Design and Analysis of a Bluetooth-Based Indoor Localization System⋆

Raffaele Bruno and Franca Delmastro

IIT Institute CNR
Via G. Moruzzi, 1 - 56100 Pisa, Italy
{raffaele.bruno,franca.delmastro}@iit.cnr.it

Abstract. The emerging context-aware applications relying on the wireless networking, require the availability of network services that provide even simple notions of context as position and identity. In the last few years several localization and tracking systems for indoor environment have been proposed. However, these systems are often very expensive since they require the installation of dedicated infrastructures of sensors based on radio, ultrasound or infrared technologies, to determine the user position. This paper aims to propose and analyze a novel indoor localization and tracking system based on the Bluetooth technology. This is a short-range, low complexity and low cost wireless technology that supports explicit device discovery services. The proposed localization system fully integrates these device discovery procedures into the infrastructure-based network of Bluetooth access points that provides the wireless connectivity inside a building. This design approach guarantees a cost-effective solution for the positioning service implementation. In this paper we have extensively investigated through simulative analysis the performances of the proposed localization system, showing that it has a negligible overhead, still providing a timely update of user location.

1 Introduction

Currently, the research interest in the design of technologies, systems and protocols that enables the deployment of new context aware applications within distributed and mobile computing systems, is rapidly growing. The term *context* is used to indicate the set of information that describes both the identity, activity and state characterizing a user, or a group of users, and the state of the objects that interact with them. The use of simple forms of context, such as time and identity are already an integral part of mobile computing. The proliferation of wireless networking and sensor technologies are enabling portable computers and handheld devices to be used in more places and to have a greater awareness of the dynamic context. In the last few years a wide class of new context-aware applications that use elementary notions of context (e.g. location, proximity and activity) have been developed [1]. All these proposals and prototypes demonstrated that one of the basic descriptions of the context is the physical location

---

of objects and users. The localization systems proposed so far differ in many parameters: techniques and physical phenomena exploited, technologies which are based on, the precision of the position information and so on. The interested reader can find an exhaustive taxonomy with a brief description of main localization systems for mobile computing applications in [2]. The aim of this paper is to propose and analyze a novel system that provides position information of mobile users inside a building based on the Bluetooth wireless technology [3]. Hereafter it will be indicated as *BIPS* (*Bluetooth Indoor Positioning System*). The Bluetooth technology is one of the most promising and cost-effective candidates to provide short-range wireless connectivity for mobile users. Key features of this technology are the robustness, the low complexity, the low power consumption and the short range (below 10 meters).The choice of the Bluetooth technology as base for the BIPS system is motivated by the use of this technology as a *de facto* standard to construct Wireless Personal Area Networks. The core of the BIPS system is constituted of a network of Bluetooth access points, acting as masters, distributed inside the building. The Bluetooth access points are interconnected through a dedicated wired network (e.g. an Ethernet LAN) and their operations are coordinated by a central server machine. A single room of the building is the granule of the position information that BIPS considers. We define a single room as the space that can fit into a circle of 10 meter radius since this is the dimension of the greatest coverage area of a Bluetooth piconet. The BIPS system is devoted to track the mobile users that walk (or stand) inside the building at a low speed. The user localization and tracking services are fully integrated within the same network infrastructure that provides the wireless connectivity for the data transfer, guaranteeing a cost-effective implementation of the localization system. The other proposal that fully integrates the tracking system with the same infrastructure that provides the wireless networking is the RADAR system developed by the Microsoft Research group [4] and based on the IEEE 802.11 Wireless LAN technology [5]. The RADAR system achieves a better accuracy than the BIPS system (about$3-4meters$ of precision), paying the cost of higher complexity, since it requires either scene analysis (i.e., an *a priori* measured map of the radio signal strength inside the building) or multi-lateration (i.e., the triangulation of distances between at least three base stations). The paper is organized as follows. In section 2, the Bluetooth technology is briefly outlined. Particular attention is dedicated to describe the support for device discovery through the inquiry and page phase. Section 3 details the BIPS-system organization and an accurate analysis of the device discovery performances. The performance evaluation and comparison between the solutions based on different discovery approaches is showed in section 4. Concluding remarks and considerations on the ongoing work are presented in section 5.

2   Bluetooth Basics

The building block of a Bluetooth network is the *piconet*, a small cluster of devices that share a common physical channel, i.e., they are synchronized to the

same clock and adopt the same frequency hopping sequence. Within the piconet, one device assumes the role of *master*, and all the other devices associated to the piconet play the role of *slaves*. The slaves derive the channel-hopping sequence as a function of the master's clock and address. The transmission scheme adopted in the piconet is the Time Division Duplex (TDD) scheme. The channel is divided into time slots, each $625\,\mu s$ long. To establish a connection it is required that the devices agree on the Bluetooth role to assume, either master or slave, and exchange their 48-bit address (BD_ADDR) and 28-bit native clock (CLKN). To achieve this the Bluetooth specification standardized a link establishment process constituted of two different phases. The first initial phase is named *inquiry*, and it allows the *inquirer* to discover the identity of possible slaves. The second final phase is named *page*, and it corresponds to initial connection setup, in which the *pager* informs the *paged unit* about its identity and imposes its clock as the piconet clock. In both phases the master and the slaves perform different actions that we will briefly summarize in the following subsections. At the end of these phases, the devices are synchronized on the clock of the inquirer/pager that becomes the master of the piconet. Then the connection is established and the devices can start exchanging data.

2.1   Inquiry Phase

A master begins the device discovery by entering into the inquiry state. In this state, the master broadcasts messages using a pool of 32 frequencies called inquiry hopping sequence, splitted into two 16-hop parts (train A and B). On every even slot, the master sends two ID packets switching between two frequencies of the same train every $312.5\,\mu s$. Thus $10\,ms$ are required to scan a single train, and it 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's switches must take place. A slave that wants to be discovered enters in the inquiry scan state, during which it listens for ID packets on the same 32 dedicated frequencies used from the master. The slave changes listening frequency every $1.28\,s$ and it keeps listening on the same frequency for the $T_{w\_inquiry\_scan}$ time, that 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 is not longer than $2.56\,s$. If $T_{inquiry\_scan} = T_{w\_inquiry\_scan}$ then the inquiry scan activity is continuous. Both interval values can be changed and they are set by default as $T_{inquiry\_scan} = 1.28\,s$ and $T_{w\_inquiry\_scan} = 11.25\,ms$.

2.2   Page and Connection Phases

During the inquiry phase, the inquirer collects the BD_ADDRs and CLKNs of all the units that acknoledge the inquiry messages. However, the inquiry messages sent by the inquirer don't carry any information about the sender. Therefore, a further phase is needed by the master to establish a connection with a slave. This is the page phase, in which the master tries to capture a specific slave sending page messages, containing its own synchronization data. As in the inquiry phase,

the master uses 32 frequencies belonging to the page hopping sequence, splitted into two 16-hop parts (train A and B). Since the master does not know when the slave will enter into the page scan substate, each train must be repeated at least N_{page} times. N_{page} is defined depending on the T_{page_scan} value, which is the interval between the beginning of two consecutive page scans. Default values for page scan parameters are equal to the inquiry scan default values: $T_{page\_scan} = 1.28\,s$, $T_{w\_page\_scan} = 11.25\,ms$. At the end of the page phase, the master enters into the connection state and sends a connection request to the slave. If this agrees and acknowledges the request, the connection is established and the devices can start exchanging data packets.

3   BIPS System Design Principles

BIPS is an indoor positioning service designed to track mobile users that may be in motion inside a corporate building. Each mobile user is represented by a handheld device equipped with a Bluetooth interface in order to interact with the static infrastructure of the BIPS system. This is constituted of a set of Bluetooth access points, placed in fixed positions inside a building and interconnected through a dedicated network (for instance a wired Ethernet LAN). They always assume the master role in order to discover and enroll mobile devices in their coverage area. The intelligence of the BIPS system is located in a central machine, hereafter denoted as BIPS-server, that performs the coordination of the masters to locate users and track their movements[1]. Each access point executes two main tasks: one is the discovery of the units entering into the radio coverage area of its Bluetooth interface; the other one is the data transfer towards/from users that are associated to its piconet. The BIPS system should guarantee to achieve the timely update of the user's position without negatively affecting the throughput of data connections. It is straightforward to observe that these two goals are complementary since a prompt user-movement tracking can be obtained performing frequently the device discovery procedures. In subsections 3.1 and 3.2 we will analyze the overhead introduced by the device discovery task inside an isolated piconet. Then in section 4 the impact of the device discovery on the data transfer will be quantified. It is worth pointing out that the main objective of this paper is to design an effective discovery scheme that minimizes the localization system overheads preserving the promptness of the tracking service.

3.1   Device Discovery

The BIPS system implements the localization and tracking functionalities through the coordination of the device discovery phases executed by the master units. The performance evaluation of the overhead introduced by the execution

---

[1] The BIPS-server may implement further functionalities in addition to the user position update, as user authentication, mobile users lookup and user profiling. The reader is referred to [6] for a detailed discussion about the services that can be implemented in the BIPS system.

of the device discovery task will be very usefull in the design of the BIPS system. Because of the users' mobility, some links may disappear since the devices exit from the radio visibility of the master. In the same way new links can be established when two nodes are close enough to allow the receivers to listen to the transmitted radio signals. The localization of the users can be realized by tracking the links that disappears and the new links that are established. An efficient design of the coordination among masters' operations to promptly identify disconnections and, at the same time, to open new connections, cannot be done without precisely identifying the limitations of the device discovery procedures standardized by the Bluetooth technology. The link establishment process requires that each device becomes aware of the other unknown devices that enter into its radio coverage area. The connection establishment procedure described in section 2 are designed to enable a device to discover which other devices are in range, and their addresses and clocks. In previous studies the Bluetooth device discovery procedures have been analyzed only when the link establishment is accomplished in an ad hoc fashion without any coordination, namely, without any explicit master or slave role pre-assignment. In [7] a symmetric mechanism that guarantees an ad hoc point-to-point connection between two devices by forcing the nodes to independently alternates between Inquiry mode and Inquiry Scan mode, was proposed. However, the effectiveness of the proposed protocol was evaluated only in a simplified scenario where all the nodes are in radio visibility of each other, i.e., the nodes form a single-hop network. In [8] the protocol defined in [7] was generalized to the case of a multi-hop networks and the performance of the Bluetooth device discovery procedures where evaluated in terms of the percentage of neighbors discovered within a given discovery-time interval. These previous works clearly show that the device discovery procedures standardized in [3] are robust against interference due to the concurrent operations of nodes in highly dense networks, but they introduce significant delays in the neighboring discovery process. In this section we investigate the Bluetooth discovery procedures from a completely different perspective. Specifically, we will consider a scenario where the devices' operations are coordinated and there is a pre-assignment of the master and slave roles. The designated master will be the *active inquirer*, namely the node in charge on scanning its coverage area to discover the presence of other units willing to connect to it, whereas the designated slaves will be the *passive listeners*, waiting for being discovered from the master closest to them. There are many cases in which is reasonable and advantageous to construct the Bluetooth network in an infrastructure-based fashion, and to select a-priori the nodes that will act as masters. For instance, in a Bluetooth-based indoor tracking system, we can assume that there is a network of *inquirers* that actively search the users and track their movements. The simulations we have conducted are aimed to investigate the amount of resources that an inquirer should spend in gathering an exact knowledge of the units present in its coverage area, at least with a very high probability. Specifically, a first set of simulations were conducted to evaluate the setup connection delay in realistic conditions, without the biasing factor of periodic alternation between Inquiry and Inquiry

Scan mode. Moreover, these simulations allow us to better understand the weight of the different phases in the link establishment process. A second set of simulations has been carried out to determine the amount of time the inquirer should dedicate to the device discovery task to localize a given number of neighbors. These simulations provide an estimate of the efficiency of the Bluetooth device discovery procedures, and allow us to better determine how many mobile units can be reasonably tracked by a single inquirer.

3.2   Numerical Results

In all the simulations carried out the master is completely dedicated to execute the discovery procedure, i.e., it performs the inquiry phase until all the slaves have been discovered. The units to be discovered act as slaves and they start their operations in inquiry scan mode. To analyze the most favorable case from the perspective of device discovery, the slave executes a continuous scanning. As soon as the master receives a FHS packet from the slaves it goes to page mode to enroll the new discovered unit in its piconet. In the same way, as soon as the slave sends an FHS reply it goes to Page Scan mode to open a connection with the master. The Inquiry and Page mode parameters were selected according to the default values. In particular $N_{inquiry} = 256$ and $N_{page} = 128$. As explained in the previous section, the first set of simulations is aimed to evaluate the link setup delay between a couple of Bluetooth units. Fig. 1 shows the time distribution of the link setup delay. The shape of the distribution clearly highlights that the link setup delay is uniformly distributed into two separate ranges, the lower one is the range around $[0.1s, \ldots, 0.7s]$, instead the upper one is the range around $[2.6s, \ldots, 3.4s]$. The explanation of this particular distribution is straightforward. Let us consider the case in which both master and slaves are tuned to a frequency picked from the same train, and calculate the maximum time required to setup the link. The slave receives the first inquiry message that triggers the back-off timer at most within the interval needed to the inquirer to completely scan the train (i.e., $10\,ms$). The maximum back-off length is 1023 time slots (i.e., $639.375\,ms$). After the back-off timer expiration, another scanning interval of $10\,ms$ is needed to the inquirer to receive the FHS packet. The page phase that immediately follows the inquiry phase has a significant lower duration. Considering that both the master and the slave starts the Page mode and Page Scan mode respectively picking up a frequency from the same train, a 16-frequencies scan interval is needed to achieve the coarse synchronization between the master and the slave. When this coarse synchronization is obtained the remaining part of the connection setup requires a few couples of time slots to be completed[2]. The sum of these contributes is equal to $673.425\,ms$. The second lobe of delay distribution is obtained when the master and slaves pick a frequency from opposite trains and a further startup delay of $2.56\,s$ is added to the link establishment process. The numerical results showed

---

[2] The interested reader is reminded to the Bluetooth specification [3] for the details of message exchange during the Page Response mode and at the beginning of the connection establishment.
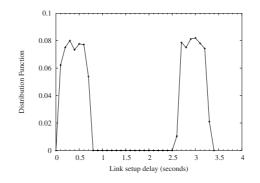
Fig. 1. Link setup delay.

in Fig. 1 quantify the time spent by a master in opening a single connection. It is necessary to investigate how the device discovery procedures scale up with the number of slaves that should be discovered (i.e., the number of connections that should be opened). To this end, we performed a second set of simulations where master's and slave's behavior is the same of the first set of simulations, but the number of slaves ranges between 2 and 15. The results obtained are showed in Fig. 2. The figure shows that less than $1.5\,s$ is always sufficient to establish a connection towards half of the slaves that are close to the master. This time reduces to $0.8\,s$ for piconets smaller than 7 slaves. The remaining slaves have to wait $2.56\,s$ to be discovered because the master has to change its frequency train. All the N slaves are connected to the master within $4\,s$. The percentage of discovered slaves shows a step-function shape due to the random selection of the train used during the inquiry phase. Specifically, the 50% of slaves start to listen on a frequency picked from a train different from the one used by the master. In this case the link setup delay is due to the second part of the Fig. 1. This implies that at least $2.56\,s$ are needed before the slaves are reachable by the master, therefore on average half of the slaves are reachable before $2.56\,s$, whereas the remaining half is reachable only after the first $2.56\,s$. Two useful observations can be taken comparing the numerical results showed in Fig. 1 and Fig. 2. The first one is that the back-off part of the link establishment process is the main cause of delay in forming the Bluetooth connection. The second one is that the back-off range introduces an excessive spread of the slave replies since, even in dense piconets, the collision probability is negligible because of the not-correlated nature of the slaves' native clocks. To summarize, the design of a localization system, relying upon the Bluetooth device discovery procedures, has to deal with three main intrinsic technological limitations: i) the use of two separated frequency trains that can introduce a delay of up to $2.56\,s$ during the link establishment; ii) the use of a long random back-off delay that increases the average link setup delay of about $300\,ms$; and iii) the use of a broadcast inquiry message that doesn't carry any information about the inquirer. In the following section, we will show that, by a limited tuning of the Inquiry mode parameters,
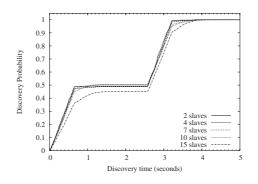
Fig. 2. Device discovery efficiency for a piconet of N slaves.

and designing a localization scheme relying only upon the paging phase, these drawbacks can be significantly overcome.

3.3 Page versus Inquiry

Observing the results obtained from previous simulations, we can notice that the overhead introduced by the device discovery task based on the inquiry phase negatively influences the user tracking performances. For this reason we propose and analyze a scheme for implementing the user positioning service relying on the page phase, in order to reduce the impact of the device discovery on other services like data transfer. Hereafter, we refer to the proposed localization system as $BIPS_P$ system. Specifically, the $BIPS_P$ system assumes that the BIPS-server commands the execution of a new page phase when a user disconnection is discovered by the network access points. This solution requires that masters and BIPS-server exchange information about the current status of their piconets, e.g. user disconnections and associations, allowing only the discovery of known devices. For this reason it is always necessary the execution of a periodic inquiry phase with a period much greater than page's one. The $BIPS_P$ system will be compared with a reference architecture in which the localization task is performed by the network access points without exploiting masters' communication and coordination. Specifically, each master performs the discovery procedures independently from the other masters. Information about user disconnections are not exchanged. Since masters don't know the identity of the devices to track, and when it is necessary to execute a new discovery phase, this scheme has to rely on the inquiry phase (outlined in section 3.1). For the sake of brevity this second solution will be denoted as $BIPS_I$ system. The remaining of this section details the $BIPS_I$ and $BIPS_P$ system operations, considering only the overhead introduced by the discovery of known devices. A critical design choice for the localization system is to define the rules adopted by the master and slaves to become aware of a link disconnection due to a slave departed from the radio coverage area of the piconet to which it is associated. The Bluetooth specification introduces a supervision timeout, named *supervisionTO*, used by both masters

and slaves to monitor link losses. Specifically, if a device doesn't receive any correct packet which is destined to itself before the supervisionTO expiration, then the connection is automatically closed. The default value is $20\,s$, but it is an exceedingly high value to define any meaningful localization and tracking service. Therefore the supervisionTO should be modified to better deal with the users' dynamics. We have considered an ideal scenario in which no packet losses are allowed during the Connection state, hence the master can assume a slave as disconnected when it doesn't answer immediately to its polls. As far as the slaves' behavior, we have considered only active slaves that are involved in high loaded data connections served with a Round Robin scheduler. Therefore, each slave is polled at least once per polling cycle[3]. Hence the supervisionTO needs to take into account only the maximum size of time interval in which the master suspends serving data connections to execute the device discovery phases. In section 4, the proper value for the slaves' supervisionTO will be evaluated. In the following subsections we analyze the parameters values used for simulations in both systems.

BIPS$_\mathbf{I}$ System. Each master periodically performs an active scanning of its coverage area, broadcasting inquiry messages. The period between two successive inquiry phases is referred to as T_{rep}, while the lenght of each of them is indicated as W. The ratio between W and T_{rep} represents the percentage of channel resources dedicated to the discovery task. This overhead is almost independent of the user mobility since the discovery phase is periodically performed by the masters. To guarantee that a Bluetooth device, in Inquiry Scan mode, has the chance of listening to at least one inquiry message , the W period is equally divided into two parts dedicated to scan both train A and B. This requires the modification of the $N_{inquiry}$ default value (i.e., 256). On the other hand, when a user realizes to be disconnect from the piconet (thanks to the supervisionTO expiration), it immediately goes to Inquiry Scan mode. As soon as the master receives the reply to its inquiry messages from a new device entered into its coverage area, it tries to connect to it. To this end, it performs immediately a page procedure. The duration of this phase should be limited to reduce the time the master is not serving data connections. We select it equal to W. Now we are able to estimate a conservative value for the supervisionTO, which should be used by the slaves[4]. In fact, the maximum delay between two consecutive master's polls, under the ideal conditions of no packet losses during the Connection state, is lower bounded by $2 \times W + C^7_{MAX}$, and the supervisionTO should be selected greater than that value.

---

[3] The maximum length of a polling cycle for a Round Robin scheduler used in a piconet with N slaves (C^N_{MAX}) when only DH_5 packets occupy the channel (i.e., the maximum length baseband packet) is $C^N_{MAX} = 2 \times N \times DH_5$. In a piconet with 7 slaves we have $C^7_{MAX} = 8.75\,ms$.

[4] It is worth reminding that the slaves' promptness to discover the disconnection from its master depends on the supervisionTO.
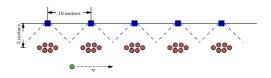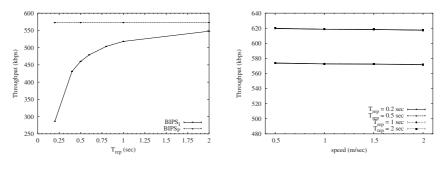
Fig. 3. Simulations scenario.

BIPS$_\mathbf{P}$ System. The masters perform only page procedures to track the movement of the users. Specifically, as soon as a master identifies a link disconnection, it informs the BIPS-server via a specific "alarm" message that carries the identity of the lost user, say U_L, and the up to date clock information available at the master. These information are delivered to all the masters "close" to the one that originated the alarm. The notion of proximity can be implemented in several ways. The simplest is the physical proximity, i.e., the masters nearest to the one that originated the alarm are designated to search U_L. However, more complex policies can be thought if the user profile is available at the BIPS-server, e.g. the most visited access points, the final destination etc.. Then, these designated masters begin the execution of a page phase, trying to open a connection towards U_L. This phase lasts for W and it is periodically repeated with period T_{rep}. On the other hand, when a user realizes to be disconnected from the piconet (thanks to the supervisionTO expiration), it immediately goes to Page Scan mode. As soon as a master succeeds in finding the user, it notifies the BIPS-server about the detection, so that it stops the other designated masters to seek for U_L. The slaves' supervisionTO is chosen as $2 \times W + C^7_{MAX}$, to be as much possible close to the BIPS$_I$ system operations.

4   Performance Evaluation

To compare BIPS$_I$ with BIPS$_P$ system we have identified a realistic scenario, that is showed in fig3. Specifically, we consider a corridor, where 5 Bluetooth access points are located at a distance of 2 meters from the ceiling, spaced by ten meters. Each access point has 6 static slaves associated to its piconet. Highly loaded data connections are active towards each slave (e.g. FTP data transfers), served according to a Round Robin polling scheme. A single user moves with a constant speed v, and no active data connections are active towards the mobile user. The length of the TCP payload has been selected equal to 1000 bytes that allows us to have a maximum TCP throughput per piconet around 620 Kbps [9]. During all the simulation we have assumed $W = 100\,ms$, a time long enough to scan five times both the train A and B[5]. The performance indexes we investigated, are aimed to quantify both the systems promptness in tracking the user movement and the impact of the discovery task on the data connections performances (e.g., connections throughput, MAC delay and End-to-End delay). We present a comparison between the two systems about the

---

[5] This choice implies that the supervisionTO cannot be lower that $210\,ms$. In all the simulations we have assumed the supervisionTO equal to $250\,ms$.

(a) TCP throughputs for the user speed equal to $1\,m/s$.

(b) BIPS$_P$ throughputs (TCP and L2CAP).

Fig. 4. Throughputs comparison.

TCP throughput per piconet versus the T $_{rep}$ parameter, the link setup delay after a link disconnection and the MAC and End-to-End delays. The MAC delay represents the transmission time required by a packet at the head of the queue, and the End-to-End delay is equal to the MAC delay plus the queueing time. The BIPS$_I$ and BIPS$_P$ system operations were implemented in the *ns2* simulator [10], extended with the IBM module *BlueHoc* [11]. All the simulative results showed hereafter are presented with a 95% confidence interval.

4.1   Simulative Results

In this section we describe the simulative results related to the performances of the BIPS $_I$ and BIPS$_P$ systems using the scenario showed in Fig. 3. Fig. 4a) shows the throughput per piconet achieved by the TCP connections versus the T $_{rep}$ parameter, when the user speed is $1\,m/s$. For the BIPS$_I$ system the same results are obtained with different user speed values. This is explained observing that it executes the device discovery procedures periodically, independently of the user behavior. We vary the T$_{rep}$ parameter to evaluate how the frequency, used by the system to perform the localization task, affects the data connections throughput. The buffer size at masters and slaves queues has been selected large enough to avoid packet losses due to buffer overflow. It is straightforward to notice that the throughput scales up rapidly with the T_{rep} value, since the time dedicated to the discovery phase decreases. For instance, the choice $T_{rep} = 1\,s$ implies that 10% of the channel is occupied by the execution of inquiry phases, with the remaining 90% dedicated by the master to serve data connections. Instead $T_{rep} = 0.2\,s$ implies that only the 50% of the channel time is dedicated to serve data connections. In the BIPS$_P$ system the masters page the mobile user only when it is disconnected, and they continue to periodically page it until it is not enrolled again in a piconet (with period T_{rep}). Therefore, we expect that the T_{rep} has a negligible impact on the throughput performance if a few pages are enough to establish the connection. The numerical results clearly confirm that the throughput dependency on the T_{rep} parameter is negligible. On the other

hand, because of the user sojourn time into the radio coverage area of a piconet is inversely proportional to its speed, the frequency, used by the masters to perform the tracking task, increases with the user speed. Hence, we expect that, differently from theBIPS$_I$ system, the speed value may affect the throughput performance. However, Fig. 4b) clearly highlights that the throughput decrease is negligible, also when the user mobility is high (e.g. $2\,m/s$). Figure 4b) shows the throughput per piconet achieved by the L2CAP and TCP connections versus the speed of the mobile user. We investigate the throughput performance varying the T_{rep} parameter. The TCP throughput is represented by the lower curve. The difference with the L2CAP throughput is due to the presence of the TCP acknowledgment traffic that flows from the slaves towards the master in the active L2CAP connections. It is worth pointing out the BIPS$_P$ system adapts its behavior to the user mobility, minimizing the overhead introduced by the tracking service. Therefore, even in low mobility scenarios (e.g. $0.5\,m/s$) the piconet throughput is always higher in BIPS$_P$ than in BIPS$_I$ ($580\,$Kbps against the $550\,$Kbps achieved when $T_{rep} = 2\,s$). To complete the performance analysis of data traffic, Fig. 5(a) and 5(b) show respectively the MAC and End-to-End delays for the L2CAP packets versus the T_{rep} parameter for the user speed equal to $1\,m/s$. In [12] it was proved that the aggregate MAC delay of L2CAP packets measured in asymptotic conditions (i.e., when there is always a queued packet in the master's and in the slaves' local queues) for a Round Robin scheduler, depends only on the number of slaves and on the average number of time slots needed to transmit the L2CAP packet. Instead, the End-to-End delay mainly depends on the buffer size. However, the BIPS$_I$ system introduces a further W delay to fragmented L2CAP packets, which fragments are not all delivered within the channel time between two successive inquiry phases. It is straightforward to observe that the probability to introduce this fixed delay grows by decreasing the T_{rep} value. This implies that both MAC and End-to-End delays significantly decreases as the T_{rep} value increases. At the opposite, the BIPS$_P$ system shows a significant improvement compared with the BIPS$_I$ system. In fact the adaptive policy used to decide when to page a mobile user, together with the exploitation of the page's properties, guarantees to keep the packet delays almost constant. In this case, both MAC and End-to-End delays are almost independent of the T_{rep} value, and further results have demonstrated that they are slightly affected by the increase of the user speed. Even in low mobility scenarios (e.g. $0.5\,m/s$) the measured MAC and End-to-End delays are always lower in the BIPS$_P$ system than in the BIPS$_I$ system. As final result we have to investigate the effectiveness of each scheme in providing a prompt update of the user position. To quantify this performance we have identified two indexes: the first is the average time the system requires to track the new user position, say T_{blind}^M; the latter is the average time the user is not associated to any piconet, say T_{blind}^S. Specifically, T_{blind}^M is the average time between two events: the first is a master becoming aware of the mobile user disconnection, and the second is a master opening a connection towards the mobile user. In the same way, T_{blind}^S is the average time between two events: the first is the mobile user becoming aware of not being associated
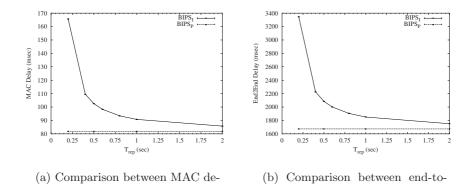
(a) Comparison between MAC delays

(b) Comparison between end-to-end delays

Fig. 5. MAC and end-to-end delays.

anymore with the master, and the second is the mobile user enrollment into another piconet. The T_{blind}^M value assesses the system promptness in discovering the mobile user location, whereas the T_{blind}^S value measures the user perception of the network failure. Fig. 6 shows the comparison between T_{blind}^M in the two systems. It is straightforward to notice that, for the BIPS$_I$ system, as the T_{rep} increases, also the delay between tracking updates increases. In fact, at least two successive inquiry phases are required to open a connection: the first one to trigger the back-off selection, the second one to trigger the page phase. Hence, T_{blind}^M and T_{blind}^S cannot be lower than the T_{rep} duration. Nevertheless, the inquiry and page procedures, thanks to the very fast hopping scheme adopted, are very robust against the interference due to the contemporary data transmissions coming from overlapping piconets. Therefore, the probability of connection failures after the back-off expiration is very low. This implies that usually two inquiry phases are enough to establish a new connection. Hence, the BIPS$_I$ system introduces an overhead independent of the mobile user speed, but depending only on the T_{rep} value. All the performance indexes related to data connections worse as the T_{rep} value reduces, since the time dedicated to locate the mobile user increases. For the BIPS$_P$ system, to better explain the numerical results, we have to detail the timing of alarms generation in the masters. Specifically, when a master realizes that a user departed from its piconet, it sends an alarm to the BIPS-server only after a supervisionTO. This guarantees that the masters, designated by the BIPS-server to discover the new user position, begin the page procedure when the user is already seeking for a new network access point. However, this choice also implies that the first page phase is executed by the designated masters with a starting delay of the supervisionTO. Hence, the T_{blind}^M value cannot be lower than the suspervionTO. The numerical results plotted in Fig. 6a) clearly show that in both systems, the T_{blind}^M value increases as the T_{rep} increases. However, since the page procedure is less time-consuming than the inquiry (there isn't a back-off delay before the slave responses), the T_{blind}^M value is always lower in BIPS$_P$ than in BIPS$_I$. As we can see in Fig. 6b), the T_{blind}^S value is always about

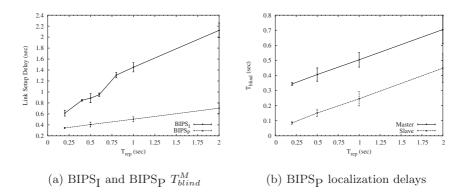(a) BIPS$_I$ and BIPS$_P$ T^M_{blind} (b) BIPS$_P$ localization delays

Fig. 6. Localization delays.

$200\,ms$ lower than the T^M_{blind} since the slaves became aware of a link disconnection after the supervisionTO delay. Simulations with different speed values have provided the same results, and are not reported here. To conclude, adapting the discovery phases lenght to the user mobility, we can guarantee to keep the overhead due to the execution of the localization service very low. Hence, there is a negligible impact on the performances perceived by the users data connections. On the other hand the use of the page procedure guarantees the reduction of the setup delay needed to open new connections towards the mobile user.

5   Conclusions and Future Works

In this section we have proposed a solution to offer an indoor localization service that is fully integrated in the infrastructure that guarantees the wireless networking inside a building. Our proposal is based on the exploitation of an emerging technology for short range and ad hoc wireless connectivity: the Bluetooth technology. We have proposes a scheme that adapts, to the user mobility profile, the frequency used by network access points to perform user tracking. The localization service has been implemented using the paging procedures defined in the Bluetooth standard. This design approach guarantees a cost-effective solution for the implementation of the localization service. Our simulative investigations have shown that throughput and packet delays are negligible affected by the load generated by localization service. In addition, our scheme is able to guarantee a prompt update of the user localization, significantly limiting the handoff delays between different piconets. During the analysis we assumed a symmetric traffic distribution for the slaves associated to the network access points. The master serves the data connection using a simple Round Robin scheduler. Further studies are required to design novel scheduling algorithms that can efficiently integrate the device discovery procedures inside the polling scheme. The final goal should be the development of an architecture that provides the wireless networking adapting the access point operations to the user mobility and traffic demands.

References

1. R. Malladi and D.P. Agrawal. Current and future applications of mobile and wireless netwoorks. *Communications of ACM*, 45(10):144–146, October 2002.
2. G. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*, pages 57–66, August 2001.
3. Bluetooth SIG. *Specification of the Bluetooth System, Volume 1–Core*. Version 1.1, February 2001.
4. P. Bahl and V. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of the IEEE Infocom 200*, pages 775–784, Tel-Aviv, Israel, March 2000.
5. IEEE Standard for Wireless LAN. *Medium Access Control and Physical Layer Specification*. P802.11, November 1999.
6. G. Anastasi, R. Bandelloni, M. Conti, F. Delmastro, E. Gregori, and G. Mainetto. Experimenting an indoor Bluetooth-based positioning service. In *Proceedings of the International Workshop on Mobile Distributed Computing*, Providence, RI, May 2003.
7. T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire. Distributed topology construction of Bluetooth personal area networks. In *Proceedings of the IEEE Infocom 2001*, pages 1577–1586, Anchorage, AK, April 2001.
8. S. Basagni, R. Bruno, and C. Petrioli. Device discovery in Bluetooth networks: A scatternet perspective. In *Proceedings of the Second IFIP-TC6 Networking Conference (Networking 2002)*, volume 2345 of *LNCS*, Pisa, Italy, May 2002. Springer.
9. R. Bruno, M. Conti, and E. Gregori. Wireless access to internet via bluetooth: Performance evaluation of the edc scheduling algorithm. In *First ACM International Workshop on Wireless Mobile Internet (WMI 2001)*, pages 43–49, Rome, Italy, July 2001.
10. The VINT Project. *The ns Manual*. http://www.isi.edu/nsnam/ns/, 2002.
11. IBM. *BlueHoc: Bluetooth Ad Hoc Network Simulator*. Version 2.0, June 2001.
12. R. Bruno, M. Conti, and E. Gregori. Bluetooth: Architecture, protocols and scheduling algorithms. *Kluwer Cluster Computing*, 5(2):117–131, April 2002.