d). The Context-Aware API consists of a number of modules (Cisco, 2010d) which include server connection, Server Administration, Network Design Interface, Tag, Rogue Clients, Rogue Access Point, Presence, Asynchronous Notifications and the Mobile Station module. The server connection API is used to authenticate clients against the server database and to establish a business session. The business session identifies the user's privileges and enables them to login and logout from the server. The server administration API is used to retrieve the context-aware server information such as the server details and objects that have changed within a certain date. The network design interface is used to obtain a specific network design object or a list of network design objects contained in the location server. The tag API is used to monitor and control up to 2500 active RFID tags. This includes the ability to add, edit and delete tags, retrieve the location of specific tags or a list of tags, and also retrieve tag history and statistics.

The rogue client API is used to identify and control rogue clients located on the WLAN. This provides the ability to retrieve rogue client information, location history and to delete a rogue client from the server. The rogue access point API is used to identify and control rogue access points. It provides the same functionality as that for the rogue clients. The presence API provides the ability to configure parameters such as location resolution type, location format and response encoding, allowing users to retrieve client locations in encoded format from the MSE. Another module is the asynchronous notification API, which is used to register and set asynchronous events within the server. Notifications are created based on a set of triggers that are set against a tracked asset within the system. Triggers can be set to notify users on a number of events. If an asset is within a certain area, building or floor, or the status of the devices battery life. It can also be used to notify if the tracked asset has moved a certain distance greater than, or less than, the specified amount or if the asset is not located on the server. Finally, the mobile station API is used to identify and control 802.11 clients located on the network. This includes laptops, phones, tablets etc. all of which are referred to as stations. The API provides a number of methods to monitor and control stations. These include methods to add, edit and delete stations, retrieve station information, station statistics, station location and station history for the specified device. There are also methods available to retrieve the same information on an entire list of stations.

The MSE and Context-Aware Mobility API can provide its consumers with many benefits (Cisco, 2010b) such as flexibility, allowing the user to determine whether to use one or multiple modules depending on their needs and the data can be retrieved using a pull (query) or push (subscription) model. The API can also be used with any programming model. It provides network usability in which the API provides contextual network information without the need for the user to understand how this information is retrieved or how it is represented by the different devices in which the MSE gathers data from. This allows the user to apply more focus on delivering the service which uses the contextual data.

It also provides location usability as the API uses the physical environment to provide location information, removing the underlying location techniques used to calculate the location. This allows users to use the API without the need to understand these location techniques. Another benefit is accessibility, where the API communicates with the MSE using SOAP/XML it allows web based services such as .Net, C++, Java, etc. to access this information using web technologies such as WSDL. It also provides the benefit of locating multiple devices. The API caters for many devices such as stations, tags, and rogue devices etc. providing contextual data through a common form. This allows the user to become familiar with any of the API modules once they are familiar with one of them. It also provides contextual combination based on statistical information and network access information. This allows users to build products that use this information without the need of having to retrieve it from multiple sources.

4. ANDROID LOCATION DETERMINATION APP

It was initially intended to connect to the Cisco MSE and retrieve location details directly from the android device using the Cisco API. However, after some investigation it was clear that this was not feasible due to the API using Remote Method Invocation (RMI) which is not supported by android. Therefore, the decision was taken to deploy the necessary API methods on a web service, which in turn could be called by the android device. JAX_WS was chosen as the programming model as it simplifies application development and has replaced the Remote Procedure Call programming model (JAX-RPC) as the current industry choice (Apache, 2012). The web service consists of a main class which is used to login to the MSE, get the required location information and logout of the MSE. The Location class is used to hold the location information retrieved from the web service methods and is then passed to the android device. All of the web service methods were excluded from the WSDL, apart from two, which only consisted of the actual location details, hence hiding the connection calls to the MSE. The sequence of interactions between the android device, the web service and the Cisco MSE are as follows:

1. The android device calls the get Location web service method.
2. The web services call the API login method to login to the server.
3. The MSE server authenticates the user and creates a new session.
4. The session accesses the cisco database, retrieving the user's location information.
5. The location information is passed back to the session, to web service and finally to android device.
6. The android device then starts the activity to display the map, which contains the user's position.
7. The update Location method is performed in the same manner as above.

The app consists of three activities. The first activity is the SignIn activity which checks if the Wi-Fi is enabled. If the Wi-Fi is disabled, then the user is prompted for permission to allow the app to turn on the Wi-Fi. The Wi-Fi must be enabled on the android device in order for the MSE to detect its location. The SignIn activity then calls the web service method to get the devices location. Upon a successful response from the web service, the Map activity is started. The Map activity displays the map of the college, and placing the location of the device on the map. The Map activity is used to update the devices location by calling the web service method. It also contains all the necessary classes which allow the user to interact with the map, such as zooming and panning. The Details activity can be started from the Map activity, allowing the user to view the complete location details of the device. The Map Package contains all the necessary classes that are required to display and interact with the map. This includes a MapLocation class which is used to hold the image of the map and the user's location. The EventHandler class is used to handle the user interaction with the device, such as zooming (double tap and pinch to zoom) and panning. The AspectQuotient class is used to determine the aspect ratio of the device and the map image which is then used to display the map. The MapControl class is used to control the zoom level and limit the amount of panning that can be performed and the DrawMap class is used to draw the map and marker and limit its boundaries to that of the devices screen.

4.1. Web Service

The Web Service package contains the necessary classes and methods that are required for the android app to communicate with the Web Service. This includes the relevant connection details and the building of the SOAP/XML. It also contains the necessary methods for calling the Web Service along with the parsing of the Web Service response from XML into a Location object.

The Web Service calls are run on a separate thread from the main UI thread. This is enforced by Android and requires the use of AsyncTasks. “This class allows to perform background operations and publish results on the UI thread without having to manipulate threads and/or handlers.” (Google, 2013).

Using the AsyncTask the web service calls can be performed in the background and upon completion the results of the calls can then be published on the UI thread.

The first step in implementing the JAX_WS Web Service was to add the required methods to login and logout of the MSE. The login method consisted of creating a new login instance which involved passing the username and password. The username and password are validated against the users stored in the MSE database. If the login is successful, a business session ID is returned which is used to identify the session and also identify the user’s privileges. This business session ID is required by all other methods of the MSE API. The logout method disconnects the user from the MSE, freeing up the business session and resources. Both the login and logout methods contain a AaaServiceStub for the mse-aaa.wsdl. Default username and password were used which granted access to the MSE server. The default username and password should be removed from the MSE database as it is a security exploit. The Web Service also requires an SSL certificate, which was retrieved from the MSE using the Cisco InstallCert Program and stored locally on the machine. The SSL certificate was added to the Web Service properties using the System.setProperty method in which the location of the certificate was passed to.

The next step was to implement the Station Location method. The Station Location method consisted of passing the business session ID, retrieved from calling the login method and the mac address of the android device. The Station Location method makes use of a LocationStub for the mse-location.wsdl. The mac address of the android device was passed up in the web service call getLocation and set using the setMacAddress method. Before the mac address is set, it is validated using regular expression to ensure that it a valid mac address. The Station Location method returned a response document object which contained all the location details of the android device. This included information such as the x and y coordinate of the mobile device in feet, the floor id, the mac address of the device, the mac address of the associated AP, the SSID, username used to connect to Wi-Fi and a textual representation of the floor hierarchy. A Network status was also included which indicates whether the network is active, inactive or legacy. The confidence factor is also contained within the Station Location response document. The confidence factor is a floating point scalar used to calculate an area which the MSE is 95% confident that the mobile device is located in. A Location object was also implemented to hold the location information retrieved from the MSE. This Location object is initialised in the getLocation method and the location details are assigned to it using the populateLocation method. The Location object is then return by the method. All of the web service methods were excluded from the WSDL apart from two (getLocation and updateLocation) which only consisted of the actual location details, hence hiding the connection calls to the MSE.

4.2. Android Application

The app (Figures 2 and 3) consists of three activities, SignIn activity, Map activity and Details activity. An activity is the presentation layer and handles the user interaction. The first activity created was the SignIn activity. Although the app does not require a login, it does require the Wi-Fi on the android device to be enabled. The reason for this is that the MSE detects a mobile devices location based on RSSI retrieved from the devices Wi-Fi. To ensure that the Wi-Fi is enabled the SignIn activity first checks whether the Wi-Fi is enabled on the android device. This involves creating a WifiManager object which controls the devices network state. The WifiManager check if the Wi-Fi is enabled before allowing the app to proceed any further. If the Wi-Fi is disabled, then a notice dialog is displayed indicating that Wi-Fi is required to use the app and provides the option to enable it. If the user selects to enable the Wi-Fi the WifiManager checks if the Wi-Fi is not currently being enabled and enables it. The app waits for fifteen seconds, allowing the device time to enable the Wi-Fi before proceeding.

Figure 3. Android app details activity

When the Wi-Fi is enabled, the android devices mac address is retrieved using a WifiInfo object. A WifiInfo object is used to hold the Wi-Fi information which can be retrieved from the WifiManager. A WifiInfo object can contain information such as Mac Address, RSSI, IPAddress and SSID to name but a few. The mac address is used by the MSE to identify which mobile device to retrieve location details for. The mac address is passed to the getLocation web service method which is discussed later in this chapter. Upon a successful response from the web service, a Location object is instantiated within the SignIn activity. The Location object is used to hold all of the location details received from the web service. The Location class implements the Parcelable interface allowing it to be passed from activity to activity. The Location object is stored in a Bundle object and the Bundle object is contained within an Intent. The Intent is then used to start the Map activity.

The Map activity is the most important activity as it is used to display the android device's location on the college map. Using the Location object passed from the SignIn activity, the location details are used by a number of classes which display the map and allow the user to interact with the map, such as zooming and panning. These classes are discussed later in this chapter. The Map activity also contains a web service method to update the devices location. This method is automatically called every fifty seconds. The Map activity contains three options in the options menu, one to update the location manually, one to display the full location details and the other to display the log file.

The Details activity can only be accessed by pressing the Details button contained within the Map activity action bar. The Location object used by the Map activity is passed to the Details activity which simply displays the information held by the Location object using textviews.

The log file contains a record of the location details every time the getLocation or updateLocation web service methods are called. The log file can be accessed by and external application by pressing the View Log button contained within the Map activity action bar.

Upon the retrieval of location details from the web service, details are displayed on the android device. This is performed using a combination of classes. The MapLocation class is used to hold all of the details required to display the devices position on the map. This includes the X and Y position, the radius to display the confidence factor of the returned results indicating that the device is located within the displayed circle and the image of the marker. As the X and Y positions are returned in feet by the MSE, they require converting to suit the size of the map. This is done by first converting the positions from feet to inches. This is then divided by the scale of the map before being multiplied by the pixel DPI (Dots per Inch) of the image. The X and Y position are then given an offset so that they are placed in the correct position on the map. The pixel values are then converted to DIP (Density-Independent Pixel) so that regardless of the screen size of the device the location will be shown in the correct position on the map.

5. EVALUATION

The campus consists of three buildings, however only the main campus building was used for testing purposes. This building was chosen as it accounts for the most activity within the campus. This allowed the Android app to be tested at both busy and quiet periods in order to determine if network activity and interference affected the MSE results. The app was tested on an ASUS Nexus 7 android tablet running android version 4.2.2 Jelly Bean.

Testing was carried out in the main building using all three floors. Ten locations were chosen and the system was tested ten times resulting in one hundred sample points. The tests were carried out at different times with various numbers of other clients connected to the college network.

Testing was focused around three key elements. This included Level Differentiation Testing which determined how often the MSE detected the mobile device on the correct floor. The results are discussed in section 5.3.1, where an accuracy of 100% relates to the device being detected on the correct floor for each test and an accuracy of 0% relates to the device never being detected on the correct floor. The AP Differentiation Testing determined the consistency of the APs by noting the mac address of the detecting controller. The results are discussed in section 5.3.2, where a consistency of 100% relates to the device being detected by the same AP for each test and an accuracy of 0% relates to the device never being detected by the same AP. The Distance Accuracy Testing determines how accurate the MSE is in terms of location. The results are discussed in section 5.3.3, where an accuracy of x amount in feet relates to the distance of the detected location from the actual location. The Distance Accuracy Testing also determined the accuracy of the MSE confidence factor which is an area that the MSE is 95% confident that the device is located in. The confidence factor values listed in section 5.3.3 relates to the radius of the detected area.

5.1. Results

5.1.1. Level Differentiation Testing

The MSE determines which floor the mobile device is detected on and returns the floor ID for the corresponding floor. This floor ID is returned by the web service and is used to determine the accuracy of the MSE detecting the correct floor within a multi-storey building. Of the one hundred sample points recorded, only forty of these had 100% accuracy of being detected on the correct floor. Of the remaining sixty sample points, twenty had an accuracy of 90% and twenty had an accuracy of 80%. Of these sample points, location five was the least accurate, resulting in only 10% accuracy of being detected on the correct floor, taken from ten sample points.

5.1.2. AP Differentiation Testing

Cisco's MSE uses the Fingerprint method to determine the mobile devices location. As discussed previously, the Fingerprint method revolves around the measuring of RSS from the mobile device. This signal strength is recorded by the APs which detect the mobile device, therefore, it was deemed applicable to record the consistency of these APs. Contained within the Cisco API is a method to return the Mac Address of the AP that detected the mobile device. This Mac address was noted for all one hundred sampling points. The results revealed that only one location (location ten) had 100% consistency. Five locations had a consistency of 80% with another three locations having a consistency of 60%. Of the ten locations, location five again performed the poorest with a consistency of 40%.

5.1.3. Distance Accuracy Testing

The x and y coordinates along with the confidence factor are recorded in feet by the MSE. Therefore, for simplicity, the testing of locations and results were also recorded in feet. Note that due to the small scale of the map used within the android app all sample points are measured with an accuracy of plus or minus 10ft. The results revealed that location one performed the best with and average accuracy of 34ft and an average confidence factor of 93ft. Locations four, seven, eight and ten also performed reasonably well with an average accuracy of 48ft, 53ft, 61ft and 65ft respectively. However, locations four and seven's confidence factor was over twice as high as that of location one with an average accuracy of 189ft and 203ft respectively, with location eight having an average confidence factor accuracy of 123ft. Of all the tests, location ten had the lowest average confidence factor with an accuracy of 53ft, however, on 80% of occasions, the mobile device was not detected within the confidence factor range. Locations two and three had near identical results with an average accuracy of 65ft and 68ft respectively, with an average confidence factor accuracy of 221ft and 192ft respectively. Locations six and nine were amongst the poorest performers with an average accuracy of 116ft and 150ft respectively. These two locations also recorded the highest confidence factor values with an average accuracy of 275ft and 217ft respectively. It was expected that given the results of the level differentiation and AP differentiation tests that location five would also record the poorest accuracy, but this was not the case. Although still relatively poor, location five had an average accuracy of 114ft and an average confidence factor of 189ft.

The number of wirelessly connected clients on the college network was also recorded to see if the levels of network activity had any bearing on the performance of the locations. The results indicate that the level of activity within the college did not seem to have any bearing on the accuracy, as the results recorded during busy periods outperformed that of quiet periods in some instances and vice versa. (Cisco, 2012a) say that an accuracy of less than ten meters should be attainable 90% of the time and an accuracy of less than five meters should be attainable 50% of the time. This differs vastly with the results retrieved from the MSE in the college where the results reveal an estimated average accuracy of 77ft (7 – 45meters) and an estimated average confidence factor of 175ft (13 – 80 meters) for all one hundred samples. The results also differ vastly to that of other techniques mention in Chapter 2, as it lacks in comparison with ToA and AoA techniques which claim to be capable of accuracies of 1 - 2 meters. It also lacks in comparison to RSSI, TDoA and Location Patterning techniques which claim to be capable of accuracies of 2 - 5 meters. However, it must be noted that these other techniques would require testing in the college or a similar environment in order to properly compare them to Cisco's Context-Aware mobility. Figure 4 shows the average accuracy for the ten sample locations. Figure 5 shows the average confidence factor for the ten sample locations.

5.2. Interpreting the Results

To correctly interpret the results of the tests mentioned above, a number of APs and RSSI values would need to be collected for all of the one hundred sample points. The Cisco Context-Aware API provides this capability by providing a list of neighbouring AP data and a list or RSSI values. However, when this functionality was implemented in the Web Service, both the neighbouring AP data list and the RSSI list returned empty. Due to time constraints, it was not possible to investigate this further and therefore the interpretations are mainly based on assumptions. Location one performed the best and it is assumed that this is due to it being in close proximity to four APs with two of them located in the same room. This was also verified through the WCS by checking that the associated AP was visible by the other three APs. This indicates why location one had good results, as stated earlier, the higher number of detecting APs, the better the accuracy.

Location six and nine had the least accurate results of all ten locations. When these were investigated, it was clear to see in Figure 6 and Figure 7 that both locations are in close proximity to only one AP, therefore, it is assumed that the poor results are due to them being detected by mainly one AP. This was also verified through the WCS where both the closest APs were checked. The WCS confirmed that both APs for location six and nine were not visible to their surrounding APs.

Location ten was also investigated as it had the smallest confidence factor range, however for the majority (80%) of tests, the device was not located within this range. It was clear from the

Figure 4. Average distance accuracy

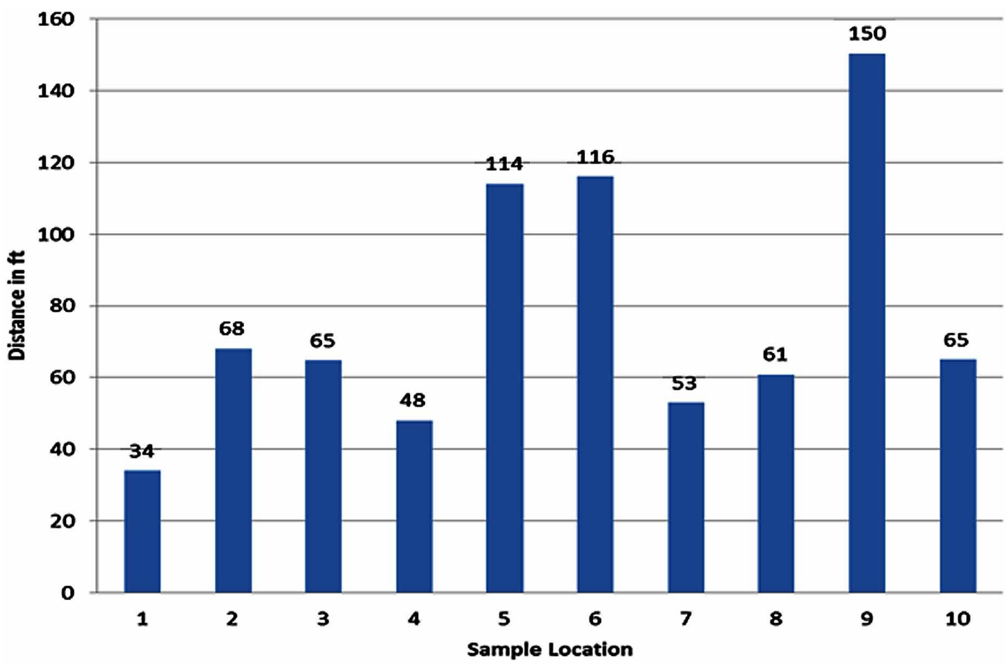
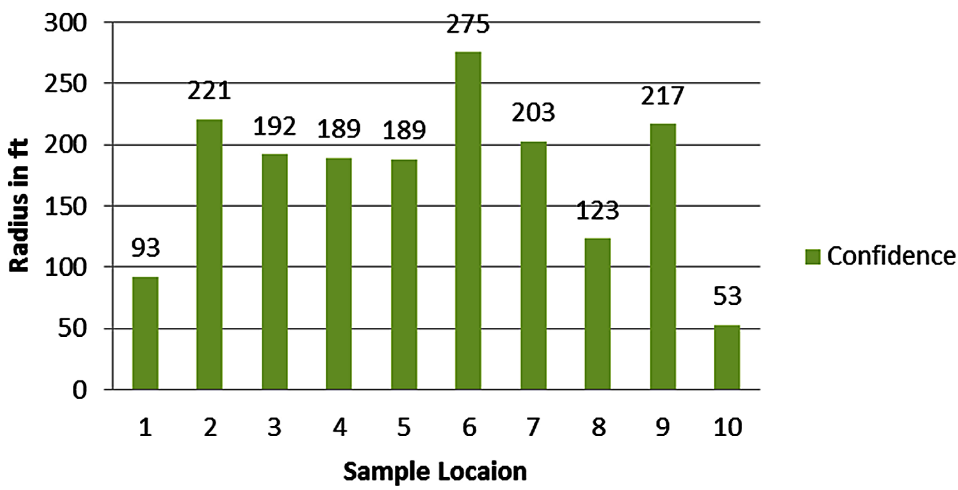


Figure 5. Average confidence factor



results that although there was an AP directly positioned at location ten, the majority of results were detecting the device close to a different AP which was located just a number of meters away. It was assumed that due to these results that there could have been a calibration error. The mac address of the two APs involved were first physically checked and then checked on the WCS where it was clear that the two APs were mixed up and therefore configured incorrectly for the MSE. This confirmed why the results were closer to the AP that was further away and why the

Figure 6. Position 6 results - scatter diagram

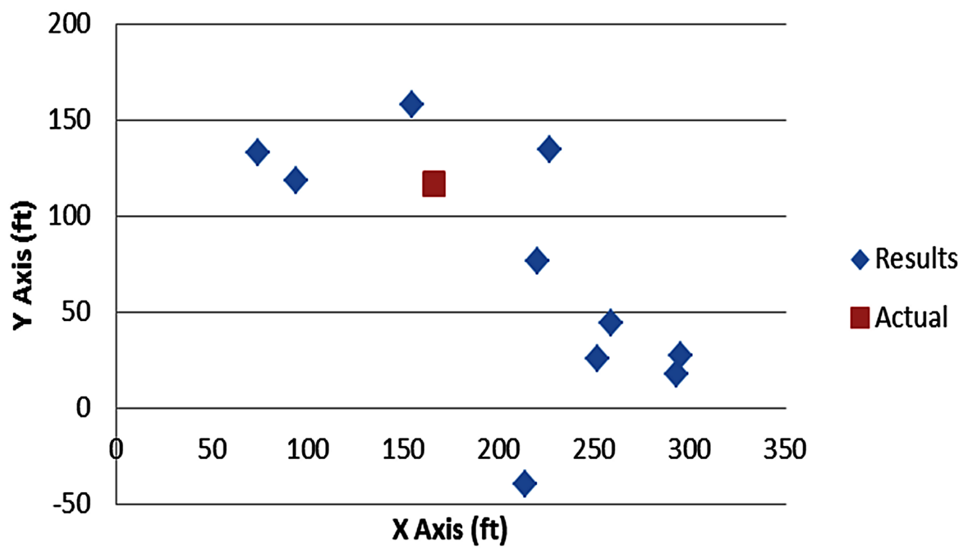
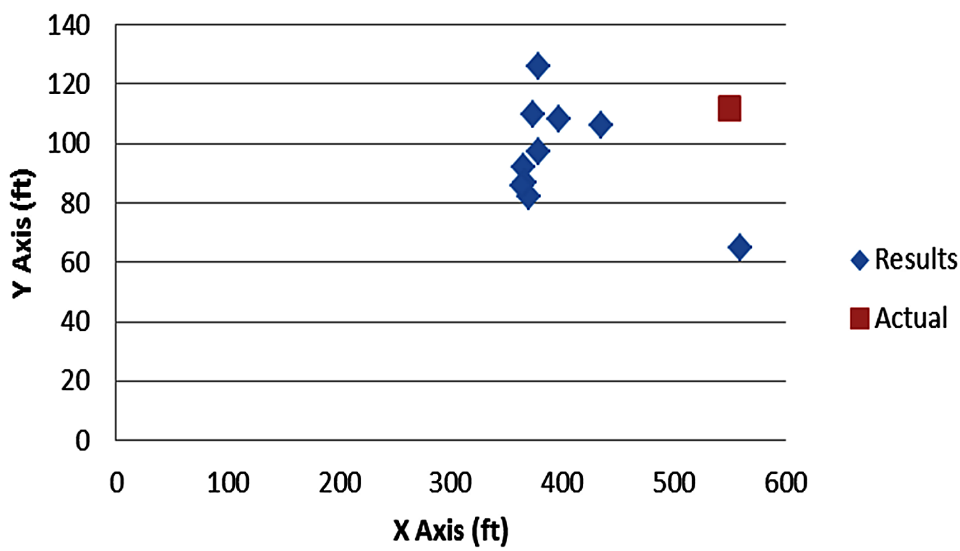


Figure 7. Position 9 results - scatter diagram



confidence factor was so small. If these APs weren't mixed up and configured correctly, then the device would be located within the confidence factor range and the distance from the sample point would be the smallest, hence, location ten would have the best accuracy. It was also noted during testing that the response time of location requests made to the MSE was slow, with each response taking around thirty seconds. The greater the number of devices being used within the college, the greater the processing load on the MSE. This affects the latency of the network, where in the case of the MSE, it refers to the delay between when the RSSI information of a mobile device is received by the MSE and when the location of this mobile device is calculated by the MSE. Latency can also be related to the NMSP (Network Mobile Service Protocol) aggregation window. This however can be tuned, allowing the user to set the time between location calculations and updates.

It is assumed that other factors have also impacted on the results of the ten locations. These could vary between the surrounding structure (steel, concrete, plasterboard), multipath interference, positioning of APs and also the configuration and calibration of the MSE, all of which can have an impact on the accuracy of the system. Factors such as impedance from fire doors, concrete walls and windows can have an impact on accuracy. Signal attenuation varies depending on the material that the signal passes through. An example of this would be if a signal of 15dBm enters through an office window, its signal strength will be reduced to 12dBm when it exits the window. The college's main building consists of concrete and brick walls and also contains a large amount of metal content. A high amount of metal content results in reflected signal creating multipath distortion. Calibration and positioning of APs can also have an effect on accuracy. To achieve optimal accuracy, the MSE must be configured correctly. (Cisco, 2012a) state that a value of -75dBm is used as the minimum signal level which must be recorded from a minimum of three AP's located on the same floor. It also states that AP's should be staggered for areas such as long narrow corridors and that there is sufficient AP density and perimeter coverage. To help identify correct locations for AP placement and density, the WCS Planning Tool can be used.

6. CONCLUSION

The objective was to investigate if the Cisco Context-Aware Mobility provided a viable solution to the ever growing needs of enterprises, universities and shopping centres to name but a few, by providing a high accuracy RTLS using Wi-Fi. GPS's poor performance indoors has placed greater emphasis on indoor positioning based on WLAN 802.11. Indoor positioning based on WLAN can provide the benefits of being cost effective and easy to implement as it makes use of an already existing infrastructure. This paper includes a Survey chapter which discussed the different Wi-Fi measuring techniques that can be used to determine one's location. The survey also looked at the Cisco Context-Aware Mobility and looked at how it could be implemented using the Context-Aware API. The Cisco Context-Aware Mobility makes use of the Cisco RF Fingerprinting technique to measure Wi-Fi signals. It uses RSSI and works in the same way as Location Patterning, however, Cisco state that it is more efficient, quicker and provides better accuracy than Location Patterning. This is achieved by building a RF propagation model using data collected from calibration phase, which calculates path loss and shadow fading deviation to account for propagation discrepancies. Cisco RF Fingerprinting does not require the same calibration effort as Location Patterning nor does it require re-calibrating as often as Location Patterning. It can also make use of pre-package RF models for similar environments.

Cisco's Context-Aware Mobility Service is used to capture information on mobile assets including devices, people or products. The information captured can include location, temperature, availability and applications used. The Context Aware Mobility runs on the MSE. The MSE communicate with WLAN Controllers (WLC) and the WCS, which is a WLAN management tool that provides the ability to manage network controllers through a web based user interface. The MSE provides real time location tracking and historical data of Wi-Fi clients, RFID tags, and rogue device. Applications can communicate with the MSE through APIs such as the Context-Aware API using XML/SOAP. With the ever increasing popularity of mobile devices, an Android application was developed to determine if the Context-Aware Mobility was capable of providing a high accuracy RTLS, that could potentially be used by a large number of students/staff. The app was designed to be used within the main building of the campus, to allow for simple interaction with the college map and that could be easily deployed onto a mobile device. The implementation of an Android application made use of the Context-Aware API through the means of a web service. Using the API, the Android application was able to retrieve location details from the college MSE and use them to display the mobile devices position within the college on a map. The Context-Aware API provides a vast array of functionality, many which have not been implemented in the developed software artefact. These include the ability to retrieve location history for a single mobile device, location history for multiple mobile devices, network design information and notifications based on location to name but a few.

Testing was carried out in the main building with one hundred sample points collected, covering ten different locations which were taken from three different floors. Three key elements were noted during testing: if the device was detected on the correct level, how often was it detected by the same AP and finally what was its distance from the actual sample point. The response time of the MSE was deemed to be slow as each response took at least ten seconds or more. This could be a critical issue in a live system as it would have a large impact on the accuracy of the system. In order to retrieve an accurate representation of the Cisco system, sufficient time was given at each sample point, allowing the application to retrieve a response from the MSE which was relative to the mobile devices location. This resulted in the tests not taking into account the active movement of the mobile device when the location request was made to the MSE and this is an area that would need to be investigated further. The results indicated that of the one hundred sample points, only forty contained a 100% success rate of being detected on the correct floor. Only one location had a 100% success rate of being detected by the same AP, with a further five locations having a success rate of 80%. The distance accuracy of the tests returned an average of 7m – 45m with 2.7m the best recorded distance of all one hundred sample points. The level of network activity within the college surprisingly had no bearing on the accuracy of tests as the results were just as good and better in some instance when there was a high level of activity.

It was concluded that the density of APs in the college is sufficient for 100% data coverage, but not sufficient enough to enable high positioning accuracy. To obtain optimal accuracy, the device must be detected by four or more APs. Therefore, an AP must be placed every 12 – 20 linear meters or one AP every 230 – 450 square meters. In order to fully test the quality of the Cisco Context-Aware Mobility, a full quality assurance and validation process would be required. This would involve ensuring that all APs correctly match up to the MSE, that all heat maps are correct and a walk test of the campus to ensure that every location in the building is visible by multiple APs. It would also require the MSE to be re-calibrated to account for changes to the building which can affect RF propagation. To ensure that these steps are carried out to a sufficient standard the WCS Planning Tool should be used to identify correct locations for AP placement and also the WCS accuracy should be used to generate accuracy reports for the specified reference points used during testing.

REFERENCES

Abdul-Latif, O., Sheperd, P., & Pennock, S. (2007). *TDOA/AOA Data Fusion for Enhancing Positioning in an Ultra Wideband System* (pp. 1531–1534). Signal Processing and Communications. doi:10.1109/ICSPC.2007.4728623

Apache. (2012). *Jax-WS Guide*. Retrieved from http://axis.apache.org/axis2/java/core/docs/jaxws-guide.html

Atia, M. M., Korenberg, M., & Noureldin, A. (2012). A Consistent Zero-Configuration GPS-Like Indoor Positioning System Based on Signal Strength in IEEE 802.11 Networks. Proceedings of the Position Locarion and Navigation Symposium (pp. 1068-1073). doi:10.1109/PLANS.2012.6236849

Brown, D. R., & Dunn, D. B. (2011). Classification Schemes of Positioning Technologies for Indoor Navigation. *Proceedings of IEEE Southeastcon* (pp. 125-130). doi:10.1109/SECON.2011.5752919

Chang, N., Rashidzadeh, R., & Ahmadi, M. (2009). *Differential Access Points for Indoor Location Estimation*. Electro/Information Technology. doi:10.1109/EIT.2009.5189622

Chang, N., Rashidzadeh, R., & Ahmadi, M. (2010). Robust Indoor Positioning using Differential Wi-Fi Access Points. *Consumer Electronics*, *56*(3), 1860–1867. doi:10.1109/TCE.2010.5606338

Ching, W., Teh, R. J., Li, B., & Rizos, C. (2010). Uniwide WiFI Based Positioning System. *Proceedings of the International Symposium on Technology and Society* (pp. 180-189). doi:10.1109/ISTAS.2010.5514639

Chon, Y., & Cha, H. (2011). *LifeMAp: A Smartphone-Based Context Provider for Location-Based Services.* In Pervasive Computing (pp. 58–67).

Chun, S. M., Lee, S. M., Nah, J. W., Choi, J. H., & Park, J. T. (2011). *Localization of Wi-Fi Access Point using Smartphone's GPS Information.* In Mobile and Wireless Networking (pp. 121–126).

Cisco. (2006). *Wi-Fi Location-Based Services—Design and Deployment Considerations*.

Cisco. (2008a). Cisco Location Based Services Architecture', in *Wi-Fi Location Based Services Design Guide 4.1*.

Cisco. (2008b). Location Tracking Approaches', in *Wi-Fi Location Based Services Design Guide 4.1.*

Cisco. (2008c). *The Technologies behind a Context-Aware Mobility Solution*.

Cisco. (2010a). Architectural Overview, in *Cisco Context Aware Mobility API*.

Cisco. (2010b). Benefits. In *Cisco Conext Aware Mobility API*.

Cisco. (2010c). *Cisco Context Aware Mobility Solution Technology*.

Cisco. (2010d). Introduction to Context-Aware APIs. In *Cisco Context-Aware API Getting Started Guide*.

Cisco. (2011). Cisco Wireless Control System Configuration Guide, Release 7.0.172.0.

Cisco. (2012a). *Cisco Mobility Services Engine - Context Aware Mobility Solution Deployment Guide*.

Cisco. (2012b). Cisco MSE API Specification Guide-Context Aware Services of MSE, Release 7.2.

Cisco. (2012c). Overview. In *Cisco Context-Aware Service Configuration Guide, Release 7.3*.

Ciurana, M., Giustiniano, D., Neira, A., Barcelo-Arroyo, F., & Martin-Escalona, I. (2010). *Performance stability of software ToA-based ranging WLAN. In* Indoor Positioning and Indoor Navigation (pp. 1–8). doi:10.1109/IPIN.2010.5648117

Cypriani, M., Lassabe, F., Canalda, P., & Spies, F. (2009). *Open Wireless Positioning System: A Wi-Fi-Based Indoor Positioning System*. In Vehicular Technology Conference Fall (pp. 1–5).

Cypriani, M., Lassabe, F., Canalda, P., & Spies, F. (2010). *Wi-Fi-Based Indoor Positioning: Basic Techniques, Hybrid Algorithms and Open Software Platform*. In Indoor Positioning and Indoor Navigation (pp. 1–10).

Del Mundo, L. B., Ansay, R. L., Festin, C. A., & Ocampo, R. M. (2011). *A Comparison of Wireless Fidelity (Wi-Fi) Fingerprinting Techniques.* In ICT Convergance (pp. 20–25). doi:10.1109/ICTC.2011.6082543

Drane, C., Mac Naughton, M., & Scott, C. (1998). Positioning GSM Telephones. *IEEE Communications Magazine*, *36*(4), 46–54, 59. doi:10.1109/35.667413

Driscoll, C. J. (2013). *Access Science*. Retrieved from http://www.accessscience.com/loadBinary.aspx?filename=YB001541FG0010.gif

Elkamchouchi, H., & Mofeed, M. A. (2005). *Direction-Of-Arrival Methods (DOA) and Time Difference of Arrival (TDOA) Position Location. Proceedings of the* Radio Science Conference (pp. 173–182).

Etutorials.org. (2013). Retrieved from http://etutorials.org/shared/images/tutorials/tutorial_158/fig436_01.jpg

Fang, S. H., & Wang, C. H. (2011). *A Dynamic Hybrid Projection Approach for Improved Wi-Fi Location Fingerprinting. In* Vehicular Technology (pp. 1–6).

Gallagher, T., Li, B., Dempster, A.G., & Rizos, C. (2010). Database Updating Through User Feedback in Fingerprint-Based Wi-Fi Location Systems. In *Ubiquitous Positioning Indoor Navigation and Location Based Service* (pp. 1-8).

Google. (2013). *AsyncTask.* Retrieved from http://developer.android.com/reference/android/os/AsyncTask.html

Google. (2013). Billing Best Practices. Retrieved from http://developer.android.com/google/play/billing/billing_best_practices.html

Hossain, M., Van, H. N., Jin, Y., & Soh, W.-S. (2007). *Indoor Localization Using Multiple Wireless Technologies. In* Mobile Adhoc and Sensor Systems (pp. 1–8). doi:10.1109/MOBHOC.2007.4428622

IBM. (2013). *Securing web services applications at the transport level.* Retrieved from http://pic.dhe.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=%2Fcom.ibm.websphere.express.doc%2Finfo%2Fexp%2Fae%2Ftwbs_secwsaatl.html

Jagoe, A. (2002). *Mobile Location Services: The Definitive Guide*. Prentice Hall PTR.

Kerner, S. M. (2012, November 15). *Enterprise Networking Planet*. Retrieved from http://www.enterprisenetworkingplanet.com/nethub/indoor-gps-cisco-debuts-new-location-awareness-technology.html

Kim, J. I., Lee, J. G., & Park, C. G. (2008). A Mitigation of Line-of Sight by TDOA Error Modelling In Woreless Communication System. *Proceedings of theInternational Conference on Control, Automation and Systems* (pp. 1601-1605).

Kim, Y., Chon, Y., & Cha, H. (2012). Smartphone-Based Collaboative and Autonomous Radio Fingerprinting. *Systems, MAn and Cybernetics, Part C: Applications and Reviews*, *42*(1), 112–122. doi:10.1109/TSMCC.2010.2093516

Kim, Y., Shin, H., & Cha, H. (2012). *Smartphone-based Wi-Fi Pedestrian-Tracking System Tolerating the RSS Variance Problem.* In Pervasive Computing and Communications (pp. 11–19). doi:10.1109/PerCom.2012.6199844

Koo, J., & Cha, H. (2012). Unsupervised Locating of Wi-Fi Access Points Using Smartphones. *Systems, Man and Cybernetics, Part C: Applications and Reviews*, *42*(1), 1341–1353. doi:10.1109/TSMCC.2012.2186800

Kos, T., Grgic, M., & Kitarovic, J. (2007). Location Technologies for Mobile Networks. *Proceedings of theInternational Workshop on Systems, Signals and Image Processing* (pp. 319-322).

Lawson, S. (2013, May 22). *Info World*. Retrieved from http://www.infoworld.com/t/mobile-technology/growing-mobile-malware-threat-swirls-mostly-around-android-219147?source=IFWNLE_nlt_sec_2013-05-23

Le, T. D., Le, H. M., Nguyen, N. Q., Tran, D., & Nguyen, N. T. (2011). *Convert Wi-Fi Signals for Fingerprint Localization Algorithm. In* Wireless Communications, Networking and Mobile Computing (pp. 1–5). doi:10.1109/wicom.2011.6040625

Leu, J. S., & Tzeng, H. J. (2012). Received Signal Strength Fingerprint and Footprint Assisted Indoor Positioning BAsed on Ambient Wi-Fi Signals. *Proceedings of theVEHICULAR Technology Conference* (pp. 1-5). doi:10.1109/VETECS.2012.6239883

Liu, H. H., & Yang, Y. N. (2011). *WiFi-Based Indoor Positioning for Multi-Floor Environment*. TENCON.

Llombart, M., Ciurana, M., & Barcelo-Arroyo, F. (2008). On the scalability of a novel WLAN positioning system based on time of arrival measurements. Proceedings of the Workshop on Positioning, Navigation and Communication (pp. 15-21). doi:10.1109/WPNC.2008.4510352

Lui, G., Gallagher, T., Li, B., Dempster, A. G., & Rizos, C. (2011). *Differences in RSSI Readings Made by Different Wi-Fi Chipsets: A Limitation of WLAN Localization. In* Localization and GNSS (pp. 53–57).

Meng, W., Xiao, W., Ni, W., & Xie, L. (2011). *Secure and Robust Wi-Fi Fingerprinting Indoor Localization. In* Indoor Positioning and Indoor Navigation (pp. 1–7). doi:10.1109/IPIN.2011.6071908

Morgan, Y. (2009). *Accurate Positioning Using Short-Range Communications. In* Ultra Modern Telecommunications & Workshops (pp. 1–7).

Niculescu, D. S., & Nath, B. (2003). *Localized Positioning in Ad Hoc Networks. In* Sensor Network Protocols and Applications (pp. 42–50). doi:10.1109/SNPA.2003.1203355

Powell, A. (2009). *Android Developers Blog*. Retrieved from http://android-developers.blogspot.ie/2010/06/making-sense-of-multitouch.html

Retscher, G., & Fu, Q. (2010). Continuous Indoor Navigation with RFID and INS. *Proceedings of thePosition Location and Navigation Symposium* (pp. 102-112). doi:10.1109/PLANS.2010.5507242

Shin, B. J., Lee, K. W., Choi, S. H., Kim, J. Y., Lee, W. J., & Kim, H. S. (2010). *Indoor WiFi Positioning System for Android-based Smartphone.* In Information and Communication Technology Convergence (pp. 319–320).

Shin, H., & Hojung, C. (2010). *Wi-Fi Fingerprints-Based Topological MAp Building for Indoor User Tracking.* In Embedded and Real-Time Computing Systems and Applications (pp. 105–113). doi:10.1109/RTCSA.2010.23

Shu, X., Du, Z., & Chen, R. (2009). *Research on Mobile Location Service Design Based on Android. In* Wireless Communications, Networking and Mobile Computing (pp. 1–4). doi:10.1109/WICOM.2009.5302615

Wong, C., Klukas, R., & Messier, G. (2008). Using WLAN Infrastructure for Angle-of Arrival Indoor ser Location. *Proceedings of theVehicular Technology Conference* (pp. 1-5).

Xiao, W., Ni, W., & Toh, Y. K. (2011). *Integrated Wi-FI Fingerprinting and Inertial Sensing for Indoor Positioning. In* Indoor Positioning and Indoor Navigation (pp. 1–6). doi:10.1109/IPIN.2011.6071921

Yu, Y., He, J., Wang, Q., Liu, F., & Xu, C. (2012). *A Query-driven Indoor Location System based on Smartphone. In* Enabling Technologies for Smartphone and Internet of Things (pp. 25–29).

Eugene Ferry completed a MSc in Computing at the Letterkenny Institute of Technology in 2014. He is now working in industry. His research interests include networking and location awareness.

John B. O'Raw was born in London, England, in 1965. He has lectured in Computer Science at Letterkenny Institute of Technology, Ireland since 2010. Prior to this he worked as an IT manager (14 years) and in industrial process automation and information systems, primarily based in Milan, Italy (8 years). John holds a BSc in Information Technology, a Diploma in Electronics and separate Master's Degrees in Applied Computing and Maritime Archaeology. He is currently studying for his Ph.D. At Queen's University Belfast and his research interests are in wireless, security and critical infrastructure".

Kevin Curran is a Reader in Computer Science and group leader for the Ambient Intelligence Research Group. Dr Curran has made significant contributions to advancing the knowledge of computer networking evidenced by over 800 published works. He is a regular contributor to BBC radio & TV news in the UK and quoted in trade and consumer IT magazines on a regular basis. He is an IEEE Technical Expert for Security and a member of the EPSRC Peer Review College.