MAX: Wide Area Human-Centric Search of the Physical World

KOK-KIONG YAP
Stanford University
VIKRAM SRINIVASAN
Bell Labs Research, India
and
MEHUL MOTANI
National University of Singapore

We propose MAX, a system that facilitates human-centric search of the physical world. Instead of organizing objects a priori, it allows humans to search for and locate them as needed. Designed for the following objectives: (i) human-centric operation, (ii) privacy, and (iii) efficient searching of any tagged object, MAX provides location information in a form natural to humans, that is, with reference to identifiable landmarks (such as, "on the dining table") rather than precise coordinates. In the system, all physical objects—from documents to clothing—can be tagged, users then locate objects using an intuitive search interface. To make searching efficient, MAX adopts a hierarchical architecture consisting of tags (bound to objects), substations (bound to landmarks), and base-stations (bound to localities). Tags can be marked as either public or private, with private tags searchable only by the owner. MAX also provides for privacy of physical spaces. It requires minimal initial configuration, and is robust to reconfiguration of the physical space. We also present a methodology to design energy-optimal and delay-optimal query protocols for a variety of device choices, this optimizes system performance, and affords insight into the appropriate actions for various scenarios. We have implemented a simple prototype of MAX, demonstrating the feasibility of the system for human-centric search over several locations across a wide area. We contend that a MAX-like search system will enable sharing (e.g., books on a college campus) and trading (e.g., buying and selling used books) of physical resources, and will be the engine for a host of new applications.

Categories and Subject Descriptors: H.4.0 [**Information Systems Applications**]: General

General Terms: Design, Experimentation, Human Factors, Performance

Additional Key Words and Phrases: Human-centric, search, physical world, landmark-based localization

---

Authors' addresses: K.-K. Yap email: yapkke@stanford.edu; V. Srinivasan email: vikramsr@alcatel-lucent.com.

ACM Reference Format:
Yap, K.-K., Srinivasan, V., and Motani, M. 2008. MAX: Wide area human-centric search of the physical world. ACM Trans. Sens. Netw. 4, 4, Article 26 (August 2008), 34 pages. DOI = 10.1145/1387663.1387672 http://doi.acm.org/10.1145/1387663.1387672

1. INTRODUCTION

The fact that information and communication technology have revolutionized the way we live is self-evident and inarguable. One particularly relevant perspective is that many of these technologies allow humans to be more disorganized, meaning that they are able to spend less time and effort in planning and organization, without compromising on efficiency. This notion is captured by the following examples.

—Wireless communications technology allows people to meet up spontaneously, without agreeing a priori on a precise location at which to meet.
—Credit cards and automatic teller machines have eliminated the need to plan ahead to ensure access to adequate amounts of money, at home or on vacation.
—Search engines, such as Google, have become so efficient at finding information on demand, that it is no longer necessary to download and save data locally.
—Gmail allows email to be labeled and searched efficiently, eliminating the need to organize and file one's emails.

It is not hard to imagine that the ability to search the physical world will allow humans to be more disorganized with their physical belongings. It will have far reaching implications and applications similar to search engines for the virtual world. We propose MAX,[1] a system and architecture that allows physical objects to be searched for and located quickly and efficiently. Current technology trends, such as smart paints, smart textiles, smart dust, RFID tags, and HP's Memory Spot, point to a not-too-distant future in which a wide range of physical objects will be tagged with small devices that have basic processing and communication capabilities. An immediate concern of such a system is the need for privacy; this is something that will be further clarified in article.

MAX allows us to move away from the paradigm of "everything has its place" to a paradigm of "everything has a place," by bestowing upon us the ability to search the physical world. As an example, an audio CD would not need to be put back into its specific jewel case after each use; instead, a CD could be put into any case, since one could perform a search for a particular CD when one wishes to listen to it. The ability to locate objects as and when we please will allow us to spend less time and effort in organizing our physical spaces, and to use our limited space more efficiently. The search capability, combined with the ability to distinguish between private and public objects, resembles the manner

[1]MAX is short for Maxwell's Demon who attempts to create order from disorder, allegedly violating the second law of thermodynamics.

in which information is shared in the digital world. We believe that this will empower a variety of new applications that involve the sharing and trading of physical resources. As an example, expensive and hard to find books could be shared between professors and students on a university campus. In addition, used text books could merely be tagged as being for sale, saving the seller the effort of advertising them.

In this article, since we are describing a system that is designed from a human-centric point of view, we will provide information in a form easily accessible to humans. What we mean by this is that when humans locate objects, they do not do so in terms of absolute coordinates, rather they use identifiable landmarks. For example, one says "my notebook is on my desk" or "the keys are on the dining room table." This simple but powerful observation will not only create a human-centric search facility, but will also greatly simplify the engineering needed to create the system. Before delving into the detailed system architecture, we outline three design goals of MAX:

—*Human-centric operation*: An objective of the system is ease of use and installation. Central to the theme of a user-friendly system is the use of natural human language, which allows a user to interact with the system at ease. Natural human language should be accepted as input, and the results of the system should also be presented as intuitively as possible. The system must also be robust to reconfiguration of physical spaces, requiring minimal or no changes when such reconfigurations occur. For example, a person might wish to move their furniture around or move to a new apartment or office space.

—*Privacy*: A requirement of privacy is that the system should not be vulnerable to unauthorized access. Privacy involves several considerations. First, the movement and locations of privately owned objects should not be tracked. Second, objects labeled private should only be searchable by their owners, while public objects may be searched for by anyone. In terms of security of physical spaces, the system should provide for varying degrees of access to spaces by allowing them to be marked as *off-limits, private, or public*.

—*Efficient search*: The system must allow thousands of objects to be searched across wide areas (e.g., office buildings, university campuses, or even across a city). Search results must be returned quickly for a satisfactory user experience, and the search process should utilize the limited system resources such as energy, bandwidth, and memory as efficiently as possible.

Several proposals for localization of wireless tags exist in current literature. These proposals can be either classified as *location tracking systems* [Bewator 2005; Ubisense 2005], or *location support systems* [Priyantha et al. 2000]. Location tracking systems are proposed for applications such as inventory tracking. In these proposals, the system proactively keeps track of the location of the objects and stores them in a central database. There exist questions on scalability and concerns over privacy, given the fact that object locations are stored. Location support systems [Priyantha et al. 2000] allow users equipped with wireless devices to locate services in their vicinity (e.g., a printer). Additionally, in location support systems, wireless tags estimate objects' locations individually, and these locations are not stored in a central database. Therefore, these systems do provide some level of privacy, as argued in Priyantha et al. [2000]. However, neither of these systems satisfy our design goals. They do not provide a mechanism to distinguish between private and public objects, or between private and public physical spaces. In addition, they are not robust to reconfigurations of the physical space, thus violating the central theme of providing a human-centric system. In order to provide accurate location information, these systems require carefully placed beacons, and a context aware middleware layer with map servers to provide human-centric information.

We propose the architecture in this article based on two observations. First, technology trends point to a not-too-distant future in which small tags with basic processing and communications capabilities will be embedded everywhere. Second, humans are powerful sensors who require only approximate coordinates, in the form of cues and landmarks, to locate objects. These observations, though simple, allow us to design an architecture that satisfies all our design objectives. We assume that all objects can be tagged. In each tag is stored a label (or descriptor), that describes the object to which it is attached. For example, a label could be "brown coffee table" or "blue denim shirt." To search these tagged objects, we propose a hierarchical architecture, in which the objects in each level may be more mobile than the objects in the higher tiers. For example, the tag associated with a room might be at the top of the hierarchy, while the furniture in the room would be at the next level down, and all other objects within the room would be at the lowest level of the hierarchy. Whenever a search is made (e.g., "cellular phone"), the object of interest is localized with respect to objects in the higher levels of the hierarchy (e.g., "coffee table," "living room").

We have motivated MAX as a system that facilitates human-centric search of the physical world, and is based on several clearly identified design goals. For the rest of this article, we present the design, analysis, and implementation of MAX. Our specific contributions in this article are:

(1) The proposal of a simple hierarchical architecture consisting of base-stations, substations, and tags. Also, the design of a system architecture over a wide area network is outlined, to provide ubiquitous access to physical spaces.
(2) The introduction of a new notion of privacy of physical spaces, providing for privacy of both physical spaces and objects.
(3) The development of a simple prototype based on the Crossbow motes platform, and providing search functionalities of the MAX system across the Internet.
(4) The design of energy-optimal and delay-optimal query protocols for different device choices, detailing the appropriate actions to take in various situations.

Contributions 1, 2, and 3 can be found in Sections 2, 4, and 7, respectively. Contribution 4 is included in Sections 5 and 6. We end with related work in Section 8, and reflect on the system we have designed in Section 9.

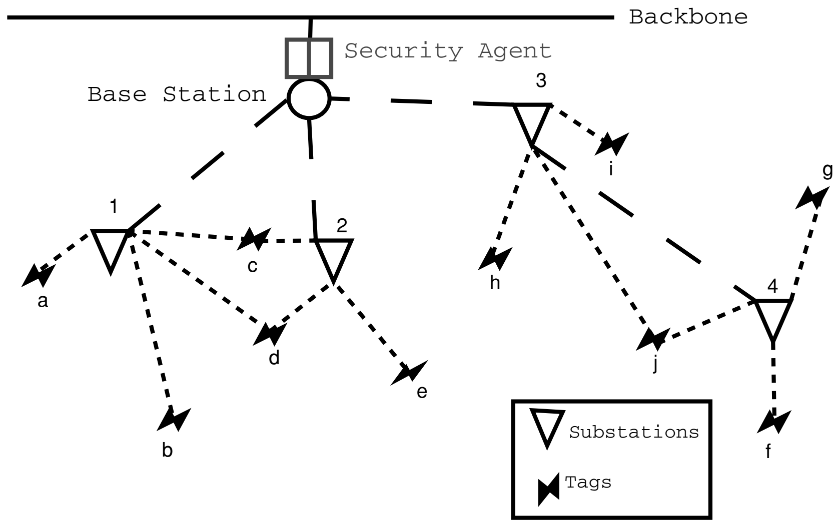


Fig. 1. The architecture of the system depicting the five elements of the architecture.

2. SYSTEM ARCHITECTURE

We will exploit our contention that humans are powerful sensors who are able to locate objects quickly based on cues and identifiable landmarks. The advantage of bringing the human into the picture is that it considerably reduces complexity, resulting in a simple and scalable architecture. For our architecture, we assume that all physical objects can have a wireless tag attached to them. Furthermore, each tag stores a description of the physical object to which it is attached. These objects can be located over a large geographical area, which we assume is divided into basic units (e.g, a room or office cubicle) called *localities*.

2.1 Architecture within a Locality

Figure 1 illustrates the architecture of MAX within a locality. It is hierarchical in nature, with different logical types of tags at each level. This three-tier hierarchy consists of base stations, which are tied to a locality; substations, which are tied to normally static objects (e.g., chairs, tables, shelves); and tags, which are tied to small mobile objects (e.g., keys, books, phone, documents). These entities are named once by the owner during initiation, and require no further maintenance by the owner. If the object changes ownership, the new owner can rename the object.

Design of this three-tier hierarchy is motivated by the way that humans organize and describe the locations of their things. Typically, people first localize an object as being in their room or office, and then describe where it is relative to easily identifiable landmarks, such as their bed or desk. In terms of ease-of-use for the consumer, having fewer device types leads to more flexibility and device reuse. With this in mind, we will discuss the four basic elements to this architecture:

(1) *Base-station*. At the highest level of the hierarchy of the local system architecture, is the base-station. It logically represents a locality that is immovable, such as a room, and it stores a descriptor of this locality (e.g., Jack's office). Depending on the size of the locality (e.g., a room), there could be one or more base-stations per locality. In addition, the base-station also acts as a gateway to the backbone network for the wireless tags. Note that the base-station logically describes objects that are static and immovable.
(2) *Substation*. At the next level of the hierarchy, is a substation, which logically describes objects that are largely static and that change positions occasionally (e.g., furniture). Substations also store the label or descriptor of the object they are attached to (e.g., coffee table).
(3) *Tags*. At the lowest level of the hierarchy are tags. Logically, these devices are attached to objects that are easily movable or highly mobile (e.g., books). Each tag stores a descriptor of the object it is attached to (e.g., Book Harry Potter). Multiple descriptor words are allowed in each tag, enabling users to label the object sensibly, so that others can locate the object based on the label. The onus lies on the owner to label sensibly if she wants the object to be found in a search.
(4) *Security agent*. The final components of the architecture are security agents, which are software agents residing at the base-stations, as illustrated in Figure 1. Users are authenticated before they are allowed to query the locality served by the base-station. Handshaking, based on public key cryptography, is used to prevent any unauthorized access of the locality, allowing the owner of the locality (and base-station) to grant and deny access as required by security. Thus localities can be made off-limits to unauthorized users, giving us the ability to protect the privacy of physical space. Security agents also allow the locality owner to mark an area as private, in which only coarse localization (e.g., whether the object is in the locality or not) is provided, facilitating sharing and sales applications.

Two other parts of the system are *writers* and *query terminals*. The tags and substations are programmed (i.e., descriptors are written into them) using the writer. We envision two ways in which a physical object can be tagged. The more likely way is for the descriptors to be programmed by the user at deployment. Alternatively, with the cooperation of manufacturers (e.g., furniture makers), objects can be embedded with a tag and a description at the time of manufacture. The query terminal is the interface through which to place queries, such as a web page with a simple search interface.

It should be noted that the naming of objects using human language, and the use of landmark-based localization have allowed the system to handle mobility gracefully. In such a system, it can be seen that neither the rearrangement of furniture, nor the movement of tags, will affect the correctness of the query results. This is because the "blue denim shirt" would still be on the "brown coffee table," regardless of where the table has been moved to.

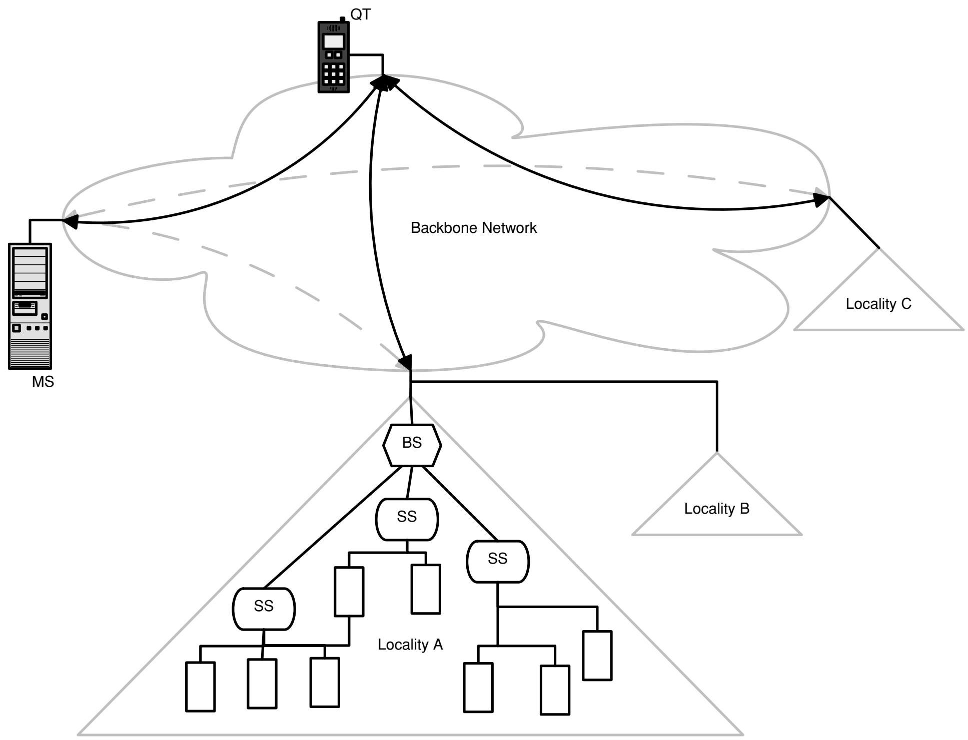


Fig. 2. Architecture of MAX over the backbone network. A user in the system is colocated with the query terminal. The query terminal communicates with the MAX server, and subsequently with the localities. The architecture at each locality is outlined. One should note the background communication between the MS and localities.

2.2 Wide Area Architecture

So far, we have described searching within a single locality, such as a room or office. However, MAX is also able to support wide area searches, meaning that a user can search remotely for objects that are located in different localities. To support such functionalities, we have designed an architecture to connect the various base stations and query terminals. An overview of the network-wide architecture is shown in Figure 2. This architecture allows a user to launch a query from any location where she has access to a backbone network. In short, we will empower people to search everywhere from anywhere.

We observe that this backbone network could easily exploit existing network connectivity, such as LANs, DSL, and WiFi. This allows for scaling of the system across a wide geographical area, possibly the world. This backbone network will connect the various base stations and query terminals, and will also allow access to the MAX server.

We will now detail the roles and functions of the various entities in the network-wide architecture. The description of some functions related to searching a locality, as discussed previously, will be omitted here for clarity of presentation.

(1) *Query terminal*: The *query terminal* (QT) is the human-machine interface used to launch a query. Here, search results must be presented to the users in the most natural manner possible. Hence, the human-computer interface should be intuitive to the user. The interface should ideally be lightweight and platform independent, allowing devices such as desktop computers, personal digital assistants, and mobile phones to be used as query terminals. The query terminal also performs cryptographic operations necessary for providing privacy. It may also have to determine the location of the user. This is a topic that is commonly studied in location-aware computing. Various techniques have been developed in this field of research to localize the user, the simplest being to use a global positioning system. While these techniques are of interest, they are not critical to our application, since we can always fall back on user input. Thus, we defer further details to Yap [2006].
(2) *Base station*: The *base station* (BS) is the representative of the locality with which it is associated. Thus, for the location provided by the user or query terminal to make sense, the base station must provide comparable information. Having said that, the function of processing queries is the core function of the base station, as previously described.
(3) *MAX server*: The *MAX server* (MS) is the central control entity of the MAX system. In its simplest form, it maintains a list of base stations, and a database of users with their keys. Its basic tasks are to act as a central store for information, and to inform the query terminals of the available base stations, as explained in the next section. In scenarios where the number of base stations is limited and manageable, the MAX server could return all of the base stations in response to a query. This would result in the query terminal propagating the query to all base stations, a process referred to as flooding the query. Alternatively, the MAX server could maintain enough additional information[2] to perform filtering operations to provide an appropriate set of base stations, analogous to the presentation of a list of relevant Web site by Web search.

2.3 System Operation

We will proceed to provide a high level overview of how the system will be deployed and function in this section. The exact details will be discussed later.

Whenever a user wishes to find an object, he or she launches the query via a simple interface on the query terminal. For example in Figure 3, the query is "*Book, Mehul*." This query is then communicated to one or more base-stations, as returned by the MAX server, which then broadcasts the query to the substations and tags in the locality. This broadcast mechanism is done over multiple hops. Specifically, the base-station first transmits the query to the substations, which in turn transmit it to the tags. The tags whose descriptors match one or more of the query words, respond to the base-station through the substations. The

[2]For example, the MAX server can maintain a Bloom filter for each base station that is updated periodically, and can efficiently rule out all base stations that have no possibility of having the item. This function is based on the zero probability of false negatives within the Bloom filter.

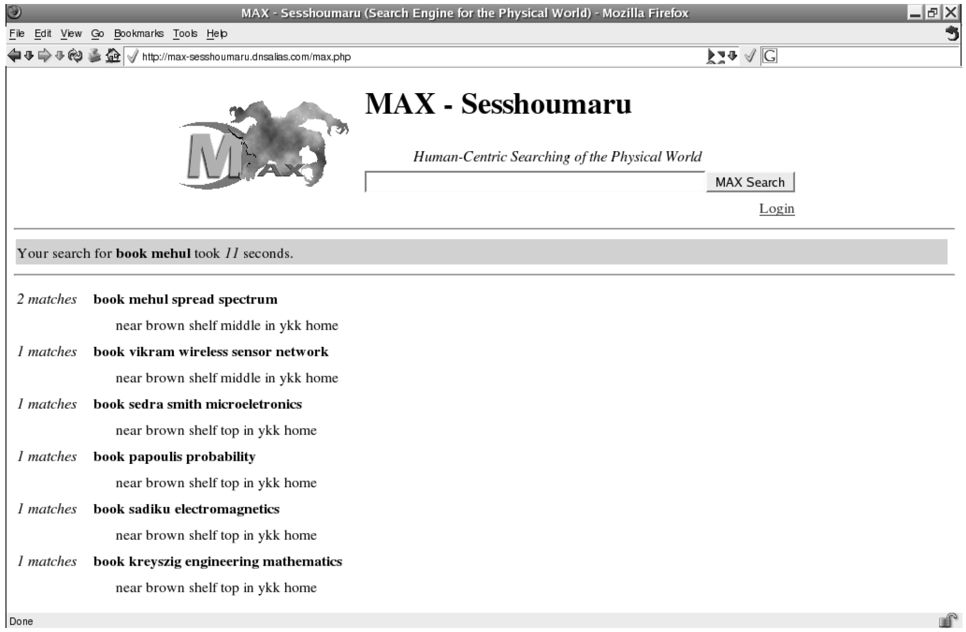


Fig. 3. System interface and search result example—an actual screenshot for our Sesshoumaru prototype described in Section 7.

substations also compute the received signal strength indicator (RSSI) of these responses, and report them to the base-station. The user then sees, at the query terminal, the results ranked by the number of matches (for example, there could be many objects that have the label "book") along with their estimated relative landmark-based location. The relative location is estimated by associating a tag with the substation that hears it with the highest possible RSSI value. A sample response is shown in Figure 3.

Note that in this discussion, we have only given a high level overview and not the precise mechanisms of the query protocol. Given the notorious nature of the wireless channel, one may be concerned about using RSSI values for estimating location. We will discuss this issue in Sections 7 and 9, in which we contend that RSSI is indeed good enough for coarse localization.

3. METRICS AND CHOICES

3.1 Performance Metrics and Terminology

The three basic metrics we consider in this article are:

(i) *Latency*: the time elapsed from the entry of the query statement, to when the results are returned.

(ii) *Energy*: the average amount of energy consumed by the battery-powered devices in responding to a query.

(iii) *Match count*: the number of descriptors on a tag that match some part of the query.

The usefulness of the first two metrics is self-evident. The last metric is useful from the user's perspective, since it determines which search result most likely corresponds to the object for which the user has queried. This is the metric used to rank the search results before presentation to the user.

We discuss below some of the other terminology used in this article.

—*Relevance*. The results returned to the user should ideally be of maximal relevance to the user. By maximal relevance, we mean that the sum of the match counts across all results returned to the user, is the highest sum obtainable. Alternatively, some form of *heuristic* can be employed to return a set of results, without any attempt to determine which result represents the highest sum of match counts. An example of a heuristic method is when each substation independently decides to return only results that are above some minimum match count threshold. This can significantly reduce the communication load of the system. In this article, depending on how the results are returned, we classify results as being either *maximally relevant* or *heuristic*.

—*Overlap*. Overlap refers to the fact that a tag might be within transmission range of several substations. If the mean number of substations covering a tag is large, the system is said to have a large overlap. Given the dynamics of the wireless channel, it is clear that if there are large overlaps, then, due to the variations in the wireless channel, it is likely that a tag might not be associated with the nearest substation.

3.2 Device Choices

We now describe the various choices for the system devices and the associated tradeoffs.

(1) *Tags*. Tags are expected to be extremely cheap and small, if they are to be widely deployed on many objects. They should have either very limited or no constant power sources. As a result, these devices are likely to have extremely limited communication and processing capabilities. Given these constraints, the most obvious choice of technology available today is Radio Frequency Identification (RFID). RFID tags can be either active or passive. Passive tags are very simple devices that can only store information and require no power from batteries. The physical form factor of the device is also relatively small, for example, a 5 cm by 5 cm flat sticker. Whenever an RFID readerir radiates them with energy, they respond by transmitting the information stored within them. The advantage of such a device is that it requires no power supply. Therefore, there is no necessity to replenish any energy source, allowing the device to last for very long durations. However, designing a scalable system using such simple devices might not be trivial. The reason is that, irrespective of what the query is, every tag will always respond with its descriptors. In response, we explore an alternative where passive tags have comparators that will allow them to compare the query with its descriptor, and respond conditionally to a query. The passive RFID tags, specified in ISO15693 [ISO 2000], are able to compare their

identities with a transmission, indicating that such modifications are possible without adding external energy sources such as batteries. We explore the benefits of this customization in terms of energy consumption and delay later in this article. Based on the power of the RFID reader and the frequency used, readers can be classified as proximity readers, with a range of 5 inches, medium-range readers, with a range of 15 inches, and long-range readers with a range of up to 10 feet. With such limited range, base-stations might not be able to hear responses from the tags. Therefore, it becomes the responsibility of the substations to relay information from the tags to the base-stations.

(2) *Substations*. We envision using substations powered by batteries. As a result, they have limited processing and communication capabilities. Given that we are considering passive RFID tags and that these tags have a maximum range of 10 feet, the substations must fulfill two basic functions. First, they must act as RFID readers to query tags in their vicinity. Second, they must relay the results of these queries back to a base-station. In addition they may have to do some limited processing to filter out the results of the queries. In order to conserve energy, they may also have the ability to route query results to the base-station over multiple hops, using other substations. Skye-Tek [SkyeTek 2005] is marketing an RFID reader that integrates a Crossbow Mote, demonstrating that this technology is already available in the market. The choice of the range of the RFID reader is a design issue of concern. If we choose a large range, then we will require fewer substations for a given area. This will also imply that each tag will be in range of many substations. However, since we are using only RSSI values to indicate proximity, we are very likely to choose the nearest substation incorrectly. Therefore, for increased accuracy, we require (1) RFID readers with small range and (2) for each tag to be in the range of very few substations. On the other hand, for scalability and cost, we require as few substations as possible. From our experiments and experience, a reader that has a range of at most one meter is required.

(3) *Base-station*. Base-stations may obtain query responses from the tags via the substations and/or can read the tags directly, using built-in powerful RFID readers. The base-stations will have significant memory and processing ability. Moreover, since they are to be connected to be the backbone network, they are likely to be line powered. Thus, we envision that they can be exploited to significantly reduce the processing and energy burden on the substations and tags.

3.3 System Choices

It is also required of us to decide how much of the computational, processing, and storage burden we would like to push down to the base-stations, substations, and readers. It is intuitive that the more burden we push onto the lower levels of the hierarchy, the lower the delays will be due to parallelism of computation. However, its effect on energy consumption has to be explored. Since the resulting requirements of the various devices would also translate into cost, they should be carefully considered.

Similarly, we have to investigate the density of substations and the degree of overlap. These are closely linked through the range of the reader. Requiring higher density of the substations would directly increase the cost of the deployment. However, a sparse deployment could result in high latency, large energy consumption, and non-useful localization. Neither extremes is acceptable.

Another system choice that has to be made is the choice of modality. The common modalities are ultra-wideband signals, radio-frequency (RF) signals, and ultrasound. Each of these modalities has its strengths and weaknesses. As will be discussed in Section 9, we have decided to use RF with RSSI.

4. PRIVACY

What we have argued so far is that MAX facilitates human-centric search of the physical world. Such a system enables new applications based on sharing (e.g., books on a college campus) and trading (e.g., buying and selling used books) of physical resources. Consider a simple application of sharing books among your colleagues on a college campus. You might designate a certain set of your books as public, meaning that users are able to determine the existence and location of these books. The rest of your books are designated as private, meaning that no one else besides yourself is able to locate these books. The usefulness and power of such an application is clear. Privacy is a critical component in such new applications.

The primary privacy risk facing MAX is the fact that it uses wireless technology to communicate. There are two connotations to privacy in MAX. The first is with respect to objects that one owns, meaning that we wish private objects to be searchable only by the owner. The second is with respect to the space that a user owns or uses. We mean by this that a user might view the ability of a third party to search within a user's personal space as an invasion of privacy. Secure authentication and access, to guard the privacy of both physical spaces and physical objects, are necessary components of the MAX system.

We now discuss mechanisms to ensure both classes of privacy, namely privacy of space and privacy of objects. The former is provided by security agents and the latter by cryptographic techniques.

4.1 Privacy of Physical Objects

We start by designing for object privacy, in which an object marked private can only be searched for by its owner. On the other hand, anyone can search objects marked as public. This notion of public and private marking has been previously described. Each user is authenticated at the query terminal and a pair of public and private cryptographic keys are retrieved (or created for new users).

Given the hardware constraints of the tags and substations, it is not feasible for any complicated authentication to be performed at these devices themselves.

We thus propose that the query terminal/base-station do all the necessary encryption and decryption.

Private objects have their descriptors encrypted using the owner's public key, while public objects have their descriptors stored as clear text. When an owner queries for her objects, the query statement is encrypted by the base station using her public key. The encrypted query and its plain text are then concatenated and sent to the substations. The encrypted component matches the private objects of interest while the plain text component will match all the public objects of interest. All results are then returned to the query terminal, where the encrypted descriptors are decrypted with the owner's private key.

While it is possible to provide a higher order of access control to the objects, it will incur a significant amount of complexity. This in turn translates into having a more powerful tag, which in the opinion of the authors will not be readily available in the near future. Therefore, we have decided to use the designations of private and public as primitives for MAX. Using these primitives, group access can be provided by having a user identity that is shared by multiple users. A user can then simply query using more than one user identity, which can be accomodated much more easily.

4.2 Privacy of Physical Space

In MAX, we will provide privacy of space that emulates that of a physical room. Hence, a physical space can be designated public, private or off-limits in MAX. Physical objects in a public space can be queried by any user. At the other extreme, only an authorized user can search for objects in a space that is designated off-limits. The authentication is done by the security agents residing at the base-station, as illustrated in Figure 1. These agents will communicate with the query terminals via secure tunnels, using standard public key cryptography and key distribution algorithms, such as Pretty Good Privacy (PGP). Before they are allowed to query the locality served by the base-station, users will be authenticated. Public-key cryptography handshaking can be used to prevent any unauthorized access of the space. The space is thus off-limits to anyone but the owner of the space.

These two extreme notions of security for physical spaces (off-limits and public) are arguably insufficient. To see this, consider what happens when a user (say, Bob) inadvertently leaves behind his object in another person's (say, Alice's) off-limits space. In effect this object is not searchable by either Bob or Alice, meaning it is lost to Bob. In another scenario, assume that Alice wishes to put up her laptop for sale. In order for interested buyers to locate Alice's laptop, Alice will have to mark her space public. Note, however, that these searches will do landmark-based localization, and return the location of Alice's laptop with respect to other objects in her house (e.g., Hi-Fi stereo). Alternatively, Alice may leave her laptop in a public space, which is physically insecure. In this case, the laptop is more likely to be stolen than sold at the end of the day. In response, we introduce the notion of a private space. With this notion, when a person searches either for his private object or a public object in another person's private space, the searcher can only ascertain the presence or the absence of the object in the space, and not its location. However, if the owner of a private space searches for his private or public objects, the landmark based localization will be provided in search results.

It should be noted that a binary answer is sufficient for an item to be found or put up for sale. This level of information is precisely what the notion of a private space provides for. We believe that the notion of private space opens up many new and exciting possibilities for MAX.

If a space is marked private, then there are three classes of objects to consider: (1) public objects, (2) private objects owned by the owner of the private space, and (3) private objects owned by a third party (e.g., a user might leave her cellular phone in another user's home or office). We refer the readers to Yap et al. [2005] for a series of examples of how search results are returned to different kinds of users in a private space. We also note that similar provisions appear to be independently proposed in Kapadia et al. [2007] for a different context, showing the need for such security of space.

5. OPTIMAL QUERY PROTOCOLS—METHODOLOGY

We now set out to investigate the correct design choices. The design choices available to us are in terms of devices (e.g., customized passive tags with comparators) and in terms of the querying protocols used. Energy and latency are of critical concern. Since we assume that the substations are battery-powered, energy should be conserved to ensure longevity of the devices. In addition, results must be returned to the user as quickly as possible to enhance user experience. To facilitate comparison of device choices, and to derive the appropriate protocols, we devise a methodology that can be used to design delay optimal and energy optimal protocols.

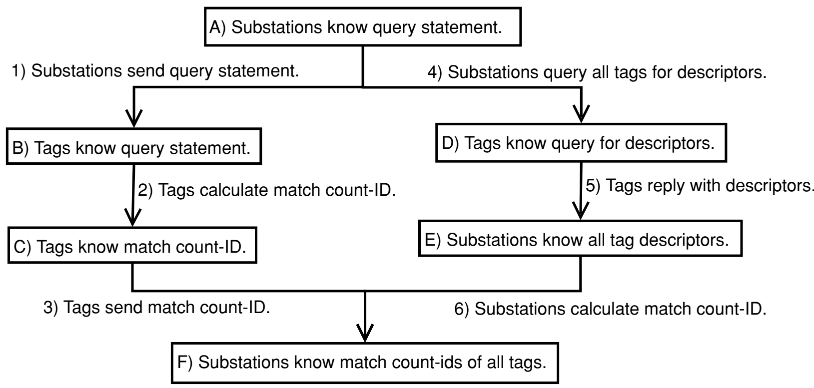
5.1 Problem Definition

We start by defining the problem of finding the optimal protocol. A valid protocol is a set of actions that brings the system from a starting state to a desired end state. Thus our aim is to find a valid protocol that minimizes the energy consumed and/or latency incurred. Such a protocol is termed the *optimal protocol*. The input to the optimization is the set of actions available to the system.

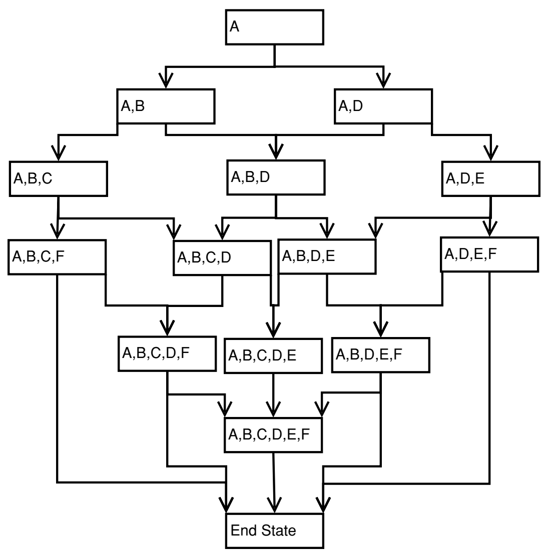
5.2 Methodology

We will follow an example, illustrated in Figure 4, to exemplify the methodology devised. The example is a small but representative subset of the final optimization performed.

A heuristic manner of arriving at a protocol is often to think of the possible approaches. In Figure 4(a), two of the many possible approaches are shown. It is possible to make a choice between these options by comparing their total costs. However, hardly any optimality can be deduced for the resulting protocol. Consequently, it will not be possible to make a fair comparison of device and system choices.



(a) Illustration of possible protocols



(b) States-actions graph for optimization of protocol

Fig. 4. Optimization of protocols.

To arrive at the optimal protocol, we will have to explore all possible combinations of actions. Each action brings the system from one state to another. In addition, each action is associated with a cost, representing the effort required to execute it. Effort, in this context, can be either energy consumption or latency, and possibly a combination of the two. Thus, the cost of a protocol is the sum of costs associated with all its actions. Each action will also require a certain prerequisite state before it can be executed.

To ensure a tractable problem, we make sure that the states are Markovian. This allows the possible actions at each state to be independent of the previous states. An immediate consequence of this formulation is an efficient manner of searching through all possible protocols. We define the following terms:

Definition 1. *A primitive state* represents the knowledge a device possesses in the system. For example, primitive state A represents that the substations know the query statement entered by the user, as shown in Figure 4(a).

Definition 2. An *action* transforms the system from one primitive state to the other, and incurs a cost in the process. Given a primitive state-action pair, one can determine the subsequent primitive state. As illustrated in Figure 4(a), action 1 is described as "Substations send query statement." The action requires primitive state A as prerequisite, and brings the system to primitive state B.

Definition 3. A *state* is defined as a vector of primitive states. For example, the state [A, B] consists of primitive states A and B, as shown in Figure 4(b). Thus, the system can be moved from state [A] to state [A, B] using action 1.

If we appropriately define the states and actions for our system, we can then proceed to design an optimal protocol by finding the minimum cost path from the specified starting state to a desired end state. In our case, several end states can be accepted. Thus, we create an artificial end state that can be reached from the many possible end states, through a costless action. In the example of Figure 4(b), the end condition is for the system to return any state with primitive state F. Therefore, a costless action is defined for every state containing primitive state F to the arbitrary state *End State*.

Thus we represent the states as vertices of a graph, and actions as edges. Each edge has a weight corresponding to the energy or delay required to execute that action. We can now employ any efficient shortest path algorithm, such as Djikstra's algorithm, to find the optimal protocol.

5.3 System Model

Using this methodology, we formulated 45 primitive states and 64 actions, which are not listed here due to space constraints. The primitive states basically represent the knowledge of each device, which can be roughly categorized into query statement, match counts, descriptors, and any decision made. This allows for careful permutation of the possible primitive states. On the other hand, the possible list of processing actions is bounded by the information available at the device, that is, the primitive states. More specifically, one can only send the information to another device, or process it to gain new information. Thus they can be exhaustively listed. To ensure consistency, it is essential to check that each primitive state must be reachable from another primitive state via an action, unless it is the starting primitive state. Similarly, it has to have an action that allows it to move to another primitive state, unless it is the ending primitive state.

If one naively lists all the states possible, by permutation, a total of 3.518×10^{13} states results. By constructing the set of states from the starting state, and recursively applying all possible actions, we can drastically reduce the number of states to 89,350. This reduction of state space is essential for the problem to remain tractable. Such consideration is demonstrated in Figure 4, where a total of 32 states can be generated while only 14 are possible. This reduction of states can be automated, allowing for a large number of states to be processed.

The hardware values used to determine the energy and latency costs of the various actions are listed in the appendix. These values are derived from currently available hardware, namely the Crossbow's MICA2 sensors, Skye-Tek's RFID reader, and ISO15693 RFID tags.

We note that an inventory of the RFID tags by the substations is a one-time process,[3] which can be ignored for each query. We also assume that the base-station is computationally powerful and line-powered. Thus any computation within the base-station suffers negligible delay, as compared to those done by substations and tags. We note that the minimization of energy is to achieve longevity, thus we neglect any energy cost of actions performed by the base-station. On the other hand, we caution that communication between the base-station and substations would incur cost at the substations, which cannot be ignored.

6. OPTIMAL QUERY PROTOCOLS—RESULTS

Armed with this technique, and the values listed in the appendix, we compute optimal protocols for various scenarios. Here, we will concentrate on our ideal application, which is ubiquitous deployment of MAX.[4] We start by exploring the use of a customized smart tag that can calculate and transmit its match count to the substation. It can also transmit its descriptor according to a match count threshold. Then, we look at the appropriate amount of processing to push down to different levels of the hierarchy. In a centralized system, the substations are asked to report all responses or their statistics to the base-station. In a distributed system, each substation can independently do some processing and decision-making to decide what subset of the results or their statistics to return to the base-station.

We note that the results can be maximally relevant or heuristic.[5] A user would like to have $N_{T(S)}$ (as defined in the appendix) results for a single query. A maximally relevant result is one that consists of all of the $N_{T(S)}$ results with the highest match counts. This would require all match count and identity pairs to be returned to the base-station, for selection to be performed. Alternatively, the base-station may decide on the match count threshold based on other statistics, such as match count vector or a subset of the match count and identity pairs. In such cases, the results will be termed heuristic.

Subsequently, we investigate the effect of overlap, which is related to the number of tags a substation can see (or equivalently, the number of substations that a tag can communicate with). To test the effect of overlap, we computed the optimal protocols after changing the values in the appendix to $\sum_i N_T^i = 7500$, $\max_i N_T^i = 225$, and scaling $\sum_i N_{T(S)}^i$ and $\max_i N_{T(S)}^i$ by 5 for decisions by

[3]In the ISO RFID standard, the reader performs an inventory of tags in its vicinity. The tags are subsequently polled when they are read.

[4]The following protocols presented are not optimized for our prototype, which is meant as a proof of concept and not a tool for performance assessment.

[5]Recall definitions from Section 3.1.

Table I. Energy and Latency-Optimal Protocol for a Centralized System with Dumb Tags that Requires only Heuristic Result

1	Substations perform inventory.
2	Query terminal queries base station.
3	Base Station sends query statement.
4	Substations send query for all descriptors.
5	Tags all reply.
6	Substations calculate match count from descriptors.
7	Substations calculate match count vector.
8	Substations send match count vector.
9	Base Station sums all match count vectors.
10	Base Station decides approximated global match count.
11	Base Station sends globally decided approximate match count.
12	Substations filter for selected tag ids based on global approximate estimate.
13	Substations filter descriptors for globally selected ones (approx-match).
14	Substations send globally selected tag descriptors (approx-match).
15	Base Station filters sum of globally approximated tag descriptors.

match count made by substations, and by 4 for decisions by match count made by the base-station.[6] These figures have changed to reflect the degree of overlap. Though the query optimization is independent of the devices used, it can be seen that different devices may have different overlap due to their communication ranges. Hence, degree of overlap will be one of the factors considered in device choices.

Finally, the number of tags is changed from 1500 to 500 and 5000, representing a low and high number of tags respectively. The results are tabulated in Tables II and III. A total of nine protocols are derived, each optimal in energy and/or latency for different scenarios. They are symbolically labeled by letters A through I. An example of the resultant protocol is listed in Table I, while the complete list can be found in Yap [2006].

6.1 Smart Tags

Smart tags are capable of calculating and transmitting their match count. This feature is always used in all optimal protocols where smart tags are allowed, namely, protocols C, D, F, and G. Thus the optimal protocol for smart tags operates in multiple rounds. In the first round, tags that have matches to the query, respond with their match counts. In the subsequent round, only tags that have match counts greater than some threshold, are polled with their descriptors.

For dumb tags, the only sensible response is for all tags to return their descriptors, which are stored in the substations for future processing. The tags are not involved in any subsequent action. This is indeed observed in protocols A, B, E, H, and I.

[6] We note that the exact number of selected tags will be transmitted if the querying is performed using identities decided by the base-station having a complete set of match count and identity pairs. However if querying is performed using a threshold match count decided at the base-station, the descriptors of tags in the vicinity of more than one substation will be retrieved multiple times due to overlap. Thus the number of returned results must be scaled to reflect this condition. The condition is made worse by ill-informed local decisions performed at substations.

We note that these observations indicate that using smart tags has lowered the memory requirement of the substations. More specifically, the smart tags require the storage of selected tags' descriptors, while the dumb tags demand the substations to store the descriptors of all tags around them. Given the experience of memory shortage in the prototype, this motivates the consideration of smart tags.

Another consequence of using smart tags is a decrease in both energy consumption and latency, as observed in Figure 5(a). The energy consumed decreases by an average of 17.66 times, while the delay incurred decreases by an average of 1.31 times for a low overlap system. An enormous benefit is therefore expected by deployment of smart tags.

Similarly for a high overlap system, the energy consumed and latency incurred by using dumb tags is increased by 37.25 times, and 2.71 times respectively, as shown also in Figure 5(a). Both values are significantly larger than those for low overlap systems. This signifies an increase in the benefits of using smart tags, since the overlap the system experiences increases.

Comment: We have shown that significant savings in memory utilization, latency, and energy can be achieved with smart tags, especially in a high overlap system. It is therefore highly desirable for the tags to be able to calculate and transmit their own match count. Thus, despite the potential cost of losing the economy of scale, the customization of tags should be seriously considered.

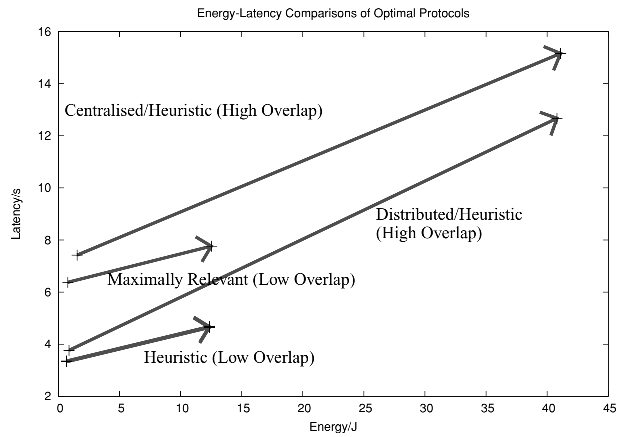
6.2 Distribution of Computational Burden

To see how the computational burden should be distributed, we compared the protocols for a centralized and a distributed system. It is somewhat surprising to observe that there was little or no difference in energy consumption or latency between centralized and distributed decisions for a low overlap system. The energy consumed and latency incurred change by a small amount, as indicated in Table II.

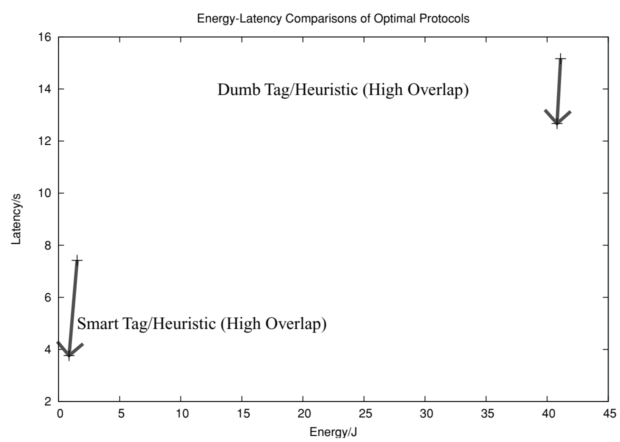
As a matter of fact, the protocols do not change if subjected to the restraining requirement of achieving maximal relevance. For a low overlap system providing heuristic results, the substations would send the match count vector for a centralized system (protocols A and C), and either send a locally decided match count (protocols E and G), or locally filter the descriptors (protocol F) for a distributed system. This change does not render a large saving in energy or latency.

This result should be put into perspective. First, the substations are modelled as Crossbow MICA2 sensors, which consume significant idle power. Thus, the energy for computation may well be significant compared to that for communication, discouraging local computation. Second, the degree of overlap of the system has significant impact on the results.

As expected for high overlap systems, significant benefits of distributing the processing can be seen, as shown in Figure 5(b). The latency is decreased by 0.836 and 0.507 times for dumb and smart tags respectively. For smart tags, a significant decrease in energy consumed of 0.566 times can also be observed.

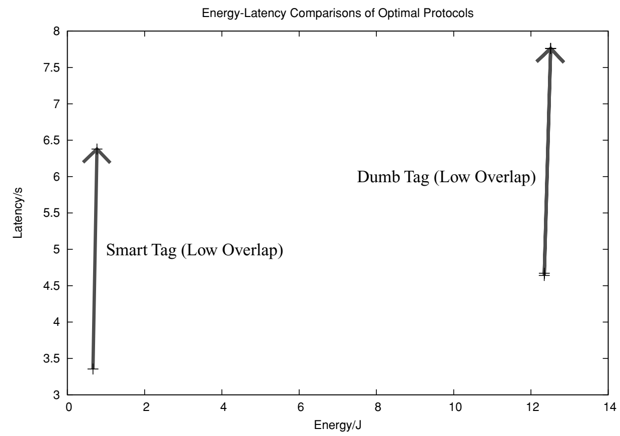


(a) Smart tags to dumb tags. Both energyconsumption and latency increases when the tags are disallowe from calculating and transmitting their match count.

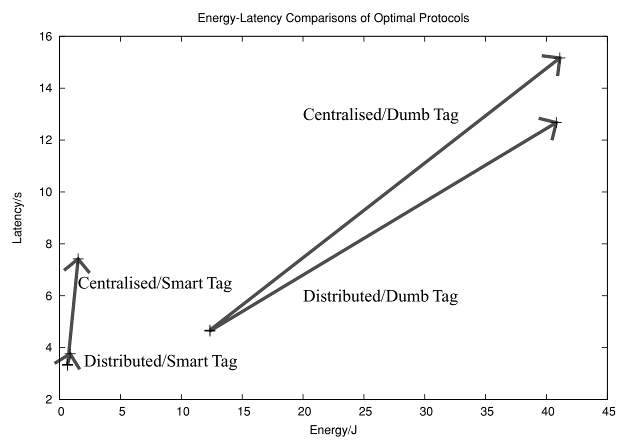


(b) Centralized decision to distributed decision. When the substations in a high overlap system are disallowed from deciding which results are to be returned, the energy consumed decreases slightl while the delay is decreased significantly.

Fig. 5. Energy and latency comparison of optimal protocols. These plots are scatter plots of the optimal energy-latency values achievable under various scenarios (values provided in Tables II and III), where the arrows are drawn to show the trends under various changes.



(c) Heuristic result to maximally relevant result. To ensure maximally relevant results, the energy consumed increase slightly but thelatency increasesd rastically.



(d) Increasing degree of overlap, Both energy consumption and latency increase with increased overlap, especially in systems using dumb tags.

Fig. 5. (Continued).

The savings are results of the substations' local filtering of the descriptors (demonstrated by protocols F and H), which greatly reduces the amount of communication between the substations and the base station in distributed systems with high overlap.

We can also see in Table III that local filtering of descriptors is used for systems with a high density of tags, namely, protocols F and H. Note that the savings are moderate for dumb tags, and more significant for smart tags.

Table II. Optimal Protocols

System/Tags	Maximally Relevant	Heuristic	Heuristic (High Overlap)
Centralized/Dumb	B	A	E
	12.511 J	12.346 J	41.097 J
	7.762 s	4.672 s	15.162 s
Centralized/Smart	D	C	D^* C^+
	0.767 J	0.663 J	1.519 J 1.613 J
	6.378 s	3.355 s	16.278 s 7.420 s
Distributed/Dumb	B	E	H
	12.511 J	12.344 J	40.810 J
	7.762 s	4.641 s	12.674 s
Distributed/Smart	D	F^* G^+	F
	0.767 J	0.637 J 0.661 J	0.860 J
	6.378 s	3.446 s 3.323 s	3.764 s

If the energy optimal and latency optimal protocols differ, they are indicated by an asterisk (*) or plus ($^+$) respectively.

Table III. Optimal Protocols (Tag Density)

System/Tags	Low Density	Moderate Density	High Density
Centralized/Dumb	B	A	E
	3.678 J	12.346 J	17.782 J
	3.413 s	4.672 s	5.729 s
Centralized/Smart	D	C	C
	0.538 J	0.663 J	0.757 J
	3.369 s	3.355 s	3.658 s
Distributed/Dumb	I	E	H
	3.672 J	12.344 J	17.768 J
	3.283 s	4.641 s	5.654 s
Distributed/Smart	D^* I^+	F^* G^+	F
	0.538 J 3.672 J	0.637 J 0.661 J	0.676 J
	3.369 s 3.283 s	3.446 s 3.323 s	3.437 s

If the energy optimal and latency optimal protocols differ, they are indicated by an asterisk (*) or plus ($^+$) respectively.

Comment: By pushing the computation load down the hierarchy, there is a slight improvement in a low overlap system. However, significant benefits can be reaped in a high overlap system, by allowing a local filtering of the descriptors. Therefore, the distribution of the computation load is more important as the degree of overlap increases in the system. However, the constraint of providing maximally relevant results may restrict the system to a centralized one, thus preventing the distribution of computation load. This should be kept in mind during design.

6.3 Maximally Relevant Results

We would like to investigate the cost of ensuring maximally relevant results in a centralized system, as shown in Figure 5(c). An increase in latency, and little change in energy consumed, is observed when maximally relevant results are required over heuristic ones. They have increased by 1.789 and 1.097 times respectively.

For maximally relevant results, the match count and identity pairs of the tags are all delivered to the base-station, accounting for the increased latency. Thus it results in the observation, as stated in Section 6.2, that demanding maximally relevant results would constrain the system to be centralized. However, since match count and identity pairs are small in size, and querying by identities eliminates any unnecessary querying of descriptors, the energy increment is minimal.

Comment: We can see that a significant saving in latency can be achieved by foregoing maximally relevant results, since this allows the system to reduce communication load by sending approximate information. However, the effect of providing heuristic results has to be investigated in an actual implementation. The aim is to provide sufficient information for users' satisfaction, without exceeding the delay constraints.

6.4 Overlap

In Figure 5(d), we investigate the effect of increasing overlap. It can be seen that the increase in energy consumed ranges from 1.350 to 3.329 times. Similarly, the increase in latency ranges from 1.133 to 3.245 times. A large increase is observed in both energy consumed and latency incurred. The large increment in cost is due to the large number of repetitions, especially with dumb tags.

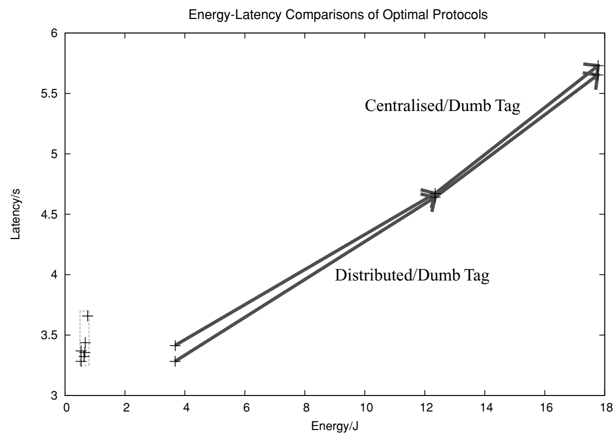
As the overlap increases, the value of distributing the computation burden also increases. By distribution of the computation, the increase in latency is reduced, as evidenced in Figure 5(d). In fact, the optimal protocols for a distributed system with high overlap prefer local filtering of descriptors (i.e., protocols H and F) to sending locally decided match count (i.e., protocols E and G). This is discussed in Section 6.2.

Comment: We can see an increase in latency and energy as the overlap of the system increases. There is also an increasing preference for distribution of computation load. Thus the degree of overlap in the system must be carefully controlled, while providing coverage within the locality.

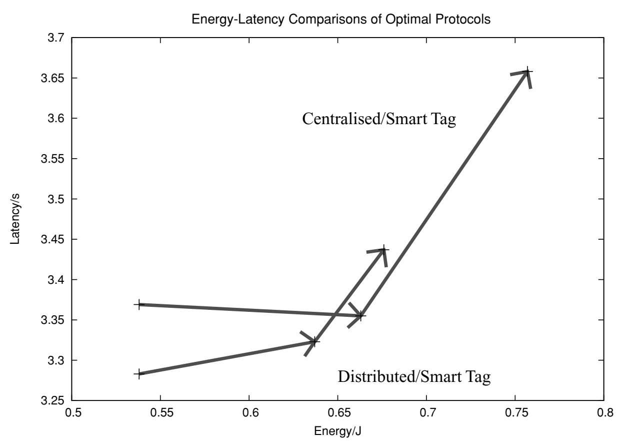
6.5 Number of Tags

We look at the effects of increasing the number of tags, with the results being shown in Figure 6(a). As the number of tags increases from 500 to 1500, the energy consumed and latency incurred increases by 2.284 and 1.198 times, respectively. An increment of 3.084 and 1.383 times is observed, respectively, in the energy consumed and latency incurred, as the number of tags is increased from 500 to 2500.

This increment is in response to the increase in number of tags. A corresponding response in protocol can be observed. For example, the protocol for a centralized system with dumb tags has shifted from protocol B (sending



(a) Increasing Number of Tags. As the numberof tags increases, the energy consumption and latency of systems using dumb tags increases. See Figure 6 (b) (expanded version of the bottom left of this plot marked by agreen box) for resultson systems using smart tags.



(b) Increasing Number of Tags (Smart Tags) as expanded from Figure 6 (a). As the number of tags increases, the energy consumption and latency of systems using smart tags increases less significantly, as compared with those using dumb tags.

Fig. 6. Energy and latency comparison of optimal protocols (density of tags). These plots are scatter plots of the optimal energy-latency values achievable under various scenarios (values provided in Tables II and III), where the arrows are drawn to show the trends under various changes.

Table IV. Time Taken to Locate Object

	Minimum	Maximum	Mean (Std. Dev.)
Manual	10 *s*	300 *s*	101.4 *s* (87.3 *s*)
MAX-aided	2 *s*	110 *s*	39.8 *s* (35.5 *s*)

all match count IDs to the base station) to *A* (sending a match count vector) and later to protocol *E* (sending a locally decided match count), as the density of nodes increases. For a distributed system, the option of locally filtering is available (i.e., protocols *F* and *H*) and thus is utilized at a high density of tags.

We can also see that the increase is marginal for smart tags, allowing for more tags per substation to be deployed. This makes smart tags especially attractive for systems with a high density of tags.

Comment: The latency and energy increase as the number of tags increases. This would place a limit on the number of tags that can be served by a base-station, given a certain delay constraint. The problem may be mitigated by using smart tags, though that may lead to a loss of economy of scale. Alternatively, this points to the need for multiple base-stations to serve a large locality, which is likely to be a more viable solution.

7. PROTOTYPE FOR WIDE AREA

We have implemented a simple prototype of MAX using Crossbow's MICA2 [Crossbow 2005] motes[7] for user trials, as reported in Yap et al. [2005]. We conducted the subjective user trial in a typical 5 m by 5 m office on our campus, containing some wooden furniture, some metal filing cabinets, and scattered books, files, and so forth. We tagged four objects in the office and placed seven substations. Ten random subjects were invited to test the system as users and were asked to locate two objects with MAX, and two objects manually in the room.

The numerical results are presented in Table IV. With the aid of the system, the timing improved to a mean of 39.8*s* with a standard deviation of 35.5*s*. The minimum and maximum times taken were improved to 2 s and 110 s respectively. We wish to remark that the large maximum time is due to the fact that the objects are books hidden in the clutter, while on the other hand there is a significantly low minimum. There were three instances in which the user gave up on searching for an object without the system, while with the system all users were able to find the objects they were looking for. We note here that the number of user trials was few and therefore meaningful conclusions cannot be drawn from the quantitative results. However, the qualitative results validate our hypothesis that landmark based localization using RSSI can be extremely useful to humans.

[7]The choice of Crossbow's MICA2 is due to the availability of the device at the point of development. The optimization framework in Section 5 can be used to derive the corresponding optimal strategies for other devices, such as the more powerful TelosB.

Table V. Memory Consumption (in bytes) of Tag and Substation

Components	Read-Only Memory	Random Access Memory
Substation	17442	1486
Tag	15130	1123

Beyond this preliminary localized prototype, we have developed a prototype based on the optimal protocols, named Sesshoumaru.[8] Provision for reliable communication is also made, while limitations of the mote platform are taken into account. It is remarked that the prototype is developed as a proof of concept for the system, but its actual performance is beyond the scope of this article.

The code size of the implementation is presented in Table V. The MAX server, base stations, and query terminals are implemented in Java. We have provided Secure Socket Layer (SSL) encryption for communication between these entities, and experimented with Elliptic Curve Cryptography (ECC) to secure private tags. More specifically, we have used the ECC implementation from Sorensen et al. [2006] for our prototype, providing 112 bit encryption. A more detailed description of the implementation is deferred to Yap [2006], while we discuss lessons learned.

Many lessons can be learned from the experience of this simple prototype, though the experience is likely to differ from an actual implementation of MAX in view of its simplicity. At the time of writing, a more market-ready prototype is being developed. The main points of the lessons learned from the prototypes are as follows:

— Despite the reputation of RSSI being an unreliable aid in localization, we found that the values returned by the motes during trials in a computer laboratory and a cluttered room were very stable. In fact, we found that the tags were associated with the nearest two substations with high probability during our user trials. A central theme of MAX is the ability of humans, as powerful sensors, to use approximate information to locate objects. From the fact that our prototype, which only provides landmark based information based on RSSI, has improved the subjects' ability to locate things in our experiments, we can deduce that RSSI is good enough for landmark based localization.
— Unlike wireless sensor networks with a large number of nodes to provide redundancy of information, the communication in MAX does not enjoy such redundancy. Thus all messages have to be reliably delivered. For example, loss of a packet that signals the end of a burst of packets, may leave the device in an unexpected state. Such behavior is highly undesirable. In the prototypes, much channel unreliability was experienced, and thus reliable communication has to be explicitly catered to in the implementation of the communication layer. This can be done in two manners, namely, (1) to cater to reliability in the protocols used, or (2) to provide a reliable

[8]This prototype of MAX is named Sesshoumaru, after a character in the Japanese animation, "Inuyasha." This character plays the role of *Hanzen Youkai* (Perfect Demon) in the animation.

communication layer and design protocols, assuming reliability is provided. The former option may provide system efficiency, such as throughput and delay, but comes at a cost of complicated protocol design. Such design also creates obstruction for code reuse and modular design. Apart from channel unreliability, small packet sizes allowed in TinyOS created several problems. With a maximum payload of 29 bytes, a single descriptor has to be sent over multiple packets.[9] Such fragmentation creates unnecessary contention in the wireless channel. Provisioning for longer packets in future devices would greatly simplify the protocol. The corresponding optimal protocols can continue to be derived using the method described in Section 5.

—From the early stages of design and development of MAX, privacy has been considered. This consideration has influenced many design decisions that are critical to the overall functionalities of MAX. Following the common advice of security experts, MAX has taken privacy as a primary concern from early phases of design. Privacy concerns are often the reason for slow adoption of technology by the general public, such as RFID technologies. From our experience in designing MAX, introducing privacy considerations at a later stage could indeed be an insurmountable task.

8. RELATED WORK

Table 1 of Hightower and Borriello [2001] describes and summarizes localization algorithms that have been developed for indoor and urban environments. The modalities involved are usually radio frequency, ultrasound, and video capture. The two main categories of these localization algorithms are location tracking systems and location support systems. More notable recent developments include the Ubisense [2005][10] location tracking system and the Cricket [N. B. Priyantha et al. 2000] location-support system. These systems use ultrasound, RF, and ultra-wideband techniques to do accurate localization via carefully placed beacons. A comparison of the RF-based systems can be found in Steggles and Cadman [2004].

Ubisense [2005] proactively keeps track of the locations of objects and stores them in a central database. A context aware middleware enables a variety of smart space applications. It requires careful placement and calibration of what are called sensors (about four for every 400 square meters) so that all tags can be accurately localized within a physical space. It can provide landmark based localization quite easily, since all object locations are tracked proactively and stored in a central database. However, since object locations are stored centrally, it does not provide for either privacy of objects or physical spaces. Also, based on our experience, it is difficult for a lay person to set up the system easily.

An interesting alternative is the Magic Touch system [Pederson 2001], which actively tracks the human hands and the objects it is in contact with. The

[9]It is possible to increase the length of the packet in TinyOS. However, the increase is limited by the buffer size of the transceiver. Therefore, for our purpose, the problem of fragmentation will continue to be present.

[10]Ubisense is the commercial product developed from the Active Badge and Bat projects.

system assumes that the last known location of the objects can be retrieved during a search. Again, object locations are stored centrally, thus it does not provide for either privacy of objects or physical spaces. A recent proposal called Ferret [Liu et al. 2006] uses RFID with a video capture system and like our system, provides a rough outline of the object's location, leaving the rest to the user's cognition. However, scaling of the system over a wide area appears to be non-trivial.

As opposed to a location tracking system, Cricket offers a location-support system. It uses a set of carefully placed beacons to estimate positions accurately. Cricket devices estimate their own locations and broadcast them as needed. Locations are not stored centrally, ensuring one aspect of privacy. However, in Cricket there is no provision to distinguish between private and public objects, or private, off-limits, and public spaces. Since it requires careful placement of beacons, it is not easy to configure the system. Finally, in order to give human-centric location information, it requires an overlay of map servers, implying that it is not robust to reconfigurations of the physical space.

Vision based systems, such as Köhler et al. [2007], are also unable to fulfill our design goals, because of their need for a field of view. An important function of MAX is to be able to locate objects out of view of the users. The form factors of such devices are also unsuited for ubiquitous tagging of objects.

Other RFID-based systems have been designed for urban deployment. The Bewator CoTag [Bewator 2005] technology is mainly designed for security monitoring and control, while Wavetrend [2005] RFID systems are targeted at pharmaceutical, heath care, manufacturing, and warehouse management. Such systems require experts for deployment. Moreover, they do not provide privacy and are also not robust to reconfigurations in the physical space.

Localization has also been intensively studied in the context of wireless sensor networks. A plethora of algorithms have been proposed for this purpose. In He et al. [2003], the schemes are divided into range-based and range-free. The common modalities used, in both categories, are radio frequency and ultrasound. These works aim to provide exact localization of the nodes or to place them in positions relative to each another. Often, beacons whose locations are known via a Global Positioning System (GPS) or other means, are required [Mondinelli and Vajna 2002; Bergamo and Mazzini 2002; Tarnacha and Porta 2003]. The inappropriateness of such algorithms for the application we have in mind is obvious.

There are also projects that target the creation of a smart environment, such as the NIST Smart Space [NIST 2005]. The smart environment in this project predicts and reacts to the needs of individual users. However, these projects hold no notion of localizing objects. They concentrate mainly on the communication of information.

It may be appropriate to relate our work to sensor database systems, such as Cougar [Cornell Database Group 2005] and TinyDB [Madden 2005], since we query the physical environment. In fact, our work is similar in design to a query process that utilizes in-network processing. However, unlike these systems, we are not dealing with sensor data. The output of our tags is generally nonredundant and thus cannot be aggregated. Moreover, these systems are the least concerned with localization.

9. REFLECTIONS

We have proposed a system that allows human-centric search of the physical world. Specifically, the system was designed with the goal of (1) human-centric operation, (2) privacy, and (3) efficiency and scalability. The proposed hierarchical (possibly hybrid) architecture exploits the fact that humans are powerful sensors who only require visible cues and landmarks to locate and search for objects. Two notions of privacy, namely spatial privacy and object privacy, are provided for in our system. It is interesting to note that the notion of privacy and security in this system is different and arguably more complex than that in traditional systems in the digital world. Finally, we have investigated the design of optimal query protocols and showed that significant improvements in energy and latency could be achieved with a slight customization of passive RFIDs. We went on to develop a more comprehensive prototype, exploring the various technical difficulties involved. From these, we have identified the following future avenues of research and development for MAX.

(1) The encryption of tags for object privacy poses a unique challenge to cryptography. The public cryptography employed must provide small cipher length. While we have demonstrated that Elliptic Curve Cryptography is a feasible option, its suitability can be challenged by other cryptography techniques.
(2) The implementation of MAX has been performed on commercial off-the-shelf hardware and open source software. While this is strong evidence as to its practicality, it is unlikely to be the final form of deployment. The capability of the system cannot be fully assessed without customized substations and RFID tags. The development of such products would be another step needed before a widespread adoption of MAX is possible.
(3) In our small scale network, each query terminal can afford to query all the available base stations. For a global scale deployment of MAX, the problem of query scoping has to be addressed. The user and the importance of location can be exploited. The details of such a query scoping algorithm that is scalable and efficient remains an open question.
(4) The main barrier to commercialization of MAX, at the time of writing, is the unavailability of low cost RFID readers with range of one meter. While such a device is not available today, we believe it is likely to be available in the near future, as was the case with TelosB, which has become available since the time of our initial development.

Most importantly, the design of MAX allows for the paradigm of everything has *a* place rather than everything has *its* place. Together with the new notions of privacy provided by MAX, this will empower a variety of new applications that involve the sharing and trading of physical resources.

From our results and experiences, we have gained valuable insights into some overarching issues regarding the design of the system. A review of these is provided in the following.

—*Use of RSSI*. We have chosen to use RSSI to estimate the approximate location of an object. In localization literature, it is usually argued that RSSI is not a good metric due to the time varying nature of the wireless channel, and the effects of shadowing. However, in this article we argue that for the purpose of our system, RSSI is sufficient. Recall that our thesis is that humans are powerful sensors and thus are skilled at locating objects when given an approximate location. Since we have considered largely static environments such as offices and homes, we can assume that channel fluctuations are small. An object might not be associated with the nearest substation due to shadowing effects. However, in our system, we argue that a transmission range of about one meter should be used for the substation. This implies that the localization error is always bounded within a circle of radius one meter. Since we are attempting landmark based localization, we believe this will be sufficient to help a human locate an object quickly. There is also the issue of locating objects that are encased in metal (e.g., a metal cabinet). For such situations, the substations will be placed/embedded in the cabinet, and it can be designed in a manner such that the antenna is on the outside of the metal cabinet (this could be done at the time of manufacture, taking aesthetic considerations into account). This will allow objects enclosed within metal cabinets to be located.

—*Choice of Tags*. The choice of devices for the tags is clearly of paramount importance. If devices such as smart dust or active RFID are employed, they would be capable of more processing, since they are battery powered. The fact that we see significant improvements in energy and latency by adding a small processing functionality to passive tags, seems to indicate that additional intelligence at the tags is desirable and worthwhile. However, there is a cost versus performance-benefit tradeoff to using active RFID or smart dust-like devices, which we still need to investigate.

—*Smart Tags*. Customized passive tags (defined as smart tags) that can respond conditionally to queries, have been investigated. These operations can be achieved through simple logic gates, summers, and comparators. This ensures that the power delivered by the RFID reader is sufficient for execution of these operations. Currently, ISO15693 standard RFID tags already have the ability to compare their identities and conditionally respond in the inventory phase [ISO 2000]. The main issue in having smart tags is customization, which may lead to a loss in economy of scale, resulting in a tradeoff between the performance and cost of the system.

—*Energy/Latency Trade-off*. The trade-off between energy-efficiency and latency is another interesting aspect. In most wireless systems, improving the performance of one metric implies relaxing the requirement on the other. However, due to the nature of passive RFID, this is not true for most cases. In our design of optimal protocols, the delay optimal and energy optimal protocols are identical in most scenarios. We acknowledge the existence of cases in which the energy and latency optimal protocols differ. The reader should note that these scenarios are restricted when the use of smart tags is employed. In these cases, the energy optimal protocols would use the identity of the tags to query for its descriptors, while the latency optimal protocols use an approximated match count to do so.

We began with the objective of designing a human-centric search system for the physical world. Observing that humans are powerful sensors with comprehensive cognitive and computational capabilities, we posited that this could be exploited to design a simple, efficient, and scalable search mechanism. Furthermore, we performed optimization of the querying protocols in various scenarios and gave insights to the appropriate actions in these conditions. The successful implementation of prototypes also demonstrates the practicality of the system. Moreover, the subjective user feedback was positive and extremely encouraging, validating our simple yet powerful hypothesis.

APPENDIX

PARAMETERS OF OPTIMIZATION

For the hardware chosen, we consider the following to arrive at Table VI.

- Information can be found in SkyeTek [2004], Crossbow [2004a, 2004b].
- The average length of an English word, considering all words, is between four and five. However, the average length of word letters in use may not be so. Thus, a convenient value of eight is chosen for analysis before as more accurate value can be determined from further research.
- The reader has a read range of 7 to 15 cm depending on the antenna used. The values are available in SkyeTek [2004]. The SkyeRead M1-Mini is assumed to be switched off when not in use, thus drawing 60 μA.
- The processing of the RFID tags is accomplished by hardware implementations and thus supports true parallel processing. Processing power would be the same as the communication power requirement at 82.5 mW, since the tags are passive. However, the speed is highly restricted by the power available. Thus an equivalent speed of 18.7 kHz is estimated from the required processing for comparing a unique identifier, as specified in ISO15693. Therefore, the convenient value of 20 kHz is used. A data rate of 26 kbps is also based on ISO15693.

As for the environment, we make the following assumptions and considerations to derive the values listed in Table VII:

- We consider a 5 m by 5 m room densely deployed with substations. Thus the total amount of furniture and immobile objects is estimated to be around 50. An average of 30 tags is expected for each substation, giving a total of 1500 tags in reality. However, due to overlapping coverage of the substations, the number is multiplied by 1.5. This degree of overlap is low, since we are considering Skye-Tek's RFID reader with a maximum range of 15 cm.

Table VI. Hardware Device Parameters

Description	Symbol	Value
General Values		
Length of Match Value	L_M	1 byte
Length of RSSI Values	L_R	1 byte
Length of Query	L_Q	4 words
Maximum Length of Descriptor	L_D	8 words
Length of Each Word	L_W	16 bytes
Base-station to Substations Communication (MICA2DOT Mote)		
Power Required to Transmit	P_S^T	8.7 mW
Data Rate	R_S	38.4 kbps
Substations (MICA2 Mote)		
Processing Speed	S_S	4 MHz
Power Required	P_S^A	46.5 mW
Substation-Tags Communication (SkyeRead M1-Mini)		
Power Required	P_T	82.5 mW
Data Rate	R_T	26 kbps
Tags (RFID)		
Processing Speed	S_T	20 kHz

Table VII. Simulation Parameters

Description	Symbol	Value
Number of Substations		
Exact Number of Substations	N_S	50
Number of Tags		
Sum of Tags under each Substation	$\sum_i N_T^i$	2250
Maximum Number of Tags under a Substation	$\max_i N_T^i$	45
Number of Selected Tags		
Total Number of Selected Tags	$N_{T(S)} = \sum_i N_{T(S)}^i$	40
Maximum Number of Selected Tags under a Substation	$\max_i N_{T(S)}^i$	10
Scaling Factor for Different Decisions		
Decision by Substations		1.5
Decision by Base-Station, where a threshold match count is used for querying the descriptors.		1.3
Decision by Base-Station based on approximated statistics, where specific identities are used for querying the descriptors.		1.15
Decision by Base-Station based on accurate statistics and specific identities are used for querying the descriptors.		1

—We assume that a user would require a total of only 40 results. This is based on the intuition that searches are seldom useful beyond a small portion of the returned results. We also assume an asymmetric distribution of desired objects in the room, since it is common to cluster similar objects together.

—We note that the values of $\sum_i N^i_{T(S)}$ and $\max_i N^i_{T(S)}$ would be different for different types of decisions made. For example, in Figure 1, the tag d will be returned twice by substations 1 and 2 if a local decision is made. This is because this overlap information is not available to all the substations. Thus we scale these values for different decisions made.

ACKNOWLEDGMENTS

The authors would like to acknowledge Chern-Sia Lim, Phillip and Huang Limei for their work in developing the various prototypes leading to Sesshoumaru. We would also like to thank our associate editor Samuel Madden and the anonymous reviewers for their time, effort, and helpful comments.

REFERENCES

BERGAMO, P. AND MAZZINI, G. 2002. Localization in sensor networks with fading and mobility. In *Proceedings of the 13th IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'02)*. 750–754.

BEWATOR. 2005. http://www.bewator.com/uk/index.xml.

CORNELL DATABASE GROUP. 2005. COUGAR: the network is the database. http://www.cs.cornell.edu/database/cougar/.

CROSSBOW. 2004a. Power technology and management, www.xbow.com. Pred in.

CROSSBOW. 2004b. Radio, Rf concepts, and TOS radio stack, www.xbow.com. Pred in.

CROSSBOW. 2005. Crossbow technology—inertial & gyro systems, wireless sensor networks, smart dust, advanced sensors. http://www.xbow.com/.

HE, T., HUANG, C., BLUM, B. M., STANKOVIC, J. A., AND ABDELZAHER, T. 2003. Range-free localization schemes for large scale sensor networks. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (Mobicom'03)*.

HIGHTOWER, J. AND BORRIELLO, G. 2001. Location systems for ubiquitous computing. *Comput. 34*, 8, 57–66.

ISO 2000. *ISO/IEC 15693 Identification Cards—Contactless Integrated Circuit(s) Cards—Vicinity Cards*, 1st Ed. ISO.

KAPADIA, A., HENDERSON, T., FIELDING, J. J., AND KOTZ, D. 2007. Virtual walls: Protecting digital privacy in pervasive environments. In *Proceedings of the 5th International Conference on Pervasive Computing (Pervasive'07)*.

KÖHLER, M., PATEL, S. N., SUMMET, J. W., STUNTEBECK, E. P., AND ABOWD, G. D. 2007. Tracksense: infrastructure free precise indoor positioning using projected patterns. In *Proceedings of the 5th International Conference on Pervasive Computing (Pervasive'07)*.

LIU, X., CORNER, M. D., AND SHENOY, P. 2006. Ferret: RFID localization for pervasive multimedia. In *Proceedings of the 8th International Conference on Ubiquitous Computing (UbiComp'06)*.

MADDEN, S. 2005. TinyDB—a declarative database for sensor networks. http://telegraph.cs.berkeley.edu/tinydb/.

MONDINELLI, F. AND VAJNA, Z. K. 2002. Self localizing sensor network architecture. In *Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference*. Vol. 1. 823–828.

PRIYANTHA, N. B., CHAKRABORTY, A., AND BALAKRISHNAN, H. 2000. The cricket location-support system. In *Proceedings of the 6th International Conference on Mobile Computing and Networking (Mobicom'00)*. 32–43.

NIST. 2005. NIST smart space laboratory. http://www.nist.gov/smartspace/.

PEDERSON, T. 2001. Magic touch: a simple object location tracking system enabling the development of physical-virtual artefacts in office environments. *Pers. Ubiquitous Comput. 5,* 1, 54–57.

SKYETEK. 2004. Product datasheet, skyeread M1 mini. www.skyetek.com.

SKYETEK. 2005. Skyetek RFID. http://www.skyetek.com/readers_Mini.html.

SORENSEN, T. B., KRAGH, T., AND ERLANDSEN, M. K. 2006. Jecc—elliptic curve cryptography in java. http://jecc.sourceforge.net/.

STEGGLES, P. AND CADMAN, J. 2004. A comparison of RF tag location products for real world applications. Tech. Rep., Ubisense.

TARNACHA, A. AND PORTA, T. L. 2003. E-STROBE: an adaptive beacon activation algorithm for sensor localization. In *Proceedings of the 58th IEEE Vehicular Technology Conference (VTC'03)*.

UBISENSE. 2005. Ubisense-the smart space company. `http://www.ubisense.net/`.

WAVETREND. 2005. Wavetrend. `http://www.wavetrend.net/`.

YAP, K.-K. 2006. MAX—human-centric search of the physical world. M.S. thesis, National University of Singapore. `http://www.stanford.edu/~yapkke/Research/MAX/2006Thesis/MAXThesis2006.pdf`.

YAP, K.-K., SRINIVASAN, V., AND MOTANI, M. 2005. MAX: human-centric search of the physical world. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys'05)*. ACM Press, New York, NY, 166–179.

Received November 2006; revised July 2007; accepted January 2008.