Experimenting an Indoor Bluetooth-based Positioning Service

G. Anastasi,
Computer Engineering Dept., University of Pisa, Italy
g.anastasi@iet.unipi.it

R. Bandelloni,
Computer Science Dept. University of Trento, Italy
r.bandelloni@iei.pi.cnr.it

M. Conti, F. Delmastro, E. Gregori, and G. Mainetto,
IIT-National Research Council (CNR), Italy
{firstname.lastname}@iit.cnr.it

Abstract

The Bluetooth wireless technology is an emerging technology originally designed as a short-range connectivity solution for personal, portable, and handheld electronic devices. This paper briefly presents the functionality and the architecture of an indoor positioning service based on this technology. Most of the design choices for the service have been strongly influenced by Bluetooth features. The effectiveness of the indoor positioning service is critically analyzed. Experimental and simulation results used for defining the policy of mobile device discovery are shown.

1. Introduction

In current distributed systems, the notion of mobility is emerging in several forms and used in many applications. One of these forms is user mobility that arises naturally in wireless mobile computing. Here, when a mobile user moves, his point of attachment to the fixed wired network changes. Current applications of wireless mobile computing cover diverse application areas, on different spatial scales, ranging from the very small dimension of sensor networks up to the worldwide size of satellite-based networks [1]. In the scale represented by corporate and academic office spaces, buildings, and campuses, the increasing presence of Bluetooth wireless technology[1] [2] [3] is reflected in the wide availability of this technology on a large number of Personal Digital Assistants and Portable PCs. In this context, the mobile users utilize the Bluetooth technology of their portable devices in office applications and to individual access to corporate services. Access to the intranet infrastructure and services of a corporation are granted by placing several Bluetooth access points inside the building.

In this paper, we present an experiment that we performed using Bluetooth and an Ethernet LAN as the enabling technologies of an indoor positioning service, named BIPS, covering a building. BIPS offers a service that allows a mobile user to visualize on his portable device the shortest path he has to follow in order to reach another mobile user inside the same building. The typical building could be an academic department and the mobile users can be students, visitors, professors, or staff members. The core of BIPS is devoted to tracking the mobile users standing or walking inside a building. To this aim, BIPS manages location information about moving users. BIPS considers each room of the building as a granule of location information. It has a station architecture consisting of a set of Bluetooth cells, one for every significant room of the building, interconnected via an Ethernet LAN with a central server machine. The location database of BIPS operates on the central server.

2. An overview of BIPS

BIPS is an indoor positioning service designed for tracking mobile users who may be in motion inside a corporate building. Every mobile BIPS user is represented by a handheld device equipped with a Bluetooth interface for interacting with the static part of the system. The static part of BIPS is a locally distributed system consisting of a centralized server machine and a set of workstations interconnected via an Ethernet LAN. Every workstation is provided with a Bluetooth interface that allows the static part of BIPS to communicate with handheld devices. The main task of every BIPS workstation is discovering and enrolling those mobile users who enter its coverage area. Once a handheld device has been enrolled, its position is communicated to the central server machine where the position is stored in a database for successive lookups. The network architecture of BIPS is illustrated in Figure 1 (according to Bluetooth terminology, a picocell is called a *piconet*). An off-line procedure has been implemented for

This work has been funded by MIUR in the framework of the VICOM Project.

[1] Bluetooth is a trademark owned by the Bluetooth SIG, Inc., USA.

Proceedings of the 23 rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03)
0-7695-1921-0/03 $17.00 © 2003 IEEE
IEEE Computer Society

registering new BIPS users. The procedure associates the name of a user with a user identifier (userid). In this phase, a password and a set of access rights are defined for enforcing security and privacy issues. When a registered user with a mobile device enters a piconet of the building for the first time, the first action he should perform is logging into the BIPS system. The most important effect of this operation is the definition of a one-to-one correspondence between a userid and the Bluetooth device address (BD_ADDR). From this moment until the mobile user logs out of the system, BIPS will constantly track all the movements of the mobile device inside the building.

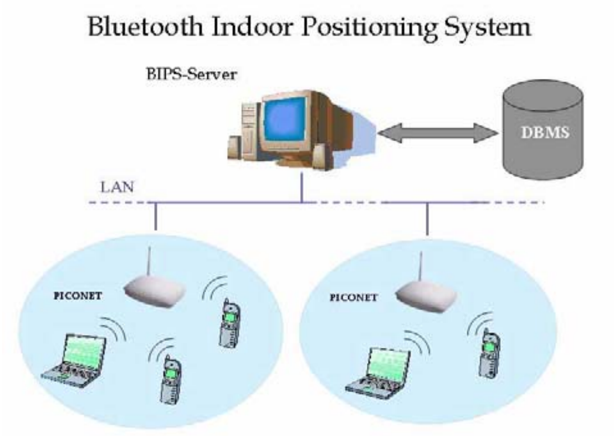

Figure 1. BIPS network architecture.

Before generating the query, BIPS verifies that the target mobile user is logged in and that the querying user has the right to formulate this question. The formulated query is a very simple spatio-temporal query that takes the following form: "Select the target actual piconet of the mobile device BD_ADDR1 where BD_ADDR1 is associated with a userid1 and where userid1 is associated with the given user name". To track the users positions the strategy of maintaining accurate position information is the most important technical issue of BIPS. The system does not impose very stringent requirements on position information maintenance since the mobile users move at a maximum speed of 2 meters per second through circles with a radius of 10 meter. In BIPS, every workstation has the task of computing the presence of those mobile devices inside the piconet. These presences are revealed at fixed intervals of time. In order to reduce the computational and communication load of the system, a workstation updates the central location database only when it reveals a new presence or a new absence in its piconet. Once the current piconet of the target mobile user has been identified, BIPS needs to determine the shortest path. To do this, BIPS defines a weighted undirected connected graph that reflects the topology of workstations inside the building. There is a node in the graph for every BIPS workstation. An edge between two adjacent nodes is defined when there is a physical path in the building that connects the rooms containing the two corresponding workstations. Every edge is labelled with a weight, a positive integer that represents the distance between the two workstations. Thus the problem that BIPS has to solve is the classical shortest-path problem, i.e. to find the path in the graph connecting two given vertices with the property that the sum of the weights of all edges is minimized over all such paths. BIPS implements the Dijkstra algorithm for solving this problem [4]. We underline that the static nature of BIPS wired network allows us to compute off-line all the shortest paths that connect all the possible pairs of two nodes. Hence the computation of the shortest path has no impact on BIPS online activities.

3. Programming the Bluetooth technology for BIPS

The basic concept in Bluetooth communications is the *piconet*, a collection of Bluetooth devices that can communicate with each other by sharing a common channel. A piconet is a star-shaped wireless network in which one device plays the role of the *master* and every other device plays the role of the *slave*. Master and slaves of a piconet are synchronized according to a channel-hopping sequence that is a function of the master's BD_ADDR and clock. Bluetooth is a Frequency Hopping System where the clock cycle lasts for 312.5µs. Thus the clock rate is 3.2KHz. In BIPS, the role of master is always played from a fixed workstation that is also connected to the wired network, and the role of slave is always played from the portable devices. Two sequential phases are needed for establishing a communication channel. The first initial phase is called *inquiry* and it corresponds to device discovery from the master. The second phase is called *paging* and it corresponds to initial connection setup. In both phases the master and the slaves perform different actions, we will briefly summarize them in the following.

3.1. Inquiry phase

A master begins device discovery by entering into the inquiry state and broadcasting messages in the following way. The inquiry hopping sequence is split into two 16-hop parts, called train A and B. A single slot lasts for 625µs. On every even slot, the master sends two ID packets switching between two frequencies of the same train every 312.5µs. On odd numbered slots, the master listens for slave answers. Since a single train will last for 16*312.5µs and the slots for sending and listening are interleaved, two 10ms trains are defined. Each train must be repeated for at least $N_{inquiry}$ = 256 times before a new train is used. In order to collect all responses in an error-free environment, at least three train switches must take place, so the inquiry state may have to last for 10.24s.

A slave that wants to be discovered enters in inquiry scan state. In this state, it listens for ID packets on the same 32 dedicated frequencies used from the master. The slave changes listening frequencies every 1.28s. The $T_{w\_inquiry\_scan}$ time, during which the slave listens on the same frequency, must last enough to completely scan one train. $T_{inquiry\_scan}$ is the time between the beginning of two consecutive inquiry scan cycles and it will last at most 2.56s. Both intervals can be changed. Default values are:

$T_{inquiry\_scan}$ = 1.28s $T_{w\_inquiry\_scan}$= 11.25ms

3.2. Page and connection phases

Known the identity of the devices in its proximity, the master of a piconet explicitly pages them to join its piconet. First the master selects the device to page and then it sends page messages on every frequency belonging to the two trains. Also in this phase two intervals are defined to establish the amount of time in which the slave listens for page messages. Default values for this parameters are equal to inquiry scan values:

T_{page_scan} = 1.28s $T_{w\_page\_scan}$= 11.25ms

After several interactions, the master and the slave enter into a substate in which the clock input to the hop selection mechanism is frozen on the same value, and they can communicate using the same hopping sequence. At the end of the page, the master enters into the *connection* state and asks to open a connection. If the slave acknowledges the request, the connection is finally established and the two devices can exchange data.

4. Initial measures and expected results

In this section, we provide some initial experimental and simulative results. The results concern the time and related probability of device discovery. The knowledge of these results is important to define the scheduling policy of a master, which must dedicate a certain percentage of its working time to device discovery and the remaining time to serve slaves applications.

4.1 The experiments

The whole system was developed on Linux platforms. The master role is played by a workstation equipped with a Pci-Pcmcia adapter from Texas Instruments and a 3COM Bluetooth card. The slave role is played by one laptop equipped with the same card. Linux operating system contains the *Official Bluetooth Protocol Stack (BlueZ)*, necessary to program card drivers. In all the experiments we have used the `ftime` routine of the Linux operating system for measuring the time spent by the master machine in discovering a new mobile device.

In all these experiments the master is completely dedicated to executing the discovery procedure, i.e. it is always in the inquiry state. This situation is the most advantageous policy of device discovery. It should be considered an upper bound that allows a first verification of the feasibility of tracking mobile users with BIPS. In the various experiments we have only changed the working hypothesis of the slave. The time that we measured in this first set of experiments is an interval that begins just before the master enters into the inquiry state and that ends when the master receives the answer from the slave to the inquiry message. Due to space limitations, we only report the results of the most significant experiment. In this experiment the slave alternates the periods of inquiry scan and page scan. This is the way in which the slave is programmed, inquiry scan mode is necessary for the slave in order to be discovered by the master, while page scan mode enables the slave to synchronize itself with the master and after to accept a connection request. In both cases the slave listens for inquiry/page messages for an interval of 11.25ms, as long as the default values specified in the Bluetooth standard. The table illustrates the outcome of the experiment.

Starting Train	Case No.	$T_{average}$
Same	236	1.6028s
Different	264	4.1320s
Mixed	500	2.865s

The table shows the average discovery time that has been measured on 500 inquiry trials. The first column of the table shows the situation when the starting frequency train used by master and slave are the same or not. The last row indicates the average result of all trials. It is worth noting that the probability that the master and the slave start on the same train is close to 50%. We note that since the slave cannot be completely dedicated to being discovered, but it should also spend some time exchanging data with the master, the average discovery times are closed to the results obtained in the case in which the slave is continuously listening to inquiry messages. However, the situation measured in this set of experiments is not realistic. We would like to measure the case in which BIPS tracks several mobile devices so that we could evaluate the real master's performance in device discovery. For defining the master scheduling parameters before the actual implementation of BIPS, we used a simulation environment to obtain some results for the discovery phase with more than one slaves. This is illustrated in the next subsection.

4.2 Simulation environment and results

To establish a correct hypothesis for master behaviour in BIPS, we have developed a Bluetooth extension to the VINT project network simulator "ns2" [5]. Our extension is based on BlueHoc, the ns2-based simulator released by IBM [6]. We have enriched BlueHoc with a mechanism for handling collisions that might arise during the establishment of a link. In the simulated scenarios, we considered a single piconet in which only one device always assumes the role of master, while the others assume the role of slaves. In particular, slaves are always in inquiry scan mode and they start listening on frequencies of train A. Since discovery time depends on the number of slaves in the piconet, we performed our simulations varying this number in the range [2,20]. In the unique simulation that we report here for space limitation, the master alternates device discovery and connection management and it transmits inquiry messages using only train A. It is reasonable to hypothesize that the master is dedicated to device discovery for 20% of its operational cycle. For this reason we have fixed the length of the inquiry phase to 1 second and the total period to 5 seconds. We can note that, when the maximum number of slaves in the piconet reaches 10, on the average the master succeeds in discovering about 90% of these slaves in the first 1 second. Only in the second operational cycle it discovers 100% of the slaves. When the number of slaves increases by 5 (i.e., 15-20), the slaves are all discovered in 2 cycles. (see Figure 2). In a real scenario, the starting trains for master and slaves cannot be defined by the programmer, so they have a 50% probability of starting with the same train. For this reason, we have to fix the length of the discovery phase to more than 3.84s. Indeed, the master spends 2.56s on the first train, discovering all the slaves listening on the same train. Then, as we have seen in the first simulation, 1.28s are enough to discover 90% of the remaining slaves listening on the other train (if they are less than 10). In summary, when there are 20 slaves in a master coverage area, if the master spends 3.84s on device discovery and the slaves are programmed for enrolling in the piconet, then the percentage of mobile devices discovered is about 95%. In fact 50% of the slaves belonging to the first train are completely discovered, and 90% of the remaining 50% will be discovered with the second train.

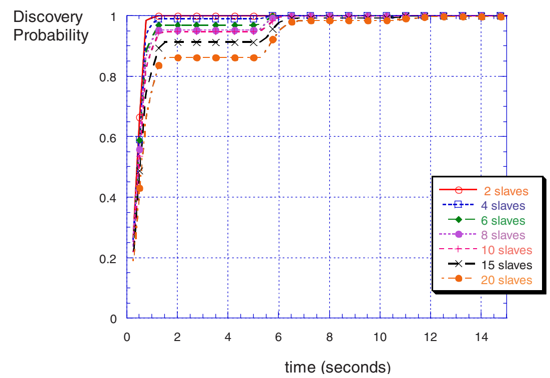

Figure 2. Inquiry and connection management.

5. Conclusions

As shown in the previous section, if the master spends 3.84s on device discovery and the slaves are programmed to enroll in the piconet, then in the average case 95% of the slaves will be discovered. This represents a percentage that is satisfactory for BIPS. To establish the percentage of working time that a master dedicates to device discovery, we have to estimate the average time that an average mobile user spends in crossing a piconet. Considering that a mobile user normally walks with a speed in the range [0, 1.5] meters per second and that the diameter of the coverage area is about 20m, we can estimate that the average walking user will spend 15.4s in the piconet (20m:1.3m/s). This time represents the length of a complete operational cycle of a master. Hence, the master will dedicate a continuous slot of 3.84s for device discovery and the remaining 11.56s for serving the slaves. The average load of tracking service in BIPS is about 24% of the operational cycle.

6. References

[1] R. Malladi and D.P. Agrawal, "Current and Future Applications of Mobile and Wireless Networks", *Comm. ACM*, *vol. 45*, *no. 10*, Oct. 2002, pp. 144-146.
[2] C. Bisdikian, "An Overview of the Bluetooth Wireless Technology", *IEEE Comm. Magazine*, vol. 45, no. 10, Dec. 2001, pp. 86-94.
[3] Bluetooth web site: http://www.bluetooth.com.
[4] E. W. Dijkstra, "A note of two problems in connexion with graphs", *Numerische Mathematik*, *1 (1959),* pp. 269-271.
[5] The VINT Project, The ns Manual, http://www.isi.edu/nsman/ns/, 2002.
[6] IBM, BlueHoc: Bluetooth Ad Hoc Network simulator, Version 1.0, http://www.124.ibm.com/developerworks/projects/bluehoc, June 2001.