Robotica (2013) volume 31, pp. 405–416. © Cambridge University Press 2012
doi:10.1017/S0263574712000422

Increasing wireless reliability for autonomous mobile robots

Joaquín López, Rafael Sanz*, Miguel D. Cacho and Amador R. Diéguez

Department of Ingeniería de Sistemas y Automática, University of Vigo, 36200 Vigo, Spain

(Accepted July 7, 2011. First published online: August 8, 2012)

SUMMARY

There is an increasing popularity of mobile robot applications over the Internet. Robots need some kind of wireless communication to receive commands and transmit information to users. This paper describes the problems encountered to keep the mobile robots connected and the solutions adopted. The first problem was related to delays and the very low throughput that occur when the robot roams to a new access point. Most commercial systems currently installed implement proprietary solutions with different behaviors. Here, a simple and device-independent solution for mobile robot applications is proposed. The second problem is the lack of wireless coverage in some areas of buildings. The solutions adopted for these problems are based on coverage maps.

KEYWORDS: Mobile robotics; Wireless LAN; Roaming; Coverage maps; Navigation.

1. Introduction

During the last few years, the mobile robot industry has grown in investment, research, and applications. Different degrees of autonomy are required depending on the robot application, but in most applications robots need to share information with remote users or even other robots. The information to be transferred from users to robots depends on the application and degree of autonomy. It ranges from basic motion commands in tele-operated robots to more general and abstract tasks in autonomous robots. Likewise, most applications need to transfer information, such as camera images, other on-board sensor data and reports, from robots to users while executing different tasks.[1] Therefore, a communication system between user and robot is required in most applications. Furthermore, a wireless system is required.[2] Examples include a variety of applications: office and hospital delivery, surveillance and reconnaissance, inspection,[3] home security mobile robot,[4] tour guide in exhibitions,[5] among others. Several wireless technologies and standards have been used to implement the communication system for mobile robots in real world applications.[6,7] The most prominent and common indoor topology is Wireless Local Area Networks (WLAN), which offer network services that coexist with fixed infrastructure allowing mobility and flexibility to users.

* Corresponding author. E-mail: rsanz@uvigo.es

Wireless Local Area Networks provide network services where deploying a fixed infrastructure is difficult or too expensive. WLANs can coexist with fixed infrastructure to provide mobility and flexibility to users. The primary WLAN standards are IEEE 802.11.[8] This set of standards defines both physical and MAC layer protocols[9] and specifies the Received Signal Strength Indicator (RSSI), which is a relative measure of the radio frequency (RF) energy received. This measure depends on the well-known WLAN Driver (WLAND) vendor parameters. The 802.11 protocol set, popularly known as Wi-Fi, includes wireless network standards that allow data transmission at different speeds.

Most of the research in the field of robotics regarding indoor Wi-Fi is devoted to localization systems. There are different ways to focus the localization problem utilizing many different signal properties. Some use physical properties of the signal, while others use the time taken for the signal to reach the destination node. However, in this paper, we focus on communication issues.

A number of issues arising from the use of wireless networks must be considered, among which are coverage, compatibility, interference, latency, safety, security, and costs. Wireless communication systems cover areas that depend on parameters of the system selected and the environment where they are installed. For example, Wi-Fi coverage in a building depends not only on the devices used for communication such as antennas in Access Points (AP) and clients, but also on construction materials, wall distribution, and other building characteristics.

During the last two decades, our lab has been working on several mobile robot projects such as surveillance[10] and tour guide[11] applications. In both kinds of applications, each robot can handle autonomously some daily routine tasks. For example, surveillance robots are able to patrol programmed or randomly generated paths at any time. However, robot sensor information, such as real-time images captured by a camera on the robot, can be transmitted back to the central management office via a local area network. Therefore, even though the robot can execute tasks autonomously, a communication system between user and robot is necessary in most applications. Some problems regarding real-time wireless communication need to be considered and solved, particularly roaming and coverage problems. This paper proposes various solutions to these communication problems for mobile robots in indoor environments with standard wireless facilities.

https://doi.org/10.1017/S0263574712000422 Published online by Cambridge University Press

1.1. Roaming problems

Roaming problems occur when the robot moves from the vicinity of one AP to next AP. The need to support real-time applications while wireless users are on move has given place to a new WLAN standard IEEE 802.11r. However, most wireless applications presently installed are based on IEEE 802.11 basic standards, and very few communication systems already installed implement this standard. A number of researchers have studied the roaming delay and transmission throughput for the same network[12] as well as different networks.[13] One goal of the mobile robot applications that we develop is to use, if possible, the communication facilities present in the buildings where the applications will be installed. The WLAND in the client is responsible for deciding when to scan, evaluate, and roam to a new AP. In most cases, the WLAND relies only on the RSSI obtained in the receiver from the connected AP to decide when to roam to another AP. However, with RSSI values near the threshold, there might be connection problems while the robot is closer to another AP. In this paper, a monitor that helps to solve this problem by introducing a minor impact on data traffic is presented.

1.2. Coverage problems

Coverage is a key feature in wireless automation facilities with mobile robots that have a great impact on distance and data rate. Different coverage problems can arise. Direct and reflected transmissions (called multi-path transmission) can appear and combine with different delays and amplitudes (called the delay spread). Reflections are time-shifted, with distances and rate varying in indoor environments. In addition, the received signal could be completely cancelled. This effect is known as multi-path fading.

There are two different approaches to address coverage problems:

- Assuring full coverage on the robots' working area, i.e. by overlapping coverage areas of different APs. This approach removes coverage blind spots and provides switchover for roaming and redundancy in case of coverage loss.
- Providing robots with mechanisms to deal with blind spots.

The second approach is addressed here for the following reasons:

- In our mobile robot applications, we intend to use the communication system already installed, and therefore, to avoid modifying the wireless facilities in buildings.
- In some cases, developing a system to provide full coverage can be expensive.
- It is very difficult to obtain an indoor RF wave propagation model to make sure that there are no dead spots where no signal can be received.
- Even if full communication coverage is available, the malfunctioning of an AP might jeopardize the mobile robot application.

Several researchers have already addressed the task of creating wireless coverage maps for outdoor[14,15] and indoor[16,17] environments. However, because their purpose was not a mobile robot application, their research focused on the use of RF propagation models to estimate RF power and relate it to the RSSI of the WLAND. However, in our mobile robot applications, the RSSI or some related value could be readily measured at different points of the environment while navigating the robots. In this paper, the following two different sets of methods are used.

- An approach based on data gathered from different points of the environment using mobile robots.
- An approach based on propagation models.

The first approach is inspired in the results of power-based wireless robot-positioning systems. Most of the research in robotics regarding indoor Wi-Fi is devoted to robot-positioning methods.[18,19] There are different ways to address the positioning problem using many different properties. Some use physical properties of the signal, while others use the time taken for the signal to reach the destination node. One common approach uses surveying of signal strength information in a particular area. This information is stored in a database that is later used to determine the location of a mobile device by a particular pattern-matching algorithm.[8] These results can be used to develop coverage maps from the data obtained by the robots.

The other approach uses different propagation models to estimate the relationship between signal strength and distance from transmitters.[18,20] Even though our problem is different, we can use the solutions provided by these methods to estimate the signal strength in different points of the environment to construct a coverage map.

The rest of this paper is organized as follows. Section 2 introduces the architecture used to control each mobile robot. Robots connect via Wi-Fi to a central server to share information with the user interfaces. The communication architecture used (Java Inter-Process Communication or JIPC) will be presented in Section 3. The roaming problem is explained in Section 4. The two following sections describe different ways to create coverage maps. Section 5 describes the method used when we have information about the positions of the antennas. If information of the antennas is not available, we can "walk" the robot and create a log with coverage readings at different positions. Section 6 describes three different ways to create coverage maps from these readings. Finally, Section 7 discusses the results from different methods, and concludes the paper.

2. Robot Control Architecture

The applications used involve a number of autonomous robots that can be monitored from the Web. Figure 1 shows one of these scenarios where the robots are connected to a central server using some kind of wireless communication. In addition, users connect to the central server using an application graphical user interface (GUI) to command different tasks to the robots and monitor them. Users might connect to the central server from different kinds of terminals such as computers or mobile phones. The kind of information transferred from the robots to the central server and from the central server to users depends on the application. For example, in a security system, information includes images from cameras or data about the on-board security sensors of

https://doi.org/10.1017/S0263574712000422 Published online by Cambridge University Press

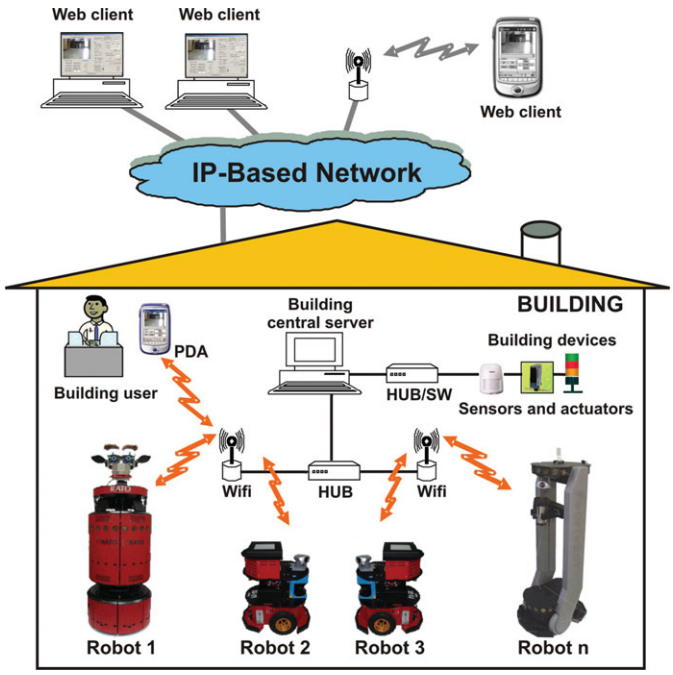


Fig. 1. (Colour online) Multi-robot application scenario where different robots are connected to a central server.

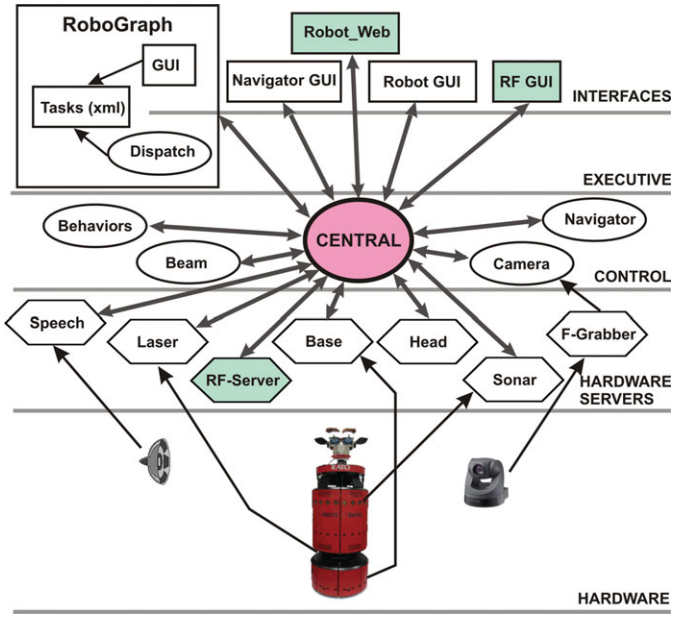


Fig. 2. (Colour online) ISANAV control architecture. Different modules are divided into several sets. The hardware server set reads sensor data and controls actuators. The control set provides several basic functions that can be used by other modules of control and executive sets.

the robots. It can also be useful to report the robot position and the planned tasks.

The robots are able to autonomously execute different tasks without the need to be continuously connected to the central server. However, a problem arises if one robot finishes a task at a point were it has no wireless coverage. Therefore, a mechanism to move the robot to a point where it can connect to the server is required.

The on-board control architecture is organized as shown in Fig. 2. Even though the different modules are organized in four sets, they can be mapped in the three-layer architecture popularized by Bonasso *et al.*[21] The hardware servers and the control set implement the functional layer, while the RoboGraph Dispatch implements the executive and planning layers. Finally, the architecture includes a set of processes to interact with the users and connect to the central server using a wireless communication system.

The navigation platform is based on CARMEN,[22] and some modules, such as localize, navigator, and base hardware servers, remain basically the same. Unlike CARMEN, motion control is divided into high-level (strategic) planning[23] and lower level (tactical) collision avoidance using the Beam method.[24]

2.1. Hardware server modules

The hardware server modules govern hardware interaction providing an abstract set of actuator and sensor interfaces and isolating the control methods from the hardware details. Most hardware devices are connected to a CAN bus using RoboCAN.[25] Some of these devices, such as laser and sonar, are used in navigation, while others are specific for the application, such as the robot head or the sound and speech system. Thanks to this layer, changes in hardware components can be made without changes in higher layer modules while keeping the same interface.

Special attention must be paid to the RF_Server module that obtains information from the wireless communication device about the signal strength from different APs. The Linux Wireless Extensions (WE) are the kernel API employed to develop the RF_Server module, and to guarantee the solution independence from the WLAND used. In particular, we use the IOCTL calls to monitor wireless variables (such as RSSI, ESSID, or AP MAC) and to modify connection parameters.

During application installation, the RF_Server module is used to create a log file while the robot moves through different areas of the building. This log file is used to create coverage maps. Even though it is not included in Fig. 2, there is a module named RF_Simulator that simulates the RF_Server behavior when a robot simulator is used. For simulation, RF_Simulator uses the log-normal shadowing propagation model to calculate the expected signal strength values.

2.2. Control modules

The control modules integrate sensor and motion information to provide improved sensor odometry, basic navigation capabilities, such as localization, path planning, or follow path, and basic application-specific functions, such as say a text or make a face expression.

2.3. Executive modules

All the modules in this layer belong to the RoboGraph application, which includes two modules: RoboGraph GUI is used only for application development, and RoboGraph dispatch is without graphical interface, which would be working when the robot is doing surveillance operations.

RoboGraph GUI is the programming integrated development environment (IDE) used by application developers to program robot tasks. These tasks coordinate the works of the rest of the modules, including some actions

https://doi.org/10.1017/S0263574712000422 Published online by Cambridge University Press

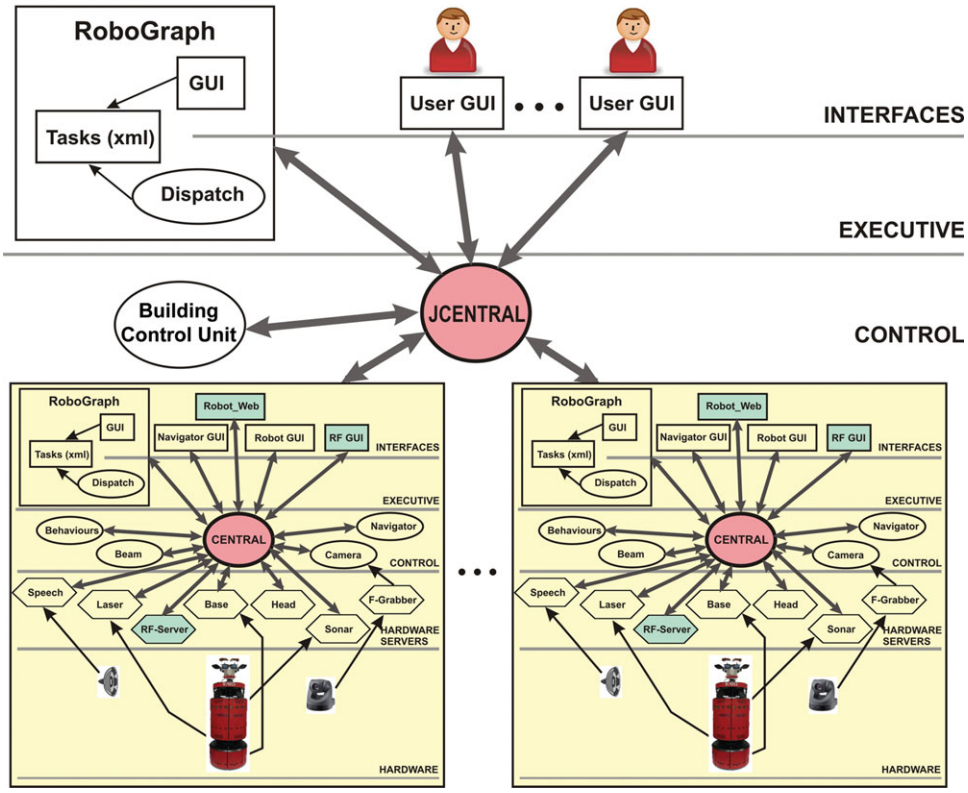


Fig. 3. (Colour online) Multi-robot control framework. The framework used to control a robot using IPC is extended to control and connect the robots, the building control unit that manages all building sensors and actuators, RoboGraph, which executes the tasks (Petri nets), and the user interface, which is an applet running on the users web browser.

carried out by the users on different interfaces. Tasks are defined as Petri Nets stored in XML files.

Robograph Dispatch loads tasks from XML files and executes them, sending commands to other modules and subscribing to their events.

2.4. Interface modules

A number of interface modules for application can be used by developers to debug and trace the control and hardware server modules. However, only one interface module on board allows the user to interact with the robot. There are two modules (Robot_Web and RF_GUI) on this level that are directly related to this research, as shown in Fig. 2. Robot_Web is a module that exchanges information between the robot and the building's central server station. RF_GUI is a module used in this research to create and display coverage maps, and create a new AP configuration and store it in an XML file. In addition, the GUI shows the information published by RF_Server or RF_Simulator.

3. Communication Framework

Each module in Fig. 2 is a Linux process that exchanges information with other modules using Inter Process Communication (IPC).[26] Developed at Carnegie Mellon's Robotics Institute, IPC provides a publication-subscription model for processes to exchange messages via a central server process. Each application registers with the central server, and specifies what types of messages are published and what types are listened. Any message that is sent to the central server is immediately copied to all other processes subscribed to it.

Our current research addresses architectures that extend the architecture described in Fig. 2 to control several mobile robots and connect them to a central server (Fig. 1). Figure 3 shows the proposed architecture for two robots connected to Wi-Fi, but it can be extended to multiple robot systems. Regarding the communication topology, the framework shown in Fig. 3 is very similar to the framework used inside each robot. Each IPC set of modules shown in Fig. 3 controls one robot and is running in the onboard computer. JCentral, the building interface, Robograph multi-robot, and video streaming are running in the central server computer that is connected to the Internet.

There are two information channels between the robots and the central server. One channel is established between the "Robot_Web" module (robot) and JCentral (central server), and the other channel (video channel) is established between the F-grabber module (robot) and the streamer (central server). Similar channels are also set between users and central server. A WLAN is used for communications between robots and the central server.

The JIPC framework is used with functionalities similar to IPC. However, it has been fully developed on Java and extended for multi-robot, multi-user applications.

JCentral needs to differentiate between temporal and final robot disconnections. For example, the Robot Web module can be stopped without being disconnected from JCentral. In this case, JCentral has to cancel the connection. In addition, a robot can loose temporal connection and JCentral should

https://doi.org/10.1017/S0263574712000422 Published online by Cambridge University Press

be able to resume communications afterwards. JCentral implements a timeout mechanism to disconnect modules when communication has been stopped for longer than a threshold time.

The building might also have a set of devices such as sensors and actuators that must be connected to the central server in some way. The building interface module shown in Fig. 3 handles all the information related to building devices.

4. Roaming Problem

While a mobile robot moves through the working area, it needs to change the AP association from one AP to another AP within the same WLAN. The robot connection system is responsible for deciding when it needs to scan, detect, evaluate, and roam to an alternative point. However, this problem is general to any mobile client that wants to be connected to the Wi-Fi while moving inside the building.

The original IEEE 802.11 standard does not provide optimal services for real-time applications. To tackle this problem, the 802.11e standard includes an amendment to provide needed Quality of Service (QoS). Besides, the 802.11i amendment has solved security issues by adding stronger encryption, authentication, and key management methods to guarantee data and system security. However, these amendments introduce expensive management frame exchanges between the mobile station and the AP. Some researchers[12] show that 802.11i may not be suitable by itself for voice transmissions during roaming. 802.11r standard is aimed to provide solutions to reduce the time taken for the transitioning process.

Our experimental results showed a relatively long delay and a low throughput when the robot moves from one AP to another. For most mobile robot projects, there is an interest in the use of the existing communication infrastructure available in buildings. At the moment, very few network devices implement 802.11r standard and therefore we do not want to restrict our system to work only with 802.11r-compliant devices. Our first approach to deal with this problem was to use antennas with higher gain. Surprisingly, rather than solving the problem, it made the problem worse. For this reason, a careful study of the roaming mechanism was needed.

In a managed Wi-Fi network, the client chooses the AP with which it wants to connect.[27] In general, the WLAN standards do not specify when a client should roam. Instead, the roaming algorithms are proprietary of each vendor and are not generally published. However, following are some common situations that typically cause roaming:

- Data retry count is exceeded: The IEEE 802.11 standard specifies the protocol to retry the transmission of data frames for which an acknowledgment was not received.
- RSSI threshold: The client device can decide to roam when RSSI drops below a threshold.
- Signal to noise ratio (SNR) threshold: The client device can decide to roam when SNR drops below a threshold.
- Load balancing schemes: The client device can decide to roam in order to balance client traffic across multiple APs.

WLAN Drivers are configured with a default set of RF roaming parameters, which are used to set RF thresholds adopted by the client to decide when to roam.

As stated earlier, roaming algorithms are proprietary of each vendor, and tests made with different devices can produce different results. For most devices, it seems that the WLAND roaming algorithm is mainly based on the RSSI threshold, named Roaming Threshold. When the robot moves from one AP to another and reaches the second AP vicinity, the RSSI from the first AP can be still good. However, the signal–noise ratio might be very poor. In this situation, the WLAND does not roam to the second AP, even though communication performs poorly and hardly any data are transmitted.

To solve this problem for a general client device, a new network module (RF-Server in Fig. 2) has been developed to monitor this situation and force the roaming. In order to detect these situations, it is necessary to scan for the available APs. However, special care must be taken because during a channel scan the client is unable to transmit or receive any data traffic. Clients may take two main approaches:

- Background scanning: Clients scan continuously for available channels. Therefore, when they need to roam, they already have the information about the available APs.
- On-roam scanning: Clients scan for available channels after a roam has been determined necessary.

To detect the situation described above, a background scanning approach is needed. A program that continuously searches for the best AP has been developed. When the difference between the present AP and the best AP reaches a threshold, it roams for a new AP.

5. Creating Coverage Maps From Propagation Models

Central server sends each task to the robot, which executes it autonomously. However, when one robot mission ends in an area without Wi-Fi coverage, the central server cannot connect to the robot and command new tasks. The solutions adopted for these problems are based on coverage maps. A coverage map was built for every AP. If there is no coverage in an area that should be covered by an AP, the robot moves to an area with good coverage for a different AP. In some applications, information about APs, such as AP position, can be easily gathered. In these cases, it is possible to obtain the expected RF power and deduce the RSSI measurements for our WLAND at different points of the environment using some propagation model. Nevertheless, these power signals are complex radio waves and many properties underlying electromagnetic wave propagation attributed to reflection, diffraction, and scattering should be considered.

A free space propagation model is used to predict received signal strength when the transmitter and the receiver have an unobstructed line-of-sight path between them. In this case, the received signal power is well known and decreases with the squared distance. However, in many practical indoor circumstances, there is no line-of-sight path between transmitter and receiver.

In real world situations, the channel changes according to the movement of the communicating entities, and many

https://doi.org/10.1017/S0263574712000422 Published online by Cambridge University Press

objects, still and moving, produce reflections, diffraction, and scattering. Attenuation of the signal is not only a factor of distance, but also of the obstacles between transmitter and receiver. Even people moving in a room will affect signal strength. Multi-path propagation may result in dead spots where no signal can be received. Therefore, the propagation of an RF wave in an indoor environment is difficult to predict.[28] Thus, RSSI measurements are sensitive to multi-path, fading, non-line of sight measurements, noise, refraction, and diffraction.

Many authors have conducted experimental and theoretical studies of indoor propagation models,[28–31] focusing on a particular characteristic such as temporal fading or inter-floor losses. Various models, such as COST 231 or Ericsson model 9999, among others, have been compared in ref. [32]. For example, the Two-Ray Model[29] relies on detail knowledge about the indoor environment. However, it is difficult to obtain such information for most mobile robot applications. We are more interested in empirical models that are useful in reducing computational complexity as well as increasing the accuracy of the predictions.

5.1. Log-normal shadowing propagation model

This section describes a method used to obtain an empirical map when the position of the antennas in our experiments is known.

This model defines the path loss for a receiver separated by a distance d from the transmitter as follows:

$$PL(d)_{dB} = PL(d_0)_{dB} + 10n \log\left(\frac{d}{d_0}\right) + X_\sigma(dB), \quad (1)$$

where $PL(d)$ is the path loss, $PL(d_0)$ is the path loss at reference distance d_0, $X_\sigma = 10\ \log(\mathrm{X})$ is the zero mean Gaussian-[1]distributed random variable (in dB) with standard deviation σ (also in dB), and n is the path loss exponent, which indicates the rate of increase in path loss with distance d. The value of n depends on the specific propagation environment, the type of construction material, the architecture, and the location within the building. For example, for buildings in areas where the transmitter is in the line-of-sight of the receiver, n ranges from 1.6 to 1.8, while in obstructed areas, n ranges from 4 to 6. However, in obstructed areas of factories, the path loss exponent, i.e. n, ranges from 2 to 3.

However, a preferred alternative is to calculate $PL(d_0)$, n, and standard deviation σ from the data gathered by the robot. First, the path loss (PL) at the reference distance (d_0) is measured. Next, the robot is moved through our environment while the RF-Server (Fig. 2) logs RSSI samples. Each sample contains the RSSI and the robot position in the map. The robot and the AP position is used to obtain the distance between the AP and the robot when the i measurement was taken (d_i). The AP position is a known data (stored with the map). The robot position is obtained by the localization module. That module integrates odometry and laser readings using a variation of the Monte-Carlo localization[33] algorithm to obtain the robot position in the map. A full description of the Robotics Integrated Development Environment (RIDE)[34] that we use to develop the mobile robot applications can be seen at http://webs.uvigo.es/vigobot/.

[1] X itself is not Gaussian, its log is Gaussian, so it is called log-normal, i.e., normal in the log scale.

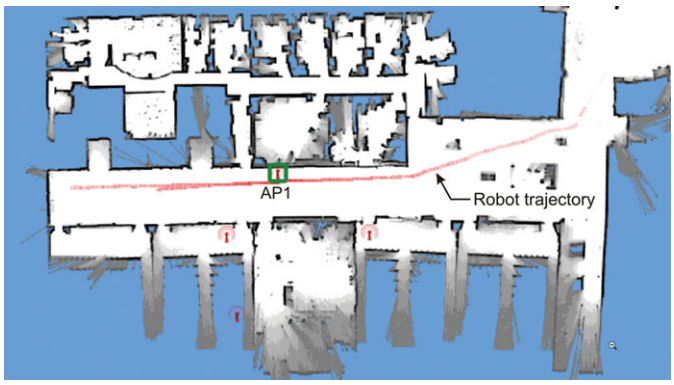


Fig. 4. (Colour online) Occupancy grid map of our department and path travelled by the robot while gathering signal strength samples.

The logs generated are used to obtain $PL_M(d_i)$ for different distances (d_i). Then n is determined as the value that minimizes the function:

$$F(n) = \sum_{i=1}^{k} [PL_M(d_i) - PL_{\mathrm{LN}}(d_i)]^2, \quad (2)$$

where $PL_M(d_i)$ is the path loss measured by the receiver on the robot, $PL_{LN}(d_i)$ is the path loss obtained according to Eq. (1), and k is the number of samples. Using Eq. (1) in Eq. (2), we obtain

$$F(n) = \sum_{i=1}^{k} \left[PL_M(d_i) - PL(d_0)_{\mathrm{dB}} + 10n \log\left(\frac{d}{d_0}\right) + \mathrm{X}_\sigma\right]^2, \quad (3)$$

$F(n)$ has a minimum for n:

$$n = \frac{\sum_{i=1}^{k} \left[\log\left(\frac{d_i}{d_0}\right)(PL_M(d_i) - PL(d_0))\right]}{10 \times \sum_{i=1}^{k} \left[\log\left(\frac{d_i}{d_0}\right)\right]^2}. \quad (4)$$

Figure 4 shows the experimental occupancy map of our department computed using this approach. White areas are places where the robots can move. In the area depicted in Fig. 4, there are three APs, but for simplicity we consider only the signal coming from the AP labeled as "AP1" in Fig. 4. The line on the main corridor shows the trajectory followed by the robot and the samples gathered by the robot. Darker points represent better coverage than lighter points.

Figure 5 shows the signal level from the robot while travelling a section of the path described in Fig. 4, and the logarithmic approach (Eq. (1)) without the random variable. The parameter n for the logarithmic approach was obtained using a similar dataset from the same robot. Note that the logarithmic approach is good on average, even though the data set has a quite high standard deviation. Such a deviation is the reason to add the random variable in model defined by Eq. (1).

https://doi.org/10.1017/S0263574712000422 Published online by Cambridge University Press

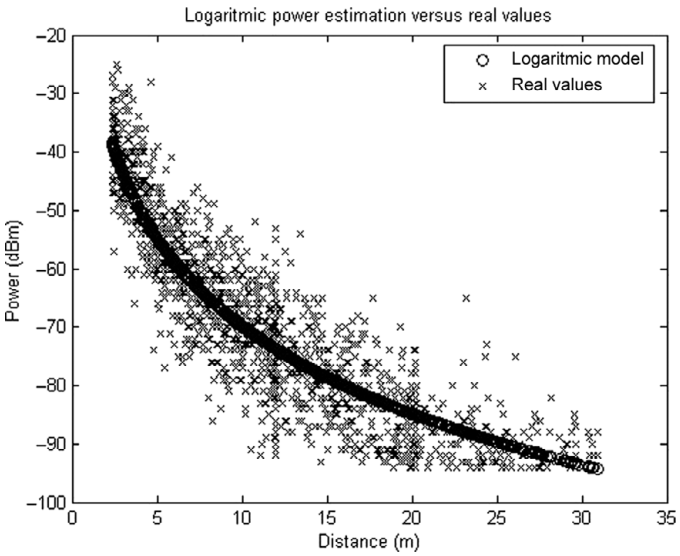


Fig. 5. Samples obtained with the robot for different distances against values obtained with the logarithmic model (Eq. (1) without the random variable).

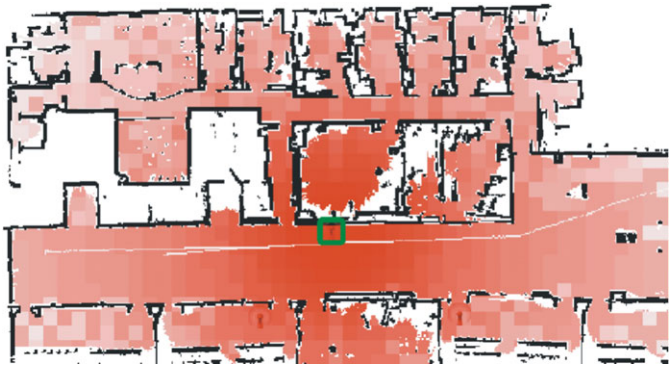


Fig. 6. (Colour online) Coverage map created using the log-normal shadowing propagation model.

After n has been obtained, standard deviation σ can be calculated from Eq. (5):

$$\sigma^2 = \frac{1}{k}\sum_{i=1}^{k}[PL_M(d_i) - PL_{LN}(d_i)]^2. \qquad (5)$$

Finally, after having calculated all the log-normal path loss model parameters, path loss is computed for all cells of the environment using Eq. (1). An example of the coverage map created using this method is shown in Fig. 6. Since we are interested in average values, Fig. 6 has been created by removing the random variable from Eq. (1).

Instead of using only one value of n to obtain the coverage in each point of the map, we have considered two different values. The first one is used for the points in the map that are in line-of-sight (situation 1) with the AP, and the second one for the points in the map where the AP is obstructed by some obstacle such as a wall (situation 2). Both values are obtained according to Eq. (5) by considering two experimental data sets. The first set (Fig. 5) was obtained in map positions that have line-of-sight with the AP, and the second set was gathered where the AP is obstructed.



Fig. 7. (Colour online) Coverage map created using the expansion algorithm.

Two different values of n have been used to create the map shown in Fig. 6. The first value is used for the AP line-of-sight areas, and the second value is used for the rest of the areas.

6. Creating Coverage Maps From Samples

Three different solutions are presented in this section, which use the data stored in the log file by RF_Server module (Fig. 2). Therefore, a preliminary step to apply these methods is to command the robot to navigate through different areas of the workspace while RF_Server stores measurements in the log file. As the robot moves, the RF_Server module continuously gathers the wireless signal strength from the wireless communication device and stores it together with the robot position.

Figure 4 shows the robot trajectory while collecting samples used for testing different approaches. The measurements used are the same as those used in the previous section.

6.1. The expansion algorithm

This algorithm obtains the coverage of a position from the values of its neighbours. First, a set of cells is initialized with the readings from the log file. Then Algorithm 1 is applied to obtain the values in all the cells of the environment. The aim is to compute coverage in unknown cells as the weighted sum of coverage in the neighboring known cells.

Algorithm 1. Expansion algorithm

```
for each read_cell from the redings set
    for each cell nearby the read_cell that has no value yet
        obtain value as the average of the valuated contiguous
cells
        assign value to the cell
        add cell to the readings set
    end for each cell
    remove read_cell from the readings set
end for each read
```

Figure 7 shows the map created by using Algorithm 1. Note that this is a very unfavorable case for this algorithm, since there are only coverage values in the main corridor.

6.2. The voting algorithm

As in the method presented in the previous section, the voting algorithm uses only the log information without AP positions. In this case, the expected RSSI values for all the

https://doi.org/10.1017/S0263574712000422 Published online by Cambridge University Press



Fig. 8. (Colour online) Coverage map created using the voting algorithm.

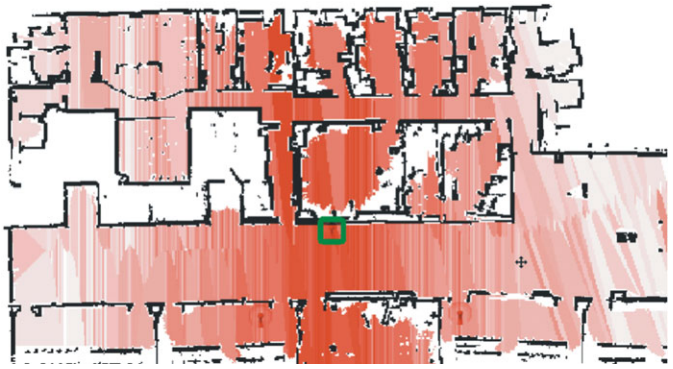


Fig. 9. (Colour online) Coverage map created using the closest reading mechanism.

cells of the environment are obtained using the RSSI values of those points stored in the log file. The voting algorithm obtains the value in a cell as the sum of point values where there are measures, weighted inversely proportional to the distance. It can be seen as if every point where there is a measured value voted its own value, and the weight of its vote was inversely proportional to the distance. Therefore, closer points will have more weight than the points that are far away. Using this algorithm, the value of a cell m(x,y) is computed according to the following expression:

$$P_m = \eta \sum_{i=1}^{k} \frac{P_i}{d_i}, \tag{6}$$

where P_i for $i = 1, 2, 3 \ldots k$ are measures obtained from the log files, and d_i is the distance between a specific cell and the point where measure i was taken. Finally, η is a normalizing factor:

$$\eta = \frac{\prod_{i=1}^{k} d_i}{\sum_{i=1}^{K} \prod_{r \neq i} d_r}. \tag{7}$$

Figure 8 shows the coverage map obtained using this algorithm with the same data set that was used in the previous section.

6.3. The closest reading algorithm

This method obtains the expected value for every cell in the workspace as the value of the closest point in the log file.

To calculate the value in cell m(x,y), the closest point j in the log file, i.e. the point i that minimizes the Euclidean distance, must be found first:

$$j = \arg_i \min \left[\sqrt{(x_m - x_i)^2 + (y_m - y_i)^2} \right]. \tag{8}$$

This simple method obtains good results when the measures are scattered over the entire environment. However, it does not work very well when samples cover the area of interest only partially, as in the case study. Figure 9 shows the computed coverage map.

7. Experimental Results

This section includes results of some experiments performed with real robots in the main building of our School using the shared wireless network. These experiments allowed us to evaluate the different solutions adopted, and to learn from the different solutions developed for both wireless communication problems, i.e. roaming and coverage problems.

7.1. The roaming problem

To illustrate the advantage of using the roaming monitor (RF_Server module), the following three throughput-based experiments were performed:

- Experiment 1: Standard wireless connection without running the roaming monitor.
- Experiment 2: Wireless connection with one Wi-Fi card that performs data transmission and the roaming monitor.
- Experiment 3: Wireless connection with two Wi-Fi cards, one performs data transmission and the other performs the roaming monitor.

These three experiments were repeated several times. Similar but not identical results were obtained for the different executions of the same experiment. The differences between the executions within the same experiment are due to the random nature of wireless communications, mainly caused by environment changes and the traffic of wireless links. For this reason, without loss of generality, only one result for every experiment is presented.

For all experiments, the robot moved between points A, B, and C of Fig. 10, following the sequence A->B->C, stayed in C for a minute, and finally went back to A, following the sequence C->B->A. In order to evaluate the throughput, extra traffic was generated, transferring a big amount of data between the robot and the central station, located outside the wireless network. Note that the traffic generated in real mobile robot applications developed in our laboratory usually produces much less data traffic.

Figures 11 and 12 show the throughput obtained from experiments 1 and 2 respectively. The throughput shown in Fig. 11 was obtained when the robot was not running the roaming monitor. This monitor was activated in the second run for experiment 2, and produced the throughput shown in Fig. 12.

https://doi.org/10.1017/S0263574712000422 Published online by Cambridge University Press



Fig. 10. (Colour online) Occupation grid map of our department and path travelled by the robot while taking signal strength samples.

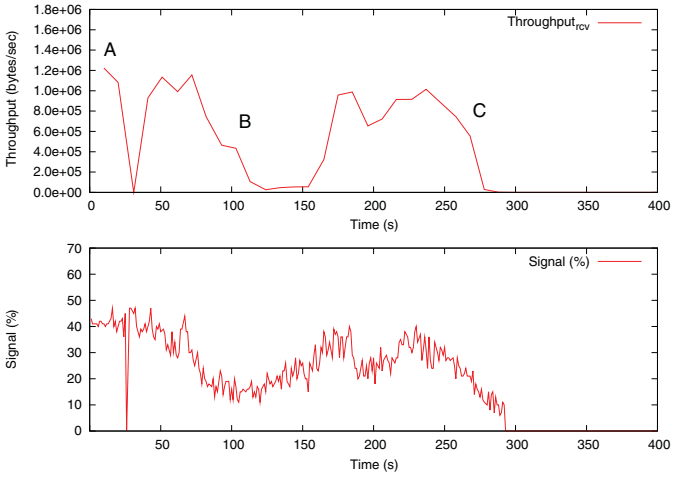


Fig. 11. (Colour online) Throughput and signal power for the test without roaming monitor.

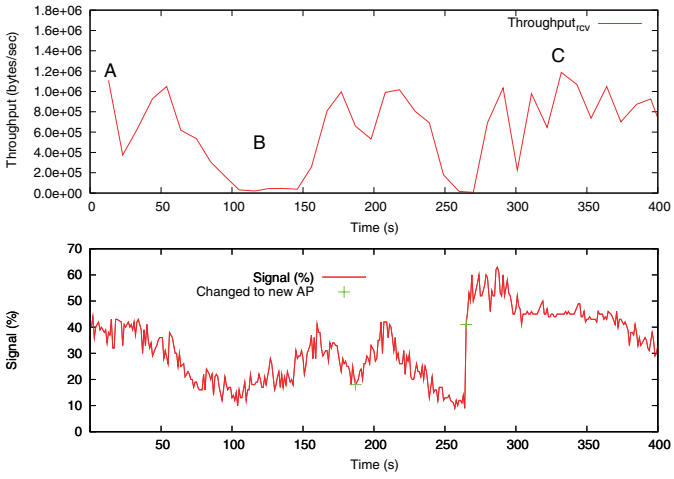


Fig. 12. (Colour online) Throughput and signal power for the test with roaming monitor.

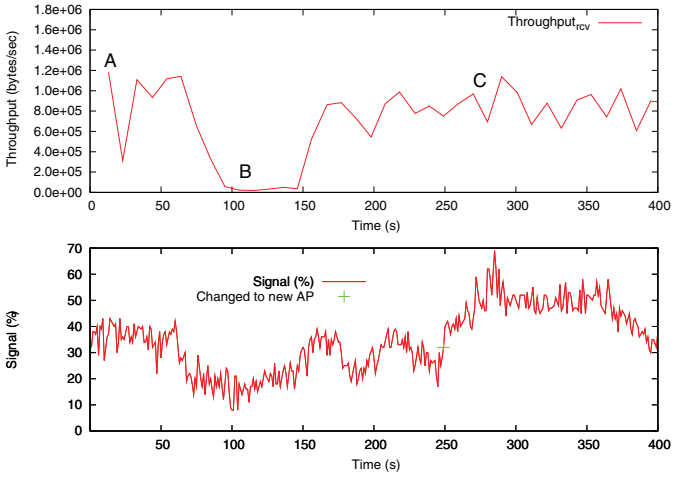


Fig. 13. (Colour online) Throughput and signal power for the test with roaming monitor using two Wi-Fi cards.

In the first section of the trajectory, while the robot moves toward B, the throughput decreases in both cases. Point B is far from all the APs. Therefore, both executions have similar results because there is no alternative AP to switch.

However, when the robot moves toward C, the roaming monitor finds better APs to switch, thus obtaining a better throughput. The worst roaming conditions occur in C, because the robot was not able to get the new command without the roaming monitor and it was necessary to use the on-board GUI to command the last section of the trajectory. This kind of situation was the main motivation to develop the roaming monitor. Most of our mobile robot applications do not need much bandwidth, but they require that robots be connected to the best AP anytime. For the experiment in Fig. 11, the robot lost connection and was not able to recover it again. Even though this is not the most common behavior, it has been observed in a few runs of the number of tests carried out for this research.

During a channel scan, the robot is unable to transmit or receive data. Nevertheless, most of the applications developed allow delays of up to 1 s. The nature of the data transmitted through JCentral allows for such delays. In video streaming, the buffers make these delays transparent to users. However, there is a constant delay (2 to 3 s) between the images on the camera and the GUI.

For applications where the throughput must be maintained, we propose to use two Wi-Fi cards (experiment 3). The first card is used for the scanning process, and the second card is used for data transmission. Therefore, the scanning process is independent of whether data transmission has obtained a better throughput. The results of experiment 3 with two Wi-Fi cards solution are shown in Fig. 13.

Again, the main difference between Figs. 11 and 13 is the throughput value when the robot gets closer to point C. In Fig. 11, without the roaming monitor, the connection with the robot is lost, whereas in Fig. 13 the roaming monitor switches to AP4, keeping the robot connected. In Fig. 12 and 13, maximum throughput is very similar and the difference between using one or two Wi-Fi cards cannot be observed. The throughput depicted in Figs. 11–13 is the average over a window of few seconds. However, if throughput is represented in the interval of each second, the difference becomes evident (Figs. 14 and 15).

https://doi.org/10.1017/S0263574712000422 Published online by Cambridge University Press

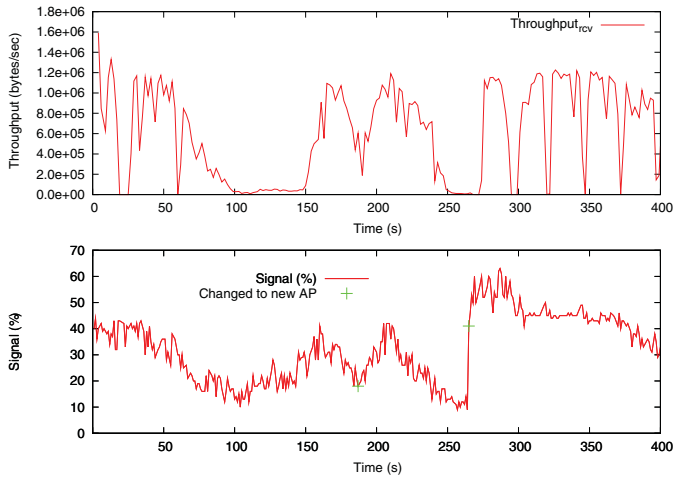


Fig. 14. (Colour online) Instant throughput and signal power for the test with roaming monitor using one Wi-Fi card.

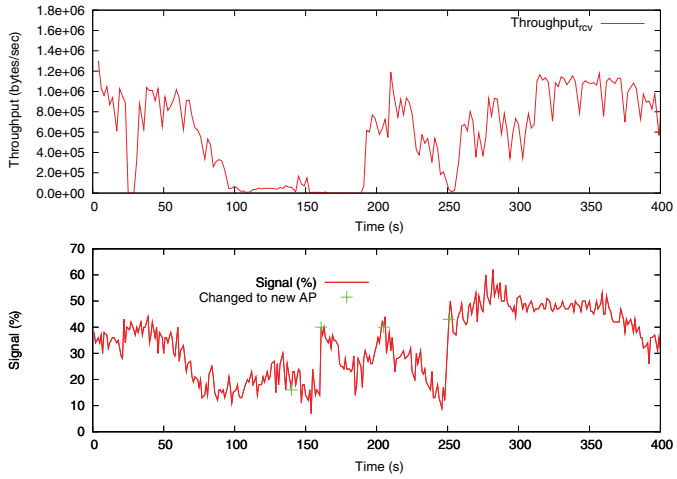


Fig. 15. (Colour online) Instant throughput and signal power for the test with roaming monitor using two Wi-Fi cards.

As shown in Fig. 14, during a channel scan, the robot is unable to transmit or receive data. Using two Wi-Fi cards (Fig. 15) helps to overcome this problem.

7.2. Coverage problems

A coverage map was built for every AP. If there is no coverage in an area that should be covered by an AP, the robot moves to an area with good coverage for a different AP. However, in order to show the differences of the algorithms presented here, it is more convenient to show the coverage map for only one AP. In the previous sections, we have shown coverage maps for AP1 when data are only available in a small area (Fig. 4).

In the following figures we show the maps generated by different algorithms when data are available over different areas. The path followed by the robot while gathering the data is shown in Fig. 16.

Figures 17 to 19 show the coverage maps generated by the three proposed methods for AP1 from data gathered by the robot while following the path shown in Fig. 16.

According to the coverage map shown in Fig. 17 and generated using the expansion algorithm, areas without



Fig. 16. (Colour online) Occupancy grid map of our department and path travelled by the robot while taking signal strength samples.

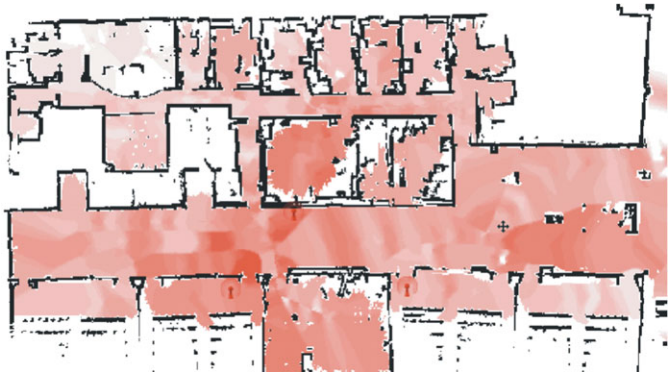


Fig. 17. (Colour online) Coverage map using the expansion algorithm and the log data shown in Fig. 16.

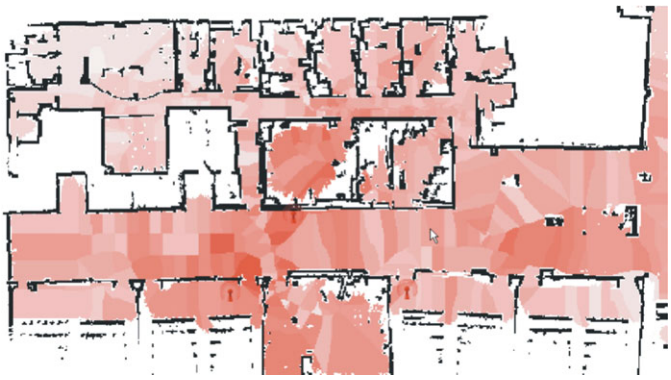


Fig. 18. (Colour online) Coverage map using the closest algorithm and the log data shown in Fig. 16.

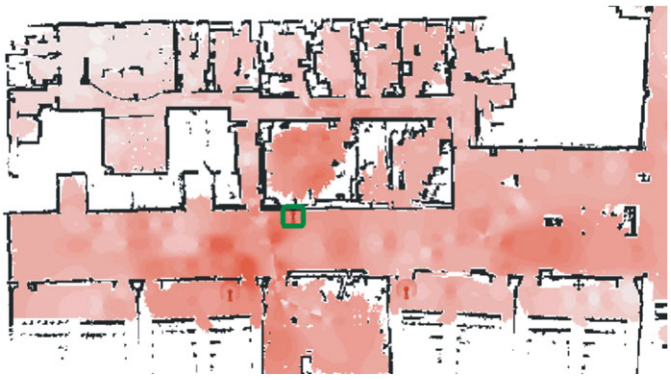


Fig. 19. (Colour online) Coverage map using the voting algorithm and the log data shown in Fig. 16.

https://doi.org/10.1017/S0263574712000422 Published online by Cambridge University Press

coverage correspond to upper-left corner offices because the signal has to traverse several walls. Similar results have been obtained using the closest algorithm (Fig. 18) and the voting algorithm (Fig. 19).

As expected, the decision on use of algorithm will depend on the number of samples provided. Thus, if there are only a few samples and the AP position is known, the log-normal shadowing propagation model is a good choice. On the other hand, if there are a large number of samples for most map areas, one of the other two methods should be used. In addition, some combined method can be used when there are few data available.

The main reason for increased reliability of gathered data is that it is almost impossible for propagation models to take into account all the signal propagation impairments. Actually, even the presence of a vending machine can modify the coverage of its surroundings.

Regarding the difference between the sample-based methods, an increase in the data available will cause and increase the similarity of maps. The closest algorithm uses rough data gathered from the readings. Because of the deviation of the samples (Fig. 5), high differences are observed between neighboring areas. For this reason, some filter should be applied. Nevertheless, the voting algorithm includes this filtering, and produces the map shown in Fig. 19.

A detailed look at all the maps created from samples reveals the presence of some low-coverage spots, even in areas that should have high coverage, such as the corridor where AP is located. There are several possible causes for these spots. As shown in Fig. 5, the first cause for low-coverage spots is sample deviation, even though measurements were taken in an area where transmitter and receiver have an unobstructed line-of-sight path between them. The second cause is the location of a wireless device in the robot and possible interferences with other devices on board. Moreover, it is necessary to consider the potential wireless interference sources that can affect wireless transmission, such as fading, multi-path caused by reflections, diffraction or scattering, and noise. Some blind spots caused by interference with different electrical devices, such as the vending machines, were also identified.

8. Conclusions

We have identified and proposed different solutions for the communication problems found while developing mobile robot applications that rely on WLANs.

Using commercial APs and clients may bring about situations with a very low connection throughput while the client (robot) moves from the vicinity of one AP to the vicinity of another AP. The source of the problem is that the WLAND client on the robot is still connected to the first AP and does not roam to the closest AP. The problem becomes worse as we get higher gain antennas because the WLAND does not decide to scan for alternative APs. A simple and WLAND independent solution has been proposed to this roaming problem. The solution consists in developing a monitor that scans for alternative APs and forces the client to roam to the closest AP when necessary. There is an impact on client traffic because during channel scanning, the client is unable to transmit or receive client data traffic. Nevertheless, to minimize this impact, scanning can be done in a very low frequency, which is enough for this kind of applications because the robots move at a rather low speed. Maximum robot speed is 0.5 m s^{-1}.

Regarding the coverage problem, different methods to create Wi-Fi coverage maps that can be used in applications based on mobile robots that need to be connected using the wireless communication system have been presented. Two different approaches can be used. The first approach is based on determining the location of APs and obtaining the expected signal strength values in different areas of the environment, while the second set of methods obtains the maps from samples taken by the robots while navigating through the environment.

In current applications, maps are used only when the robot ends a task at a point without coverage. However, maps can also be used in path-planning methods to avoid (if possible) non-coverage spots. As a future work, the maps can be used to improve Wi-Fi-based localization methods as suggested by other researchers.[35]

Acknowledgments

This work was supported in part by the Spanish Comisión Interministerial de Ciencia y Tecnología; project DPI2008-06738-C02-02.

We would like to thank all the people that have had an influence on this work. In particular, thanks to Diego Pérez, Manuel Álvarez, and Verónica Otero. The authors would also like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper.

References

1. P. X. Liu, M. Q-H. Meng and S. X. Yang, "Data communications for Internet robots," *Auton. Robots* **15**(3), 213–223 (2003).
2. R. Simmons, J. L. Fernández, R. Goodwin, S. Koenig and J. O'Sullivan, "Lessons learned from Xavier," *Robot. Autom. Mag.* **7**, 33–39 (2000).
3. D. Wu, H. Ogai, Y. Yeh, K. Hirai, T. Abe and G. Sato, "Pipe inspection robot using a wireless communication system," *Artif. Life Robot.* **14**, 154–159 (2009).
4. Y.-G. Kim, H.-K. Kim, S.-H. Yoon, S.-G. Lee and K.-D. Lee, "Home Security Robot Based on Sensor Network," **In**: *Proceedings of the. SICE-ICASE 2006 International Joint Conference*, Seoul, Korea (2006) pp. 5977–5982.
5. W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artif. Intell.* **114**, 3–55 (1999).
6. Z. Wang, L. Liu and M. Zhou, "Protocols and applications of ad-hoc robot wireless communication networks: An overview," *Int. J. Intell. Control Syst.* **10**, 296–303 (2005).
7. Z. Wang, M. Zhou and N. Ansari, "Ad-hoc Robot Wireless Communication," **In**: *Proceedings of the. IEEE International Conference on Systems, Man and Cybernetics*, Washington, D.C., USA (2003) pp. 4045–4050.
8. K. Kaemarungsi and P. Krishnamurthy, "Modeling of Indoor Positioning Systems Based on Location Fingerprinting," **In**: *Proceedings of the IEEE Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, Hong Kong, China (2004) pp. 1012–1022.

https://doi.org/10.1017/S0263574712000422 Published online by Cambridge University Press

9. IEEE Computer Society, *ISO/IEC 8802-11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications* (IEEE, New York, 1999).
10. J. L. Fernández, D. P. Losada and R. Sanz, "Enhancing Building Security Systems with Autonomous Bobots," **In**: *Proceedings of the 2008 IEEE International Conference on Technologies for Practical Robot Applications*, Boston, MA, USA (2008) pp. 19–25.
11. J. L. Fernández, R. Sanz, E. Paz and C. Alonso, "Using hierarchical binary Petri nets to build robust mobile robot applications: RoboGraph," **In**: *Proceedings of the IEEE International Conference on Robotics and Automations (ICRA 2008)*, Pasadena, CA, USA (2008) pp. 1372–1377.
12. C. Yap, E. Qi, K. Sood, S. Bangolae and C. Bell, "Issues with Real-Time Streaming Applications Roaming in QoS-Based Secure IEEE 802.11 WLANs," **In**: *Proceedings of the 2nd International Conference on Mobile Technology, Applications and Systems*, Boston, MA, USA (2005) pp. 1–7.
13. K. R. Renjish Kumar, "Service Roaming over Mobile Networks: A Reality Check," **In**: *Proceedings of the IEEE International Conference on Communications*, Glasgow, Scotland (2007) pp. 2058–2063.
14. T. K. Sarkar, Z. Ji, K. Kim, A. Medouri and M. Salazar-Palma, "A survey of various propagation models for mobile communication," *IEEE Antennas Propag. Mag.* **45**, 51–82 (2003).
15. M. F. Iskander and Z. Yun, "Propagation prediction models for wireless communication systems," *IEEE Trans. Microw. Theory Tech.* **50**, 662–673 (2002).
16. A. Hills, J. Schlegel and B. Jenkins, "Estimating signal strengths in the design of an indoor wireless network," *IEEE Trans. Wirel. Comm.* **3**, 17–19 (2004).
17. C. Cheon, G. Liang and H. L. Bertoni, "Simulating radio channel statistics for different building environments," *IEEE J. Sel. Areas Commun.* **19**, 2191–2200 (2001).
18. A. Bose and C. H. Foh, "Practical Path Loss Model For indoor WiFi Positioning Enhancement," **In**: *Proceedings of the 6th International Conference on Information, Communications & Signal Processing*, Singapore, Singapore (2007), pp. 1–5.
19. V. Matellán, J. M. Cañas and O. Serrano, "WiFi localization methods for autonomous robots," *Robotica* **24**, 455–461 (2006).
20. A. Kotanen, M. Hännikäinen, H. Leppäkoski and T. D. Hämäläinen, "Positioning with IEEE 802.11b Wireless LAN," **In**: *Proceedings of the 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC2003)*, Beijing, China (2003) pp. 2218–2222.
21. R. P. Bonasso, D. Kortenkamp, D. P. Miller and M. Slack, "Experiences with an architecture for intelligent, reactive agents," *J. Exp. Theor. Artif. Intell.* **9**, 237–256 (1995).
22. M. Montemerlo, N. Roy and S. Thrun, "Perspectives on Standardization in Mobile Robot Programming: The Carnegie Mellon Navigation (CARMEN) Toolkit," **In**: *Proceedings of the. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, Las Vegas, Nevada, USA (2003) pp. 2436–2441.
23. A. R. Diéguez, R. Sanz and J. L. Fernández, "A global motion planner that learns from experience for autonomous mobile robots," *Robot. Comput. Integr. Manuf.* **23**, 544–552 (2007).
24. J. L. Fernández, R. Sanz, J. A. Benayas and A. R. Diéguez, "Improving collision avoidance for mobile robots in partially known environments: the beam curvature method," *Robot. Auton. Syst.* **46**, 205–219 (2004).
25. J. L. Fernández, M. J. Souto, D. P. Losada, R. Sanz and E. Paz, "Communication Framework or Sensor-Actuator Data in Mobile Robots," **In**: *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE 2007)*, Vigo, Spain (2007) pp. 1502–1507.
26. R. Simmons, "The interprocess communications system (IPC)," available at: http://www.cs.cmu.edu/~/afs/cs/project/TCA/www/ipc/ipc.html (1998) online Accessed July 25, 2012.
27. IEEE Computer Society, LAN/MAN Standards Committee, *IEEE Std 802.11-2007, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications* (IEEE, New York, Jun. 2007).
28. H.-J. Zepernick and T. A. Wysocki, "Multipath Channel Parameters for the Indoor Radio at 2.4 GHz ISM band," **In**: *Proceedings of the IEEE 49th Vehicular Technology Conference*, Houston, Texas, USA (1999) pp. 190–193.
29. F. A. Agelet, A. Formella, J. M. H. Rábanos, F. I. Vicente and F. P. Fontán, "Efficient ray-tracing acceleration techniques for radio propagation modeling," *IEEE Trans. Veh. Technol.* **49**, 2089–2104 (2000).
30. M. F. Iskander, Z. Yun and Z. Zhang, "Outdoor/Indoor Propagation Modeling for Wireless Communications Systems," **In**: *Proceedings of the IEEE Antennas and Propagation Society International Symposium*, Boston, MA, USA (2001) pp. 150–153.
31. J. H. Tarng and T. R. Liu, "Effective models in evaluating radio coverage on single floors of multifloor building," *IEEE Trans. Veh. Technol.* **48**, 782–789 (1999).
32. A. Neskovic, N. Neskovic and G. Paunovic, "Modern approaches in modeling of mobile radio systems propagation environment," *IEEE Commun. Surv.* **3**, 2–12 (2000).
33. S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artif. Intell.* **101**, 99–141, 2000.
34. J. López, D. Perez and E. Zalama, "A framework for building mobile single and multi-robot applications," *Robot. Auton. Syst.* **59**, 151–162, (2011) available at: http://webs.uvigo.es/vigobot/.
35. A. Bose and C. H. Foh, "Practical Path Loss Model for Indoor WiFi Positioning Enhancement," **In**: *Proceedings of the 6th International Conference on Information, Communications & Signal Processing*, Singapore, Singapore (2007) pp. 1–5.

https://doi.org/10.1017/S0263574712000422 Published online by Cambridge University Press