Searching Techniques in Internet of Things (IoT)

Seminar Report Submitted By

Shah Mitul Abhaykumar

11535027

For the partial fulfillment of the requirement for the award of the degree of

MASTER OF TECHNOLOGY

in

Computer Science & Engineering

Under the Guidance of

Dr. Anjali Sardana

Department of Electronics & Computer Engineering

Indian Institute of Technology Roorkee

Roorkee 247 667, Uttarakhand, India

July 2012

Abstract

As technology pulls world towards new heights, we are having smart devices everywhere. Internet of Things (IoT) is the future of Internet which embraces each and every real world *objects, places* and *people.* One can see that our future will be full of sensors and these type of *smarter* objects. Which indicates that we will be having a massive amount of search space as sensors and their readings generates a lot more data than the humans do. We will need device discovery mechanisms in such environment as we might want to find something like keys or wallet. Generally sensor data are highly dynamic and possess very short lifespan, so current searching techniques like crawling and indexing would not be applicable much. And we need some new and different approaches to handle this issues which makes *'Searching Techniques in Internet of Things'* an active research area. We discuss three different device discovery mechanisms Snoogle, MAX and Dyser in this report.

Contents

Abstract ii

Table of Contents iii

List of Figures v

List of Tables v

1 INTRODUCTION 1

1.1 Internet of Things . . . 1
1.2 Applications . . . 2
1.3 Device Discovery in IoT . . . 3

2 SEARCHING TECHNIQUES 4

2.1 Major Issues . . . 4
2.1.1 Architecture Design . . . 4
2.1.2 Search Locality . . . 4
2.1.3 Scalability . . . 4
2.1.4 Real-time . . . 4
2.1.5 Privacy and Security . . . 5
2.2 Various Searching Techniques Overview . . . 5
2.2.1 Snoogle/Microsearch . . . 5
2.2.2 MAX . . . 6
2.2.3 Objects Calling Home (OCH) . . . 6
2.2.4 Distributed Image Search (DIS) . . . 6
2.2.5 Real-time Web Search Engines . . . 7
2.2.6 Dyser . . . 7

3 REAL-TIME SEARCHING TECHNIQUES 8

3.1 Snoogle - A Search Engine for Pervasive Environments . . . 8
3.1.1 Features . . . 8
3.1.2 System Architecture . . . 8
3.2 MAX : A Human-Centric Search of Physical World . . . 10
3.2.1 System Architecture . . . 11
3.2.2 System Operation . . . 12
3.2.3 Security And Privacy . . . 13

3.3 Dyser 13
3.3.1 System Architecture 13

4 COMPARISON 15

5 CONCLUSIONS 17

References 18

List of Figures

1.1 Technology Roadmap : Internet of Things [3] 1
1.2 Internet of Things : Infographic [4] 2
3.1 Snoogle Architecture 9
3.2 MAX Architecture 11
3.3 MAX System Interface and Search results example 12
3.4 Dyser Architecture 14

List of Tables

4.1 Comparison 16

1 INTRODUCTION

1.1 Internet of Things

Internet today is daily medium to connect thousands of people across the globe. And Internet of Things is going to be the future of it. The term Internet of Things was first used by Kevin Ashton in 1999 [1] . IoT mainly aims at developing our daily life more sophisticated and flexible. IoT will be a networks of smarter objects which includes sensor, actuators and humans. This each object will have their own unique identity and each will have communication and computation capabilities [2]. Each object in this network consists of different types of sensor which collects the data from its environment and it either passes that data to other nodes or does some local processing on it. The possible outline of IoT future is shown in Figure 1.1.

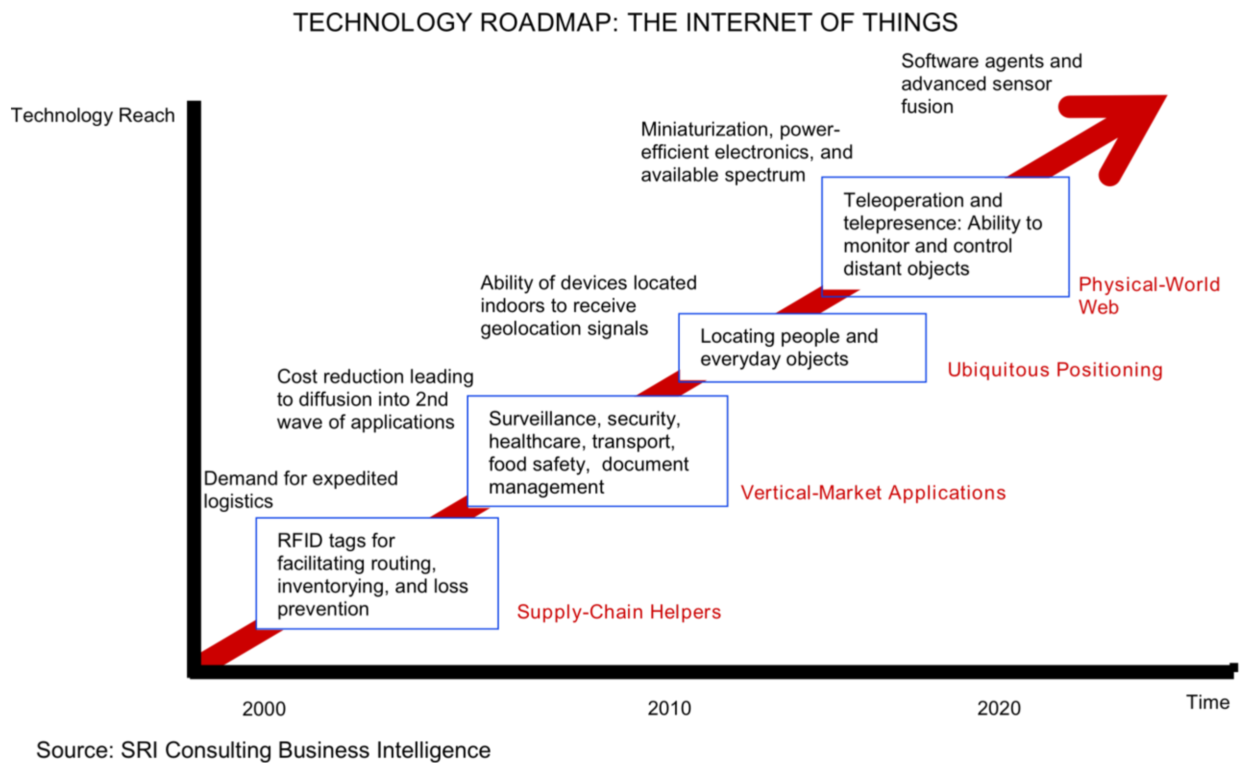

TECHNOLOGY ROADMAP: THE INTERNET OF THINGS
Technology Reach
Demand for expedited logistics
RFID tags for facilitating routing, inventorying, and loss prevention
Supply-Chain Helpers
Cost reduction leading to diffusion into 2nd wave of applications
Surveillance, security, healthcare, transport, food safety, document management
Vertical-Market Applications
Ability of devices located indoors to receive geolocation signals
Locating people and everyday objects
Ubiquitous Positioning
Miniaturization, power-efficient electronics, and available spectrum
Teleoperation and telepresence: Ability to monitor and control distant objects
Physical-World Web
Software agents and advanced sensor fusion
2000
2010
2020
Time
Source: SRI Consulting Business Intelligence

Figure 1.1: Technology Roadmap : Internet of Things [3]

These objects can be anything a home appliance, car, shoes, furniture, streetlights or even your cloth. They will then be connected to a global network which we now call as Internet.

1.2 Applications

Some of the interesting application of IoT are like to switch off the lights when you sleep, prepare a coffee when you wake up, fire or gas leakage detection and take action accordingly in case of emergency, intelligent home, smart transport services, warehouse management, precision agriculture, etc. It can also be used in military for reconnaissance and surveillance. So we can say that our future life will be full of this *smart objects* which can *sense, communicate* and *take decision* by themselves. The Internet of Things will allow everyday objects to be identified, tracked, traced and monitored. A part of nice infographic by Cisco on IoT is shown in Figure 1.2.

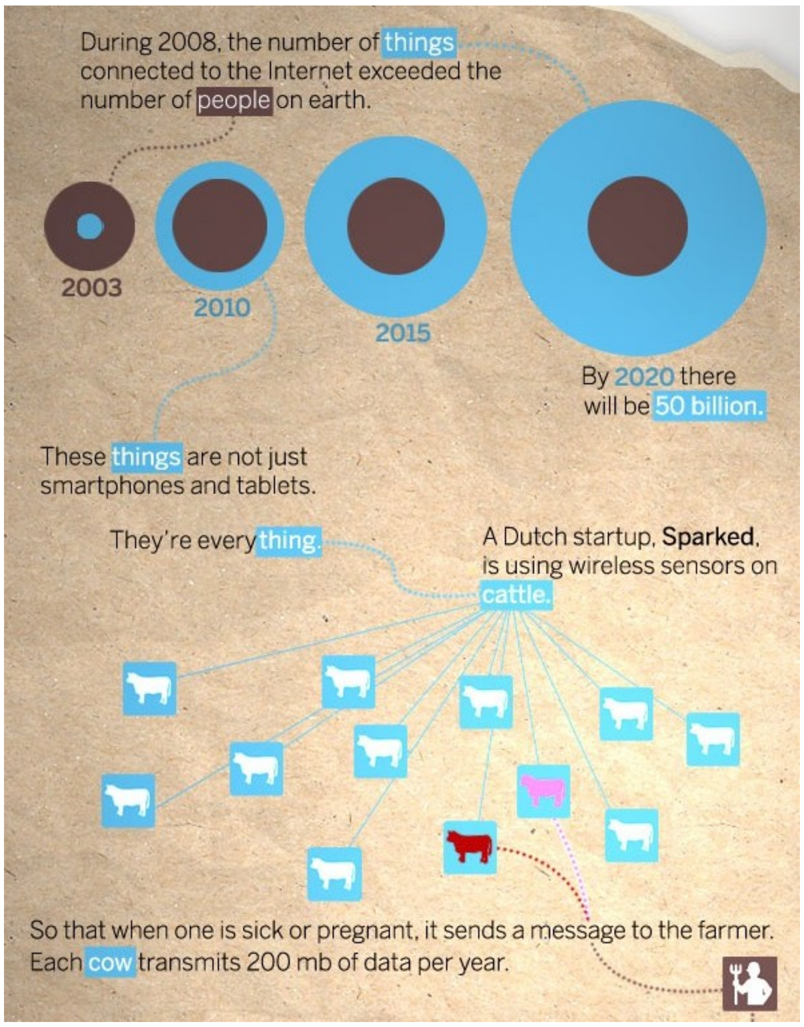

During 2008, the number of things connected to the Internet exceeded the number of people on earth.
2003
2010
2015
By 2020 there will be 50 billion.
These things are not just smartphones and tablets.
They're everything.
A Dutch startup, Sparked, is using wireless sensors on cattle.
So that when one is sick or pregnant, it sends a message to the farmer. Each cow transmits 200 mb of data per year.

Figure 1.2: Internet of Things : Infographic [4]

Some of the other already implemented applications are as follow.

- Bicing [5], a public bicycle-sharing system in Barcelona, Spain, where users can see the number of bicycles available at each rental station in real-time on the Web.
- WEMO [6] is another product which allows us to operate electronic appliances using motion or a simple smart phone application.
- Nike+iPod [7] is an product which allows to put a sensor in your shoe which will measure how much miles you ran and you can connect it to your phone then it will post on behalf of you on social network sites.
- Plants in the garden that texts their owner whenever they need water [8].

IoT gets awareness via use of sensors and to get contextual information it uses RFID tags. According to [9] by 2015, we will have hundreds of billions of RFID tagged objects, generating billions of sensor readings every moment. Radio-frequency identification (RFID) is often seen as a prerequisite for the Internet of Things. The major technologies that would dominate IoT applications are WSN, RFID and Mobile Communications. RFID will be used for addressing objects uniquely and WSN will be used for communication among this sensor embedded objects.

1.3 Device Discovery in IoT

The huge amount of sensor and their automated readings indicates that we will have massive amount of search space. IoT will work in much larger information space than current Internet. We believe as document centric web, searching will become heart of IoT as well [10]. There are many reasons to believe so as your all objects will be connected to network you might want to locate something like your keys or you might want to check if there is enough milk for the week while shopping far away from home or to find a quite place for one day picnic. But there are some challenges like IoT is built on WSN and sensor node have very limited power supply, so searching techniques must be energy efficient. Some other challenges are like scalability, real-time results, dynamic networks and security.

2 SEARCHING TECHNIQUES

This section provides overview of seaching (device discovery) techniques in IoT and also provides major issues in them. One possible solution is to equip each node with a GPS sensor, but wireless sensor nodes have limited power supply and they must be cheap in order to be used widely so this is not a feasible solution. In the case of IoT the mobile nodes will be highly dynamic such that they can be part of network at any time and leave a network without prior notification. Searching in IoT will be on local objects rather than faraway information centers.

2.1 Major Issues

The major issues [11] for searching in IoT are :

2.1.1 Architecture Design

In real world people do want to operate objects locally or remotely that distinguishes IoT search engines from commercial web search engines. And in case of wireless nodes the communication must be minimal to conserve power. Design requires new techniques ranging from crawling, indexing, storing and querying.

2.1.2 Search Locality

In case of Internet users want to search remote database while in case of IoT user wants to search a thing nearby to him like finding a key in office.

2.1.3 Scalability

In IoT the environment is saturated with huge number of sensors. Sensors and RFID keep capturing objects and generating an extremely huge number of readings. And this readings have very short life span. For instance, a reading of a RFID passive tag may be expired within 2 seconds. And traditional search engine techniques does not work in such environments. Even if we use them, the overall performance would be downgraded.

2.1.4 Real-time

This is one of the most important feature of IoT. And as sensor reading posses a short lifespan, we have to rely on real-time communication paradigm. Also real world objects are more movable and they change their location frequently. Most web-page locations on the other hand are static even though their content is dynamic. So maintaining up-to-date data is difficult and for that two communication paradigm exists *timer* or *beacon* scheme.

- *Timer Based* : In this approach the mobile node will send its own presence to central node. Once central node does not receives *alive* message from a node within some period, it assumes that node has been moved to some other region. This approach is not energy efficient as mobile agent will have to keep broadcasting its own state all the time. But it guarantees up to date data.
- *Beacon Based* : In this approach the central node will keep broadcasting beacon frame periodically. And mobile nodes will reply to that to mark their presence. If a node moves out of one's range then it wont respond to the beacon frame and central node will do the necessary changes. This techniques is energy efficient but does not guarantees up to date data as a node might move between time of two periodic updates.

2.1.5 Privacy and Security

As Ross Anderson pointed out that many security solutions will fail when applied to different environments [12]. Unlike the enterprise environment, where physical boundaries and firewalls can separate insiders and outsiders, users and service providers can coexists in a pervasive computing environments. Moreover the data content is also more important in case of IoT or Pervasive environments, it might actually have users current location or his health status which are extremely private.

2.2 Various Searching Techniques Overview

Now we will look at some of the techniques designed for searching in this environment.

2.2.1 Snoogle/Microsearch

The basic idea behind this system [13], [14] is that sensor nodes attached to real-world's objects carries a textual description of that object or other user defined information in form of keywords. Users than have opportunity to find real-world objects matching an ad-hoc query consisting of a list of keywords and the system then would return top-k entities matching the result. It uses information retrieval techniques to index information and process user queries, and Bloom filters to reduce communication overhead. Security and privacy concerns were also given a thought while designing this system. In the next section we will look at this technique in more detail.

2.2.2 MAX

This [15] system follows similar model to Snoogle, it attaches sensor to real-world objects which carries object's textual description. One of the major goals of MAX was to develop a system that is human centric. It provides location information in a form natural to human, i.e., with reference to identifiable landmarks (e.g. on the dining table) rather than precise coordinates. This system has been developed with following objectives (i) human-centric operation, (ii) security and privacy, and (iii) efficient search of any tagged object. To make search efficient MAX uses hierarchical architecture consisting of tags, sub-stations, base-stations. For security considerations you can mark a tag as a public or private tag and private tags are searchable only by owner.

2.2.3 Objects Calling Home (OCH)

This [16] system allows users to query the current location of lost real-world objects. For this purpose, real-world objects are tagged with device holding the identity of the object and mobile sensors are used to detect presence and identity of such objects. The authors have also developed a prototype in which they have used mobile phones (bluetooth enabled) as object sensors and objects are tagged with small Bluetooth modules. Here mobile phones works as object sensor to detect any objects as well as user interface to put a query and display result. User can also specify budget in terms of time or number of messages transferred as part of query. In OCH the query is not sent to all the object sensors for scalability considerations. OCH employees various heuristics to approximate location of a sought objects like (i) Object is probably at a location that is close to the location where it was last seen. (ii) The object is at a location that has been recently visited by its owner. (iii) Object is at location where user spends most of his/her time like office.

2.2.4 Distributed Image Search (DIS)

This system [17] uses camera as sensors. Users can pose a queries by specifying an image and the system returns details of sensors that have captured similar images. Then results are ranked and only top-k results are returned to user. To avoid transmitting whole image this system uses a feature extraction mechanism that maps an image to set of visterms (similar to keywords) to describe th image. After capturing an image a sensor node converts it to visterm. And this resulted visterms are stored in inverted index for efficient lookup.

2.2.5 Real-time Web Search Engines

Real time search engines provides searching through dynamic content. They do not support device discovery directly but the underlying mechanism could be used to search thorough dynamic content. Twitter search (https://twitter.com/#!/search-home) is one of the example which constantly keeps searching through dynamic content in the world of tweets. Technorati (www.technorati.com) is also another service which provides searching through web blogs in real time. One thing that is common to this both is that they have limited their search spaces. Twitter permits only messages of size 140 characters and Technorati focus on blogs only.

2.2.6 Dyser

This system [18] [19] allows real-world objects to be searched by their current state. The authors argue that users of IoT of search engines would be a Web surfer rather than a domain expert. So instead of searching for a loudness sensor having current reading below 30db, a user will be more interested in for places which are currently quite. In the model they have two elements : sensor and entities both have a virtual counter part, a web resource, a URL. We will discuss more about Dyser in next section.

3 REAL-TIME SEARCHING TECHNIQUES

In this section we are going to look at some of the techniques defined above in a more detailed manner.

3.1 Snoogle - A Search Engine for Pervasive Environments

Snoogle [13] is an information retrieval system built on low-cost wireless sensor networks. In this system each real world object is tagged with a sensor which contains short description of the object. A user can query Snoogle to find a particular mobile object, or a list of objects that fit the description. Snoogle uses information retrieval techniques to index information and process user queries, and Bloom filters to reduce communication overhead. Security and privacy protections are also engineered into Snoogle to protect sensitive information.

3.1.1 Features

- Snoogle uses compression techniques like Bloom filters to reduce the data transmitted within the sensor network. A Bloom filter is basically a way to compress information into a form that can still be searchable. It generates smaller amount of information for the objects to transmit to an IP (Index Point), but this might result in false positive.
- The distributed top-k query algorithm is used to further reduce the communication cost for user distributed queries.
- For security and privacy framework for a user to search a sensor network of objects. Each object defines its access attributes similar to a file in UNIX system, allowing for personal, group, and global permissions. A user must show that his access rights matches with the objects access attributes before searching an object. And uses public key cryptography to protect users privacy.

3.1.2 System Architecture

Snoogle consists of mainly three components, the architecture is shown in Figure 3.1. Main three components are Object Sensor, Index Points and Key Index Points.

- *Object Sensors* : The object sensor are the sensors attached to real world objects. As real world objects can be stationary or moving ,this sensors can be static or mobile. Snoogle does not requires this all sensors to be of the same kind, it requires all object sensors to communicate using the same radio frequency. The sensor data is already preloaded or incrementally added by the object owner.

- *Index points (IP)* : An IP is a static sensor device that is associated with a physical location, for example, a specific room in an office building. IP s are responsible for collecting and maintaining the data from the object sensors in their vicinity. They are responsible to build inverted index for local search. A typical IP is battery powered, and equipped with a micro controller, radio module, and a large amount of flash memory.
- *Key Index Point (KeyIP)* : The KeyIP collects data from different IP s in the network. The KeyIP is assumed to have access to a constant power source, powerful processing capacity, and possess considerable storage and processing capacity. Snoogle does not restricts the number of KeyIPs in general.

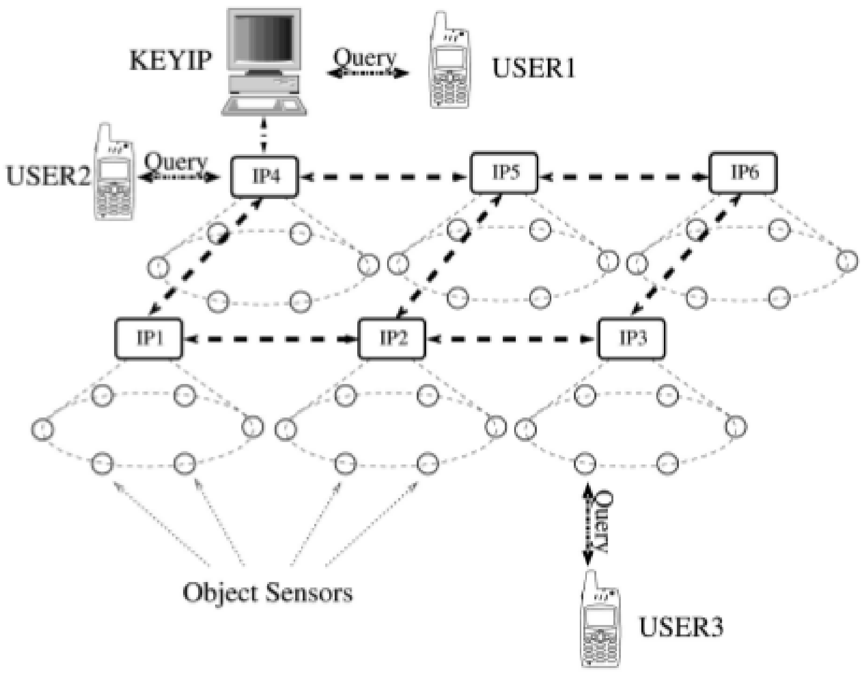


Figure 3.1: Snoogle Architecture

Snoogle adopts a two-tier hierarchical architecture shown in Figure 3.1 The lower tier involves object sensors and IPs. Each IP manages a certain area within its transmission range. Object sensors register themselves and transmit the object description metadata to the specific IP. On the upper tier, IPs have dual roles. First, IPs forward the aggregated object information to the KeyIP so that the KeyIP can return a list of IPs that are most relevant to a certain user query. Second, IPs also routes traffic between IPs, the KeyIP, and objects. The KeyIP, considered as the sink of the network, holds the global object aggregation information reported by each IP.

User can query using some smart portable device such as smartphone or a PDA. Two different kinds of query are supported local and distributed. A local query is to perform search in the region of some particular IP, when user knows the approximate location of physical object to be searched. Ands in the case of distributed query user directly contacts KeyIP.

Each object sensor posses two kind of data related to physical object, payload data and matadata. Payload data is the short description of the actual physical object and metadata is representation of payload data.

The IPs will collect the data from various sensors and organize them into indexed chains. This data is then stored in on board flash memory rather than flash memory due to reliability and limited memory concerns. For a case when an object sensor comes under range of two or more IPs, we use predetermined mapping table to identify the IP that the object sensor should select. They also send periodic updates to KeyIP. IP can mainly perform three kinds of operation Insert, Delete and Modify. These three operations will be performed whenever a new object comes in to range of an IP, moves out of range or there is change is object data, respectively.

The IPs can can use timer based approach or beacon based approach for maintaining real-time data. Author has set the limit to number of sensor under the region of one IP to be 200. This is to prevent an IP to be flooded with too much of traffic. When resolving a query, sensors are ranked according to how many of the sought keywords they contain.

Some of the limitations of this approach are : (i) It supports searches for static meta-data although there is a mechanism for updating data, for that push to IP and KeyIP is required which is not scalable approach. (ii) The centralized approach at KeyIP is not suitable as for each distributed query KeyIP requires to pull data from all the IPs. (iii) As this system is using Bloom filters the results are approximate.

3.2 MAX : A Human-Centric Search of Physical World

MAX [15] is the human-centric search engine for finding real-world objects. It provides location that is natural to humans rather than actual coordinates. This system inherently assumes that all physical objects like cloths, documents, keys are tagged with small devices which have limited communication and processing capabilities. For privacy concerns a tag can be public or a private, where public tags are searchable by everyone and private tags can be searched only by its owner. The authors of this paper observed that humans are powerful sensors who requires only approximate coordinates in terms of clues and landmarks to locate objects.

3.2.1 System Architecture

MAX uses a hierarchical structure where objects in each level are more mobile than the objects in higher levels. Like tags associated with rooms can be at highest level (Base Stations), while furniture in the room can be at next level down (Substations) and all other objects within the room at the lowest level of the hierarchy (Tags). The three tire hierarchy is motivated by the way that humans organize and describe locations of their things. The basic architecture is shown in Figure 3.2 and basic elements are defined below.

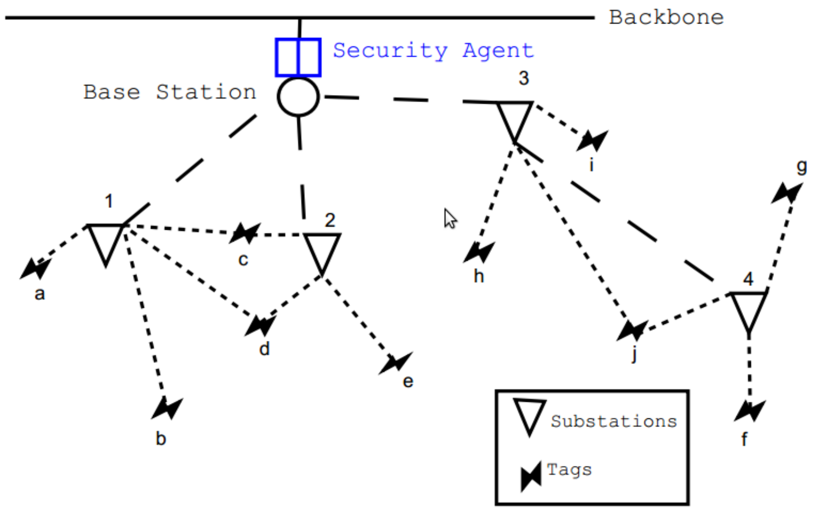


Figure 3.2: MAX Architecture

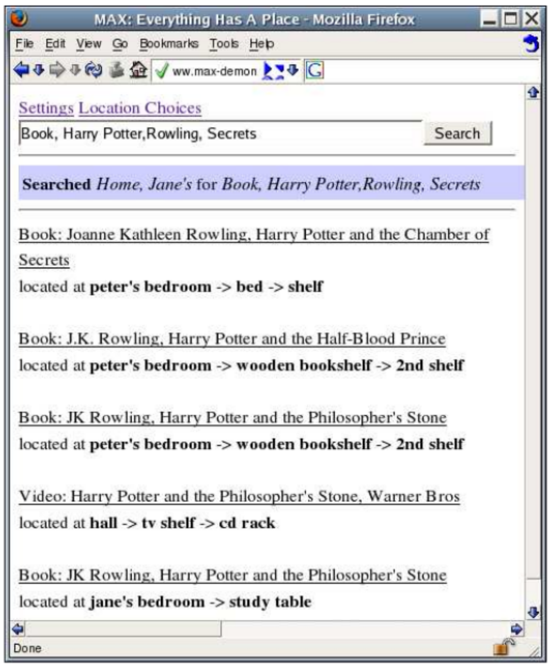
- *Base-stations* : Base-stations are at the highest level of hierarchy, and represents the immovable locality such as a room and it stores descriptor of this locality (e.g. Guest Room). This base-stations also work as mediator between wired backbone network and wireless tags. The base-stations will have significant memory and processing capabilities. They are line powered and does more processing such that it reduces the processing and energy burden on sub-stations and tags.
- *Sub-stations* : Sub-stations are at next level and they describe objects which are static and change position occasionally. Same as base-station, sub-stations also store the label or description of the object they are attached to (e.g. Study table). Sub-stations will be powered by batteries and have limited processing as well as communication capabilities. Sub-stations have dual role, first they have to act as RFID readers to query tags in their area. Secondly, they must forward that results to base-station.

- *Tags* : These tags are at lowest level and attached to the objects that are easily movable (e.g. book,key). Each tag stores the descriptor of the object they are attached to. Multiple descriptor are allowed in each tag. The authors have defined that, it is the responsibility of the owner to tag sensibly if he/she wants the object to be found. This tags has be extremely cheap and small. They will have either limited or no constant power source. They have the most limited communication and processing capabilities in the hierarchy. Given these conditions RFID tags are the most obvious option for tags. In the prototype the authors have used an alternative which allows to compare the query with the descriptors and respond conditionally.

- *Security Agent* : This are software agents which resides at the base-stations. It authenticates the user before they are allowed to query in the locality covered by that base-station. This system uses handshaking based on public key cryptography to prevent any unauthorized access to the locality owner's items.

3.2.2 System Operation

This system uses a simple yet powerful UI for querying and receiving results. As shown in th Figure 3.3 the query could be *"Book, Harry Potter, Rowling"*.


MAX: Everything Has A Place - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
ww.max-demon
Settings Location Choices
Search
Done

Book, Harry Potter,Rowling, Secrets

Searched *Home, Jane's* for *Book, Harry Potter,Rowling, Secrets*

Book: Joanne Kathleen Rowling, Harry Potter and the Chamber of Secrets
located at **peter's bedroom -> bed -> shelf**

Book: J.K. Rowling, Harry Potter and the Half-Blood Prince
located at **peter's bedroom -> wooden bookshelf -> 2nd shelf**

Book: JK Rowling, Harry Potter and the Philosopher's Stone
located at **peter's bedroom -> wooden bookshelf -> 2nd shelf**

Video: Harry Potter and the Philosopher's Stone, Warner Bros
located at **hall -> tv shelf -> cd rack**

Book: JK Rowling, Harry Potter and the Philosopher's Stone
located at **jane's bedroom -> study table**

Figure 3.3: MAX System Interface and Search results example

The query is then sent to the base stations who broadcasts that query to the substations and sub-stations in turn transmits it to tags. The tags whose one or more descriptor matches the query keyword, responds to the sub-station. The sub-stations will listen the tag responses, and computes the Received Signal Strength Indicator (RSSI) of these responses, and finally reports this all values to base-station. The tags which have the maximal matches are then displayed to the user as shown in figure.

3.2.3 Security And Privacy

As already defined for security and privacy concerns, two types of tags are supported public and private. As the hardware limitations of tags and sub-stations all the encryption/decryption task will be performed at query terminal and base-stations only. Private objects have their descriptors encrypted using the owner's public key while public key objects store their descriptor as clear text. When an owner queries for his/her objects, the query statement will be encrypted using his/her public key and sent to substations. Each tag contains a bit to specify weather it is a public or private object. And accordingly clear text or encrypted keywords will be matched. At last when all the results are returned at query terminal the text will be decrypted using owner's private key.

The only limitation of this system is that for each and every query it has to be scent to each sub-station and tag which makes this approach inappropriate for global networks.

3.3 Dyser

Dyser [19] allows real-world objects to be searched by their current-states. As the case with MAX authors of this system also believe that end users will be average web surfers rather than domain experts. In this system each sensor and entity is identified by a URL which is accessible using HTML. The relation between sensor and entity is many-to-many. The UI design is similar to web search which consists of keywords like *"Italian restaurant loudness : quite"*. In this query the first two keywords are static and are related to the entity page, while last element of the query represents the dynamic property which is determined by loudness sensor.

3.3.1 System Architecture

As shown in the Figure 3.4 Dyser consists of a resolver, index and indexer. Index stores the metadata on sensors and entities, while indexer crawls sensor and entity pages. Authors of argue that although there is a central index, the results do not rely on central database. That means there is no global view of the world and for each query the search engine needs to query relevant sensors. The authors have proposed to use sensor rankings to enhance scalability because otherwise too many sensors will be contacted unnecessarily. For this ranking process various prediction models are used which returns the probability of a sensor reading a specific value at a given point of time.

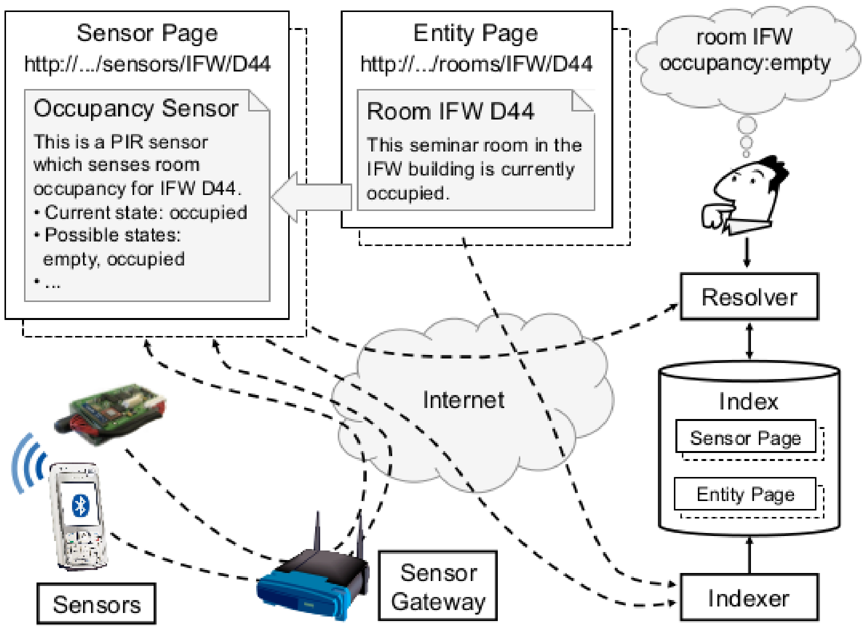

Sensor Page
http://.../sensors/IFW/D44
Occupancy Sensor
This is a PIR sensor which senses room occupancy for IFW D44.
• Current state: occupied
• Possible states: empty, occupied
• ...
Entity Page
http://.../rooms/IFW/D44
Room IFW D44
This seminar room in the IFW building is currently occupied.
room IFW occupancy:empty
Resolver
Internet
Index
Sensor Page
Entity Page
Indexer
Sensors
Sensor Gateway

Figure 3.4: Dyser Architecture

Dyser performs the following steps for each query.

1. First all the entities that do not matches to given keywords are not considered any further. Then this entities are ranked according to relevance and process starts with first x entities.

2. For each entity the probability that entity matches the current search is calculated.

3. The considered entity are the reordered according to probability.

4. Starting from the top each entity will be contacted andd cheacked weather their values actually match with search state.

5. If enough k results have been found than list is ordered according to user preference and displayed to him, otherwise the process starts with another x entities.

In summery this technique actually used two different ranking schemes one for sensors to be considered and one for user's preference to fulfill users expectations.

4 COMPARISON

In this section we compare the three real-time searching techniques according to parameters defined in [18]. The various parameters are defined below. The comparison is shown in Table 4.1.

- *Aggregation Type* : Type of aggregation used by the system to keep up-to-date data to central node. It can be timer based or beacon based or hybrid.
- *Query Type* : There are two types of queries. Ad-hoc queries, which are one-time queries where a result is returned immediately and the query is then completed. And the other is continuous queries which are active for a period of time and matches are returned while the query is active.
- *Query Language* : Query language can be key-word based or a complex language. In keyword based technique user specifies the list of keywords to be matched.
- *Query Scope* : Scope can be local or global. In local query search is performed in local domains only while in case of global query search is scaled upto global Internet of Things.
- *Query Time* : Queries can refer to the real-time state or the historical state of entities.
- *Query Accuracy* : The query accuracy is said to be exact when returned entity list is guaranteed to match query, whereas for heuristic algorithms returned entities may not actually match the query perfectly. Use of heuristics causes in approximate results.
- *Query Content* : This parameter defines the rate of change in sought data. It can be either static meta data, or pseudo-static meta data which changes rarely or dynamic contents which change frequently.
- *Entity Mobility* : This parameter defines weather system supports mobile entities or not.
- *Entity State* : The state of an entity can be described either by a finite, discrete set of categorial states (e.g., “hot”, “cold”), or by a value on a continuous scale (e.g., degrees).
- *Target Users* : A discovery system may be either designed for domain experts or for direct use by end users, similar to Web search engines.

As we can infer from the table, MAX is better compared to other techniques as it provides results which are more intuitive to humans. The query accuracy might be heuristic based but system assumes that human are power full sensors that can use approximate location of objects.

Table 4.1: Comparison

Dimention	Snoogle	MAX	Dyser
Aggregation Type	Hybrid	Beacon	Hybrid
Query type	Ad hoc	Ad hoc	Ad hoc
Query language	Keywords	Keywords	Keywords
Query scope	Local	Local	Global
Query time	Real-time	Real-time	Real-time
Query accuracy	Heuristic	Heuristic	Exact
Query content	Pseudo-static	Pseudo-static	Dynamic
Entity mobility	Mobile	Mobile	Mobile
Entity state	Categorial	Categorial	Categorial
Target users	End users	End users	End users

It supports local search, as in IoT we actually need to find the objects which are physically nearer to us. The interface is quite similar to web search engines which makes it user friendly. And finally it supports mobility via energy efficient beacon (pull) based approach.

We can say that no current system exists which posses all the idle parameters like energy efficient, search for dynamic as well as static data and support of mobile objects. We need a system that have all these features as well as it must have an easy-to-use front end, as most of the users will be web surfers and not the domain experts.

5 CONCLUSIONS

Internet of Things will open gates to many new applications of the Internet. Of course searching is the heart of this whole technology as there is massive amount of data and user requires to quickly locate physical objects. In this report, first we have discussed what Internet of Things is and need of device discovery. Then we look at various searching techniques for device discovery mechanism. Next we have discussed some search techniques which supports real-time data like Snoogle, MAX and Dyser in more detail and finally compare them on different parameters. And finally we can say that among this three MAX outperforms other two.

References

[1] Kevin Ashton, That 'Internet of Things' Thing. *RFID Journal* 22: pp. 97-114, 2009.

[2] Agrawal S. and Das M.L., "Internet of Things - A Paradigm Shift of Future Internet Applications".*Nirma University International Conference on Current Trends In Technology Engineering (NUiCONE)* pp. 1-7, Ahmedabad, 2011.

[3] "Technology Roadmap" http://en.wikipedia.org/wiki/File:Internet_of_Things.png

[4] "IoT : Infographic" http://blogs.cisco.com/news/the-internet-of-things-infographic/

[5] "Bicing" http://www.bicing.cat/

[6] "WEMO" http://www.belkin.com/wemo/

[7] "Nike+iPod" http://www.apple.com/in/ipod/nike/

[8] "Botanicalls" http://www.botanicalls.com/

[9] Das R. and Harrop P., "RFID forecasts, players and opportunities 2011-2021",*IDTechEx.com, Tech. report*, 2010.

[10] Sundmaeker H. and Guillemin P. and Friess P. and Woelffle S., "Vision and challenges for realising the Internet of Things", *CERP-IoT, European Commission*, pp. 72-73, Luxembourg, March 2010.

[11] Zhang D. and Yang L.T. and Huang H., "Searching in Internet of Things: Vision and Challenges", *IEEE 9th International Symposium on Parallel and Distributed Processing with Applications (ISPA), 2011*, pp.201-206, Busan, 2011.

[12] Anderson R., Security engineering: a guide to building dependable distributed systems. *Wiley Publications*, 2010.

[13] Wang H. and Tan C.C. and Li Q., "Snoogle: A Search Engine for Pervasive Environments", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 21, No. 8, pp.1188-1202, August 2010.

[14] Tan C.C. and Sheng B. and Wang H. and Li Q., "Microsearch: A Search Engine for Embedded Devices Used in Pervasive Computing", *ACM Transactions on Embedded Computing Systems (TECS)*, Vol. 9, No. 4, pp. 1-29, New York, March 2010.

[15] Yap K.K. and Srinivasan V. and Motani M., “Max: Human-Centric Search of the Physical World”, in *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, ACM, pp. 166-179, San Diego, 2005.

[16] Frank C. and Bolliger P. and Mattern F. and Kellerer W., “The Sensor Internet at Work: Locating Everyday Items Using Mobile Phones”, *Pervasive and Mobile Computing*, Elsevier ,Vol. 4, No. 3, pp. 421-447, 2008.

[17] Yan, T. and Ganesan, D. and Manmatha, R., “Distributed Image Search in Camera Sensor Networks” in *SenSys '08: Proceedings of the 6th ACM conference on Embedded network sensor systems*, ACM, pp. 155-168, 2008.

[18] Romer K. and Ostermaier B. and Mattern F. and Fahrmair M. and Kellerer W.,“Real-time search for real-world entities: A survey”, *Proceedings of IEEE*, Vol. 98, No. 11, pp. 1887-1902, Nov. 2010.

[19] Ostermaier, B. and Elahi, B.M. and Romer, K. and Fahrmair, M. and Kellerer, W., “Dyser: towards a real-time search engine for the web of things” in *SenSys '08: Proceedings of the 6th ACM conference on Embedded network sensor systems*, ACM, pp. 429-430, 2008.

PAPER ACCEPTED

[1] Mitul Shah, Anjali Sardana, “Searching in Internet of Things using VCS”, *1st International Conference on Securing Internet of Things*, ACM, August 2012.