A Privacy Conscious Bluetooth Infrastructure for Location Aware Computing

Albert Huang, Larry Rudolph
MIT Computer Science and Artificial Intelligence Laboratory

***Abstract*— We present a low cost and easily deployed infrastructure for location aware computing that is built using standard Bluetooth® technologies and personal computers. Mobile devices are able to determine their location to room-level granularity with existing bluetooth technology, and to even greater resolution with the use of the recently adopted bluetooth 1.2 specification, all while maintaining complete anonymity. Various techniques for improving the speed and resolution of the system are described, along with their tradeoffs in privacy. The system is trivial to implement on a large scale - our network covering 5,000 square meters was deployed by a single student over the course of a few days at a cost of less than US$1,000.**

***Index Terms*— bluetooth, location aware computing, privacy**

I. Introduction

A LOW cost and easy to deploy location awareness infrastructure requires a fast and reliable method to find nearby devices. Location aware computing provides applications with knowledge of the physical location where the computation is taking place, allowing applications to operate in a more context-sensitive fashion. However, to date, the infrastructure is expensive and difficult to deploy. Bluetooth is a stable, inexpensive, and mature technology upon which a location aware infrastructure can be built, except for the fact that naive scanning algorithms take too long and are unreliable. We present an adaptive scanning algorithm that overcomes these problems.

Fundamental to the task of location aware computing is determining the location of the computational device. In outdoor environments with unobstructed views, GPS [1] is ideal except that handheld GPS devices are still fairly expensive, but the technology is mature and the price continues to drop. Indoors, and in crowded city streets, however, the effectiveness of GPS is greatly diminished. A number of approaches have been made towards indoor localization, with varying features and measures of success, but none is currently easy to deploy, use, and inexpensive.

We present a system that provides an infrastructure that relies on technology that is already widely available and in use today. The hardware is multi-purpose and can be used for a variety of other computing tasks when not being used for positioning purposes. Many potential users of our system would not need a significant investment in capital or other resources to take advantage of our infrastructure. Additionally, the system is simple and almost trivial to deploy on a large scale.

We propose placing bluetooth USB devices in existing PC's at key locations throughout a building, turning them into location beacons. We exploit the fact that most indoor spaces, especially in work related industries, already have computers installed throughout the environment. The user is equipped with a bluetooth-enabled cell phone or PDA mobile device, which scans the environment for the location beacons. When a device is within 10 meters, the location beacons respond, thereby providing room-level location accuracy.

We argue that bluetooth is superior to other candidate technologies such as IrDA [2], IEEE 802.11 [3], and RFID [4]. IrDA is not suitable due to its directional nature and intolerance of optical obstructions, although it has its uses in location awareness, as demonstrated by Cooltown [5]. 802.11 has also been used for location awareness. One such project [6] uses Bayesian inference techniques in an area dense with wireless base stations to achieve 1 meter resolution, but requires rigorous training of the device at every possible location before the system is usable. With the restriction that a locator should not reveal its identity, an 802.11 device used in our context would not be able to use signal strength to aid its localization tasks, significantly reducing its resolution. Additionally, 802.11 draws more power than bluetooth. RFID tags are inexpensive passive devices often used in retail stores as electronic bar codes, and in identification cards to replace magnetic strips. They can be read by the more expensive RFID tag readers from varying distances. If tags were scattered throughout the premises of a building, then a mobile reader could determine its location by reading nearby tags. RFID readers are currently prohibitively expensive, however, and are also unable to provide positioning information more precise than the location of detected tags.

Several systems have been developed for indoor positioning using special purpose hardware that is more accurate than bluetooth but an order of magnitude more expensive as well. Both the Active Bat System [7], [8] and Cricket [9] use a combination of ultrasonic pulses and radio signals to provide short range location information. Since ultrasound does not pass through walls, glass or other partitions, these systems adhere to our human notion of space and locality. Active bats transmit their name to the base station which computes their location while in the Cricket system, it is the base station that transmits its name and the cricket deduces its own location; Cricket preserves privacy.

The Local Positioning Profile(LPP) [10] defines a standardized protocol for bluetooth devices to exchange positioning data. A device whose location is known runs a Local Positioning (LP) Server, to which other bluetooth devices can connect. LP Clients can request positioning information from

LP Servers, which may be derived from preset configurations, GPS data, cellular data, or automatically generated, and then infer their own positions. The primary purpose of the LPP is to provide a means for devices to exchange data, and leaves much room for techniques to be developed for determining a device's actual position given the position of other devices. The LPP also does not take privacy into consideration, as it is required for both client and server to have knowledge of both bluetooth device addresses, allowing a well-coordinated network of LP servers to track clients as they issue requests.

We are not the first to propose bluetooth as a location tracking infrastructure. Anastasi et al[11] used statically positioned bluetooth devices to constantly scan for other bluetooth devices in the vicinity. Detected devices were then entered into a central database which was used to track the location of all moving bluetooth devices. While this approach is as cost effective as ours, it allows for no privacy, requires special software on the trackers as well as their connectivity to a centralized database.

The rest of the paper is organized as follows: Section II describes the deployment of our system and techniques for detecting location beacons. Section III describes algorithms for positioning a device once beacons have been detected. An evaluation is given in section IV. Privacy issues are discussed in section V and the paper concludes with section VI.

II. Implementation and Scanning Algorithm

In this section, we describe the deployment of our infrastructure and analyze the basic technique for scanning for and discovering location beacons. We also show that using multiple co-located bluetooth devices improves the reliability and robustness of our system.

A. Deployment

Thirty D-Link DBT-120 USB Bluetooth Adapters (firmware version 1.4.2.10) were used as beacons. Research groups in our building were asked to spare a single USB port in their computers. Our beacons were then placed in computers approximately every 10 meters on six different floors. The only software installed on the hosting machines were the device drivers. On average, configuring a machine to host one of our beacons took less than three minutes. The most time consuming part of the deployment was actually tracking down the system administrators for the machines we wanted to use, and obtaining their permission.

Client software was loaded onto the locator device. A Linux client was used on laptops, desktops, and a (Linux) HP iPAQ 5550. A C++ Symbian client was used on Nokia 6600 cellular phones.

B. Naive Scanning

The locator scans for beacons, determines the location of detected beacons, computes its location relative to the detected beacons, and using information from the previous two steps, determines its absolute position.

A bluetooth device inquiry, which is a broadcast of a predefined sequence of bits while hopping channels pseudo-randomly, is used to detect nearby beacons. The locator does not reveal its own identity during an inquiry. Upon hearing a response, the locator must determine if the responder is a beacon in the locator network, or if it's merely another bluetooth device.

The ideal beacon would always listen for the inquiry sequence and respond almost immediately upon detection. A number of factors can cause a beacon to either not respond or to not detect an inquiry.

- Electromagnetic noise and interference with other devices in the 2.4 GHz range may hinder communications.
- A beacon cannot listen for an inquiry all the time. It must allocate time to listen for connection requests, and to participate in active connections.
- Upon first detecting a device inquiry, a beacon will always enter a backoff stage, in which it idles for 0 to 0.33 seconds randomly.
- A beacon, while listening for inquiries, will listen on one of 32 predefined channels at a time. During an inquiry, the locator will inquire on half of these channels for 2.56 seconds, switch to the other half for another 2.56 seconds, and then alternate two more times. Consequently, it is possible that a locator will not even inquire on the same channel on which a beacon is listening on for at least 2.56 seconds.

While nothing can be done about noise and interference from other radio sources[1], something can be done to improve the beacon detection speed. As can be seen from Figure 1, it can take 10 seconds for a locator to detect a beacon, and we have observed times when it has taken even longer.

The bluetooth specification is optimized for the situation where many devices are all in the same vicinity. Device inquiry is especially slow because of the pessimistic backoff algorithms used to minimize collisions. The recommended duration for a device inquiry is 10.24 seconds[12], which is clearly longer than many applications can tolerate.

C. Two heads are better than one

To reduce the average time to detect a beacon, we placed two bluetooth USB devices in the PC. The locator needs to wait for a response from only one beacon. Our experiments show that the beacons responded independently of each other, providing an ideal increase in response rate. The locator was placed approximately 8 meters from two co-located beacons, with a closed door, some wooden office furniture, and a metal filing cabinet in between the locator and the beacons. Additionally, a host of other active bluetooth and WiFi devices were operating in the vicinity. By adding a bluetooth device to the beacon, we significantly increased its tolerance for noise and interference. These results are shown in Figure 2.

[1]This is a problem addressed in version 1.2 of the Bluetooth specification, which allows for adaptive frequency hopping to avoid channels being used by co-located, interfering devices. However, existing bluetooth 1.1 and 1.0 devices are not able to take advantage of this ability.

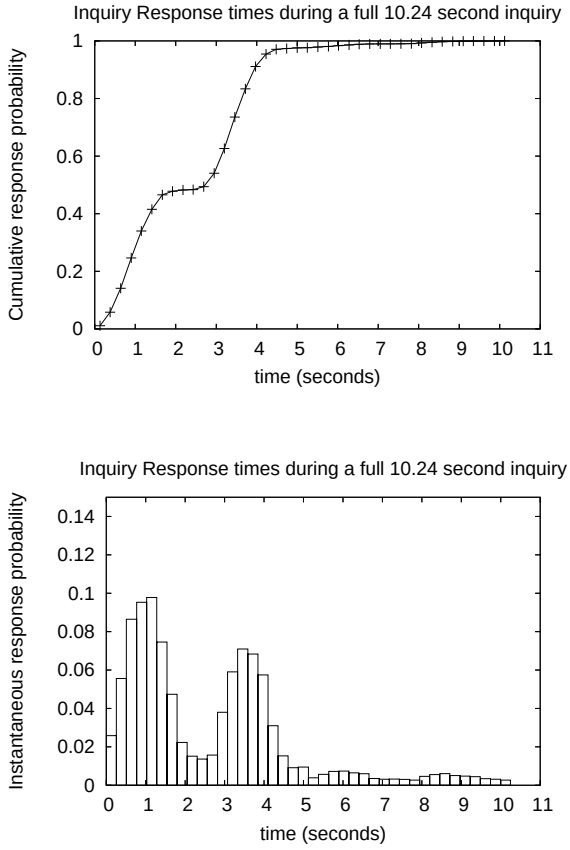

Fig. 1. The locator divides the 32 inquiry channels into two disjoint sets of channels, say S and T, If a beacon happens to be listening on a channel in S, then it will likely be discovered in the first 2.56 seconds. Otherwise, it will not be discovered at least until the locator switches to set T. The top graph shows the cumulative success while the bottom graph shows the instantaneous success.

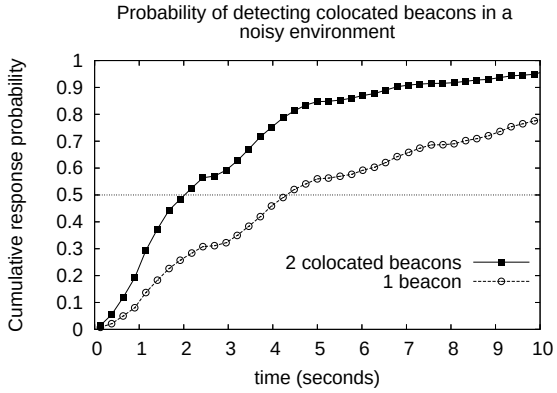

Fig. 2. Placing two beacons in one PC makes it much more responsive than a PC with only one beacon. On average, a PC with two beacons responded to an inquiry in 2.02 seconds, while a PC with only one beacon took 4.31 seconds to respond.

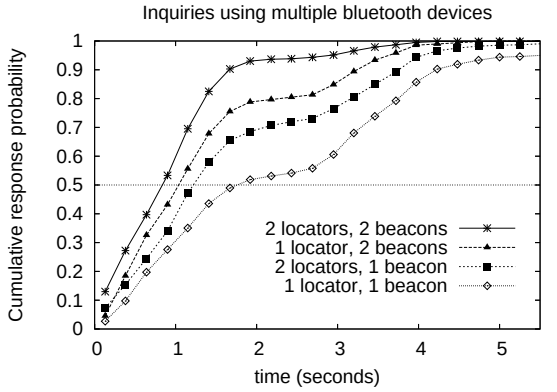

Fig. 3. When a locator is equipped with multiple bluetooth devices and uses them to perform simultaneous inquiries, devices in range are detected much more rapidly. Using two co-located beacons in addition to two bluetooth devices on the locator is even faster.

Similarly, when the locator was equipped with two bluetooth devices, and performed inquiries with both devices simultaneously, location beacons were also discovered more quickly. These results are summarized in Figure 3. In order to achieve the improved response rate, however, the discoverability of the locator's bluetooth devices had to be disabled, otherwise, performance actually decreased as the locator began responding to its own inquiries. Response rate of a single bluetooth device in range was still not as fast as when a beacon is equipped with two bluetooth devices and the locator with one. We attribute this to the backoff algorithm used during the inquiry scan process.

III. Determining location

The locator maintains a lookup table, mapping beacon identities (bluetooth device addresses) to locations. Upon detecting a beacon during an inquiry, the locator checks the lookup table for a mapping. If no mapping exists, then the locator can query the beacon for its location at the expense of anonymity.

There is one situation in which information can be sent from a device B (the beacon) to a device L (the locator) without B having knowledge of L's identity. This is when B is responding to an inquiry made by L, and no further communication between the two devices has taken place. Thus once a beacon has responded to an inquiry with its device address, L is able to determine its location with the help of the lookup table.

If a beacon B is detected that does not have an entry in the locator's lookup table, and the locator is willing to reveal its identity, then it can establish a higher level bluetooth connection with B and request more information, such as its location. We chose to embed a beacon's location inside its bluetooth friendly name, so that a locator need only issue a remote name request to determine the beacon's location. For example, B could be given the name "OKN-32-305" to signify that it is in building #32, room 305.[2] "OKN" is used here as a prefix to distinguish our beacons from other bluetooth devices.

[2]This is exactly what LPP[10] is designed to do - provide a standardized method for the transfer of positioning information from the beacon to the locator. However, at the time of writing, LPP was still in draft form and we found our method much simpler.

A. Determining locator position relative to beacons

Once a locator has detected one or more beacons, it can take the intersection of the areas covered by detected beacons to determine its approximate location. Thus, the precision with which a locator can determine its position is directly related to the number of beacons it detects. Figure 4 illustrates this principle.

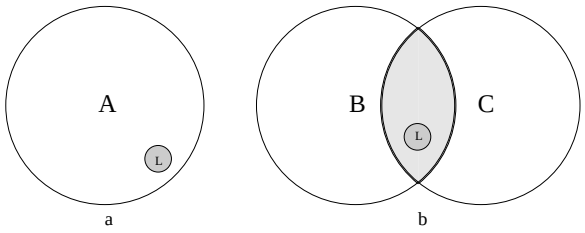

Fig. 4. a) When the locator L can only detect one beacon, it can only conclude that it is somewhere within the circle b) When two beacons are detected, much greater resolution is achievable, and the locator can conclude it is somewhere in the shaded region

The Bluetooth 1.2 specification[13] supports device inquiries that report signal strength of discovered devices. Most bluetooth devices do not yet support this feature. Currently, signal strength of another bluetooth device can only be determined after a higher level connection has been established and identities revealed.

B. Signal strength and Link Quality

In the absence of radio noise and obstruction from objects such as wood, metal, and people, the quality of a link between two bluetooth devices is inversely related to their distance. The `HCI_Read_Link_Quality` command determines the quality of a bluetooth connection with another device. We were unable to use link quality to determine exact distance from a beacon, but we could use it to establish a rough upper bound. The use of link quality can be used to resolve on which floor of a building is the locator, with the assumption that signals received from beacons on other floors are much weaker than signals received from beacons on the same floor.

Information about link quality with another device is not supported on all bluetooth devices, and the method for calculating link quality is device specific. In our limited experience, signal strength and link quality is not available on HP iPAQs, is available on D-Link USB bluetooth devices, and is available on Nokia Series 60 cell phones with the signature of a non-disclosure agreement.

Devices may also support the `HCI_Read_RSSI` command, to obtain direct information about the signal strength of a connected device. In our testing with D-Link DBT-120 USB bluetooth devices, we found link quality to be more closely related to distance than RSSI. Nilsson and Hallberg[14] found that signal strength was poorly correlated with distance.

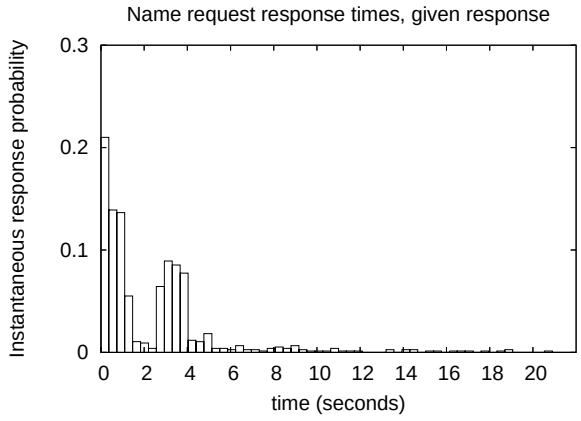

Fig. 5. The name request process is similar in nature to the inquiry process, but uses a different set of 32 channels. If the name of a device is resolved, it is usually done so during the first 5.12 seconds of the name request - the amount of time it takes for the locator to iterate through both trains A and B.

C. Early Timeout

When detected bluetooth devices are not recognized, a name request is used to determine their locations[3]. This entails paging every discovered device, a potentially time consuming process. If a full length inquiry is performed, followed by issuing a name request to each beacon one by one, the cumulative time spent becomes prohibitively expensive. A locator equipped with multiple bluetooth devices can issue these name requests in parallel, significantly speeding up this process.

Figure 1 indicates that the majority of bluetooth devices are discovered early on during an inquiry. In our measurements, 53% of bluetooth devices discovered during an inquiry were discovered in the first 2.56 seconds.

The bluetooth specification recommends 5.12 seconds as the timeout when paging[4] a remote device. Since paging is the most time consuming part of a name request, limiting the page effectively limits the name request, but also lowers the likelihood of a successful page. To see if extending the page timeout significantly increased the chance of resolving the name of a device, we performed numerous remote name requests with the page timeout set to 20.48 seconds. Figure 5 shows that if the name of a remote device was resolved during the 20.48 second time period, it was resolved in the first 5.12 seconds 87% of the time. Note that some bluetooth implementations, such as BlueZ for Linux, raise the default timeout significantly to increase the chance of successfully paging on the first try.

These observations suggest the following algorithm for determining the position the locator as quickly as possible when in the presence of unknown bluetooth devices.

Scanning Algorithm: *The locator performs a device inquiry for 2.56 seconds, during which time any unrecognized devices are queued for name resolution. For each unrecognized*

[3]If the user desires absolute anonymity, then this step shouldn't be taken, as it reveals the locator's identity.

[4]Note that the specification recommends a time period twice as long for inquiries.

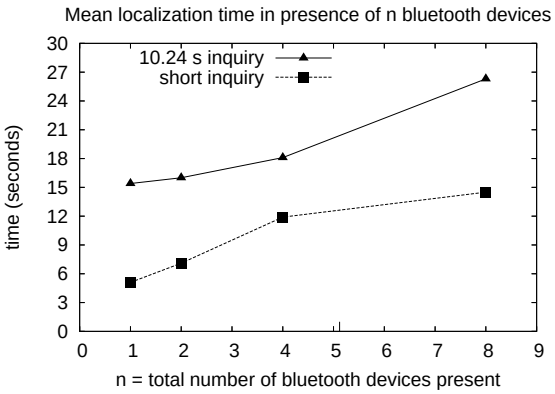

Fig. 6. By inquiring for 2.56 s, then stopping a device inquiry as soon as a device is discovered to resolve its name, we cut the expected time to determine the position of the locator in half.

device, the locator tries to resolve its name by issuing a remote name request for 5.12 seconds. When all the unrecognized devices have been queried for their names, the locator repeats its inquiry.

Figure 6 shows that after 2.56 seconds of an inquiry, cancelling to immediately resolve the names of unrecognized devices is much quicker to determine the position of the locator.

IV. Evaluation

To evaluate our system and the different positioning techniques, three locators were programmed with different positioning algorithms:

Locator A Perform a device inquiry for 10.24 seconds. Remote name requests are then issued to discovered bluetooth devices. Responses are cached in software. Name requests time out after 5.12 seconds. Once all name requests have been issued, the algorithm repeats.

Locator B Perform a device inquiry for 2.56 seconds. After this, the inquiry is canceled as soon as a device is discovered whose name is not in the software cache. Remote name requests are sent to these devices, and the responses cached. If no such device is discovered, the inquiry continues for at most 7.68 more seconds. The algorithm then repeats.

Locator C A software cache containing all known bluetooth beacons in the building is is preloaded into memory. The locator repeatedly performs device inquiries, and never issues any remote name requests. Unrecognized bluetooth devices are ignored.

For each of the three methods, whenever a cached beacon is discovered, or a name request reveals a beacon, the locator concludes it is within 10 meters of the beacon[5]. If a beacon is not heard from in 15 seconds, then the locator assumes the beacon is no longer in range.

[5]Even this isn't entirely correct, as we have observed ranges of 25 meters when a clear line of sight between two bluetooth devices is present. In an office environment, however, this is a rare situation, and the 10 meter limit is almost always sufficient

locator	only to detect	first	second	third
A	0	2	3	16
B	0	3	19	2
C	10	20	3	1

TABLE I

Relative speed in detecting nearby beacons

Three locators were carried around the building for forty minutes, collecting localization data. Our results, summarized in Table I, show that method C was by far the fastest. Out of 34 beacons detected by locator C, it was either the first or only locator to detect the beacons 30 times. Locator B consistently detected beacons faster than A, but slower than C, and locator A was usually the last to detect a beacon.

For each beacon that a locator discovered, we averaged the difference between the time that it was discovered and the time that it was first discovered by any of the other two locators. We found that on average, locator A was 19.5 seconds slower to detect a beacon than the first locator (not necessarily C), locator B was 8.5 seconds slower than the first locator, and locator C had only a 0.9 second delay on average.

Locator C had the advantage of not needing to perform any remote name requests at all. In areas dense with bluetooth devices, the algorithm could safely ignore unrecognized devices. Additionally, as it doesn't establish any link-level connections, locator C can guarantee itself complete anonymity. The only disadvantage is that the software cache must be obtained from somewhere else. We believe this is not a significant disadvantage, as a simple text file posted on a web site would be sufficient.

We find that locator B is useful in situations where positioning is desired in an unfamiliar environment, where the software cache for locator C could not be updated before entering the area. In a known environment, however, locator C is faster in all respects. In no circumstances is locator A to be preferred.

V. Privacy

Key to tracking a bluetooth device is its discoverability mode. If the device is set to be discoverable, then whenever another bluetooth device broadcasts an inquiry message, a response is sent back, identifying the device. If a coordinated network of devices is used to periodically issue these inquiries and record the responses, then discoverable devices can be easily tracked as they move around the environment. This is the approach taken by Anastasi et al[11].

In addition to the potential to be tracked, a host of other possibilities for abuse arises when a bluetooth device is left in discoverable mode. Unrequested advertisements[15], solicitations for sexual encounters[16], and other unwanted messages[17] could arbitrarily be sent to the user's cell phone or mobile device. While there certainly is an audience that would welcome these actions, there are also many more who would not. Many manufacturers also recognize this, and include an easily accessible option to disable the discoverability of a bluetooth device.

Devices making use of our system will periodically make inquiries of their own. While it is possible for a well-coordinated

system to track bluetooth device inquiries, the system has no way of knowing what device made each inquiry, as identifying information is not transmitted by the inquiring device. Since bluetooth also has numerous other uses, almost all of which involve making an inquiry, it is infeasible to track an individual that does not wish to be tracked. Thus, our system currently allows a user to retain their privacy, preventing other devices not under their control from discovering their location, while still being able to take advantage of our location aware services.

VI. Conclusion

We have presented a privacy conscious location aware system that is based solely on inexpensive, off the shelf components. The system is simple and easy to deploy on a large scale, provided that an existing computing infrastructure is already in place. Participants in our system would not require specialized devices, and could simply use their bluetooth enabled cell phones and PDAs. The client software needed to take advantage of our system is lightweight and easily deployed. As the infrastructure changes, clients can either obtain centralized updates, or update their cache manually by querying beacon locations in person.

With the use of bluetooth 1.1 devices, our system provides room-level granularity while retaining complete anonymity. Bluetooth 1.2 devices may utilize link quality and signal strength information to obtain even greater resolution while still remaining anonymous. If a user does not require anonymity, then a 1.1 device is sufficient for accurate, fine-grained localization.

References

[1] I. A. Getting, "The global positioning system," *IEEE Spectrum*, vol. 30, no. 12, pp. 36–47, Dec 1993.

[2] (2004, July) Irda home page. [Online]. Available: http://www.irda.org

[3] *IEEE Standard 802.11 - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHS) specifications*, The Institute of Electrical and Electronics Engineeers, Inc., 1999.

[4] (2004, July) Rf-id.com. [Online]. Available: http://www.rf-id.com/rfidtech.html

[5] T. Kindberg, J. Barton, J. Morgan, G. Becker, D. Caswell, P. Debaty, G. Gopal, M. Frid, V. Krishnan, H. Morris, J. Schettino, B. Serra, and M. Spasojevic, "People, places, things: Web presence for the real world," in *Proceedings of the 3rd IEEE Workshop on Mobile Computing Systems nd Applications*, 2000.

[6] A. M. Ladd, K. E. Bekris, A. Rudys, L. E. Kavraki, D. S. Wallach, and G. Marceau, "Robotics-based location sensing using wireless ethernet," in *Proceedings of the 8th annual international conference on Mobile computing and networking*, 2002, pp. 227–238.

[7] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application," *Wirel. Netw.*, vol. 8, no. 2/3, pp. 187–197, 2002.

[8] R. Want, A. Hopper, V. F. ao, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, Jan 1992.

[9] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 32–43.

[10] B. S. I. Group, *Local Positioning Profile, Version 0.95*, July 2003.

[11] G. Anastasi, R. Bandelloni, M. Conti, F. Demastro, E. Gregori, and G. Mainetto, "Experimenting an indoor bluetooth-based positioning service," in *Proceedings of the International Conference on Distributed Computing Systems Workshops*, May 2003, pp. 480–483.

[12] *Bluetooth Profile, Specification of the Bluetooth System, Version 1.1*, Bluetooth Special Interet Group, Feb. 2001.

[13] *Bluetooth Profile, Specification of the Bluetooth System, Version 1.2*, Bluetooth Special Interet Group, Nov. 2003.

[14] M. Nilsson and J. Hallberg, "Positioning with bluetooth, IrDA, and RFID," Master's thesis, Luelå University of Technology, 2002. [Online]. Available: http://media.sm.luth.se/publications/2002/hallberg02positioninng.pdf

[15] L. Aalto, N. Göthlin, J. Korhonen, and T. Ojala, "Bluetooth and WAP push based location-aware mobile advertising system," in *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, 2004, pp. 49–58.

[16] (2004, July) Toothing - wikipedia. [Online]. Available: http://en.wikipedia.org/wiki/Toothing

[17] (2004, July) bluejackq. bluejackQ. [Online]. Available: http://www.bluejackq.com