Wireless Network Issues for a Roaming Robot

Riccardo Cassinis, Fabio Tampalini and Paolo Bartolini
Department of Electronics for Automation
University of Brescia
IT - Brescia
{riccardo.cassinis,fabio.tampalini}@ing.unibs.it, ilbarto@virgilio.it

***Abstract*— This paper describes a work carried out at the Advanced Robotics Laboratory of the University of Brescia, which consisted in the project and the realization of a package to allow a mobile autonomous robot to manage the access to wireless TCP/IP networks by itself. With this software, the robot is able to look for an available communication channel and to establish a new connection to it, once found. Moreover, if the robot detects more than one network, it has the essential knowledge to choose the best one and to start communicating using that channel.**

The main goal of this project is to make the robot able to employ, always and everywhere, the best available way of communicating, by supplying it with the capability of deciding if the current connection is the best it could actually exploit, or if it would be useful to switch to another channel. This task is completed without any awareness of the context in which the robot is; this means that the robot can immediately start working in every new place it is located, with no need for reconfiguration.

***Index Terms*— Robot, roaming, wireless, networks.**

I. Introduction

The ongoing wireless protocols standardization, which offers the possibility of data transmission with very high bit-rate, the convenience of having a secure and inexpensive connection without the need of cabling, combined with the recent successful deployment of wireless interfaces in numerous hot-spots justify the fact that wireless technology will play a key role in data transmission. Wireless networks represent the most attractive solution to many communication problems, from a simple *ad-hoc* link between two devices transferring a file, to a third generation (3G) cellular phone that is requesting data from the Internet.

Just for this reason, during the last few years, we have been seeing a lot of new wireless standards arise, each one dedicated to a specific application or need. For instance, IEEE 802.11 Wireless Local Area Networks (WLANs) are now widely available in most organizations, while Global System for Mobile Communications (GSM) has recently upgraded to provide packet access via a Generic Packet Radio Service (GPRS). An example of the involved standards can be summarized in Figure 1.

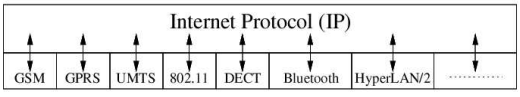

Fig. 1. Most widespread wireless protocols standards.

As we can see, Internet Protocol (IP) represents the level–3 layer for all these implementations, though we remember it was originally designed to provide a robust, best effort datagram service for Transport Control Protocol (TCP) layer and, until few years ago, it has been used mostly for asynchronous communication, such as electronic mail exchange, file transfer or web browsing. The good knowledge of the IP, added to the new studies on mechanisms to obtain the Mobile IP (MIP), makes it possible to implement TCP/IP based networks for modern purposes of high bit-rates data transfer, security and mobility.

In particular, this work focuses on the latter aspect mentioned above, i.e. the mobility of a device, and it is especially addressed to the needs of mobile robotics.

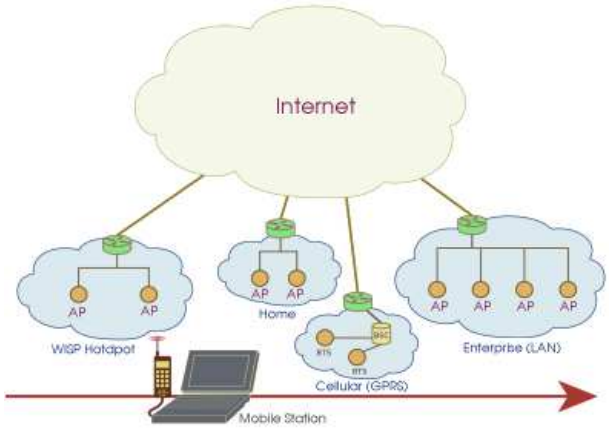

Fig. 2. A typical scenario. As the robot moves in the environment, different choices become available.

Let us describe the typical scenario we are thinking about (see for example Figure 2): a mobile robot has to carry out an assigned work in a specific area, which can be either a department of an organization or a town neighborhood, anyway an area that is covered by a wireless network, using one of the TCP/IP standards (typically, they should be 802.11 in a business environment or GPRS outdoor). The topology of the network implies that the robot has to log on a unique access point that routes its messages onto the Internet. Of course, a robot, equipped with a dedicated device, such as a wireless LAN or GPRS PC-card inserted into a computer mounted on it, is able to access (at least) one of the available networks: this feature is directly implemented by the standard protocol and the device driver. In this way the static situation is solved and also a little

dynamics is contemplated: for now, the standards support mobility between access points that are connected to the same backbone, which constitute a single IP subnet – we refer to this capability as *link level mobility*. This option is contemplated as *roaming* between access points with the same identifier and security protocols (such as ESSID and WEP for WLANs) and allows the devices to move inside the network maintaining their IP address.

What actually lacks in the standards is the possibility for a mobile station to roam from one network to another, which is essential for our purpose, because our robots must move to do its job and probably the coverage area of the network it is actually logged on is not wide enough to let it be connected everywhere it needs.

In conclusion, considering the present heterogeneous wireless infrastructure and supposing a future still more varied situation, we want to give our robot the capability, not yet implemented in standards, to roam over different networks, using adequate protocols and drivers, and let it choose always the best available way of communication, depending on the environment and the user's preferences. Networks' switchings should be as fast as possible and arguably the robot should be able to employ all the available wireless communication channels until their Quality of Service (QoS) becomes unsatisfying, in order to reduce the frequency of the switchings and minimize the handoff time.

In section II, we will examine the studies that have been completed during the last years and the state of the art of wireless communications; then, in section III, we will describe in details the solution we propose for a good communications' management for mobile robots; finally, in section IV, we will present the results of our work with its capabilities and we will explain some perspectives for our project.

II. The state of the art

Nowadays, wireless communication is a subject worldwide largely discussed, therefore mant projects have been completed and many more are in progress: in this section, we are going to point out the main trend of this activity.

A part of the current studies deals with IP mobility, i.e. the ability of changing IP address while moving between different networks. Some of these studies are centered on IP-layer mobility, which means a transparent routing of IP datagrams even as the mobile station moves and changes its point of attachment to the Internet, obtained by a tunnelling of datagrams at the IP-level; this approach is the most intuitive, but implies many complications because the responsibility of the flow-control and the session recovery is demanded to the transport-layer (TCP) [Dundar and Puri, 2002].

An alternative solution is offered by the session-level mobility: this means that all the roaming management is demanded to a high level of the ISO-OSI stack, so moving from a network to another implies a new connection with the latter, gained by almost the same operation as the first access to a network; this approach can be performed thanks to the session-resume functionality of the transport-layer, that assures that if a connection is lost, a new one can be quickly set up, without further complex computations [Sällström, 2003].

Other studies regard the integration of different wireless protocols, especially concerning WLAN and GPRS: the most important phone companies sponsor researches on obtaining access of WLANs by their cellular phones, to allow their customers very high data-rate transfers in places covered by WLANs, while providing usual services outdoor in the cities, where the mobile station can communicate by GPRS. These studies are mostly centered on a roaming transparent to the user and also systems for Authentication, Authorisation and Accounting (AAA), which include new billing mechanisms [Salkintzis et al., 2002].

In the meanwhile, the project of standardization is proceeding: as for WLANs, the Institute of Electric and Electronic Engineers is hardly working on the 802.11 standard (the system that is generically called "Wi-Fi"): after the consolidation of 802.11a, 802.11b and 802.11g for the Medium Access Control (MAC) and Physical Layer (PHY) specifications, the most relevant work related to our purpose is the Inter Access Point Protocol (IAPP), defined in 802.11f standard, that improves the handover mechanism between access points; that is still insufficient for our intention because it is limited to access points of the same network [IEEE, 2003].

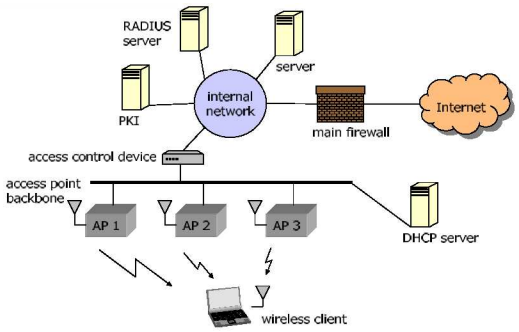

Fig. 3. Architecture of a typical WLAN: according to IEEE 802.11f standard, roaming is supported between APs with the same SSID.

Current work also involves some real implementations; in this project, we focused on software applications dedicated to computers using Linux operating system, developed to manage WLANs connections. About this subject, we found the Wireless Extensions for Linux, an API written by Jean Tourrilhes to manipulate wireless networking devices in a standard and uniform way [Tourrilhes, 1997].

Based on these extensions, several applications (like "apradar", "waproamd", "kwifimanager") have been developed, that can help the user switching from access points, but with some limitations: some of them are too pervasive and while they monitor the networks correctly, they effectively do not let the user exploit any connection; furthermore, they are usually dedicated to a specific transmission protocol and do not allow channel switching, for instance,

from a WLAN to a UMTS public network. Moreover, all the applications mentioned above are designed to present some statistics to the user and need somebody that reads those results and decides what to do: therefore, they are not suitable for our purposes of integration with robotics.

III. The proposed solution

The solution we propose is a software that acts like a daemon, running under the other applications working on the computer that controls the robot.

A. Basic idea and configuration

What this software has to do is to monitor the quality of the current connection and, if it detects that this parameter falls under a given threshold, it must look for some other available networks, choose the best one, according to some predefined user's preferences, and eventually take care of the access protocol.

The program was firstly designed for WLANs' management, but since the beginning we created a basic structure easily extensible to every useful wireless TCP/IP protocol: the program was then tested simulating a GPRS connection interacting with existing WLANs, while the real implementation of the procedures regarding other wireless standards (first of all, GPRS and UMTS) is in progress.

In our project, we proposed to build a software that could run on Linux systems, on one side to enjoy the convenience of reusing open-source code and on the other because our robot is actually controlled by programs running on this operating system.

The programming language chosen is *C++*, because we wanted both to reuse kernel source code (ordinarily written in *C*-language) and have the possibility offered by the object-oriented programming. We also realized a graphical user interface, useless for the robot (we remind that the programs behaves like a daemon), but interesting for tests or demonstrations; for this interface we focused on the *Gtk* environment, including the *gtkmm* libraries.

The starting point of our work are the Wireless Extensions for Linux [Tourrilhes, 1997]: these are an API that extend the Linux kernel, improving it with a uniform interface to manipulate wireless computer devices. The Wireless Extensions include programs like *ifconfig* and *iwlist*, support almost all the drivers for WLAN devices and have become *de facto* a standard for WLAN management on Linux systems.

In particular, we use the *madwifi* driver to control a *3com 3RCPAG175* (11 a/b/g) PC card; accessing protected networks is granted by the *wpa_supplicant* software.

B. Starting point

The Wireless Extensions implement an operation called "default scan" (executed by the bash command: *iwlist ifname scanning*), by which it is possible to recover information on the environment. More in details, the interesting data we can extract from the environment are:

- the SSID of every access point (AP) whose signal is sensed by the antenna receiver;
- the strength of the signal received from each AP;
- the protocol used on every network;
- the presence or not of a WPA protection on every network;
- the frequencies and bit-rates available on every AP.

As we can see, the *scan* provides all the information we desire, with the only cost that this operation is implemented by a temporary interruption of every useful data transmission: the device just stops sending and receiving messages for a few seconds ($5 \div 10$s) and listens to the periodical beacons that every AP sends at regular fixed time intervals. These beacons are exclusively sent by the APs to let wireless devices recognize them. So, *scan* is a powerful operation but we must not abuse of it.

Instead, there is another interesting function implemented in the Wireless Extensions, that consists in an endless and painless monitoring of the current connection: in fact, because it is the only connection continuously used, the driver has always fresh data regarding its quality. In particular, the data are the Signal/Noise Ratio (SNR) of the received signal and the Packet Error Rate (PER).

C. Working algorithm

Thanks to this feature, we can come to a satisfying solution: our program can monitor the present connection, without doing anything else until the link remains good; even as the quality of the link drops under a predefined threshold, the program must stop the communication for a few seconds and look for a better network: probably, in fact, in this case the robot has moved too far from the previous access point and now a network roaming is needed. Our program works exactly in this way, on the basic idea of "doing nothing if all works well". Moreover, there is an option by which the user can decide to scan anyway the environment, setting a (slow) timer properly.

Let us now examine what happens when a new better network is found. In this situation, the robot must disassociate from the previous – bad – access point and establish a connection to the new – good – link. The low-level operations for authenticating are still provided by the Wireless Extensions (we can use something like the bash command *iwconfig*), combined with the Dynamic Host Configuration Protocol (DHCP).

DHCP is the standard protocol that allows a device to ask for and to obtain an IP address from a (DHCP) server. This is actually the most convenient way to rapidly log on a network, especially if you are thinking about a wireless network.

D. The choice algorithm

The most innovative feature of our program, we said, is the choice of the best network. This operation is performed case-by-case by the robot, that in fact simply behaves in the way it has been programmed. There are some parameters that can be set by the user, such as the threshold above which a connection has to be considered bad, the time intervals between two *scan*s, some personal preferences on networks that should or should not be used by the robot,

the operations to perform in some particular cases, e.g. the roaming on a protected network usually requires a script to run a supplicant before trying to connect to any access point.

Finally, the choice algorithm includes a section to inhibit the access to a list of networks. This list can be updated both by the user, who for instance could not want to allow the robot to connect to all the enterprise WLANs, or by the program itself: if the robot can't connect to a specific network, e.g. due to a MAC restriction, the program would better remember this limitation, to avoid wasting time trying to access to that network in the future.

E. Other features

There are also some other criteria we kept in mind while writing the program: the first one is to create a software that could simply be expanded adding other wireless standard protocols. In fact, a user may want to use a specific transmission protocol actually not implemented in our project, or, in the future, new protocols may arise and the program must be prepared to the new demand.

Then, there is a function through which an authorized user can ask the program some useful information about the current connection, especially referred to its IP address and the router address: in this way we obtain a full traceability of the robot, that we can, for instance, control from a remote computer.

IV. Conclusion and perspectives

In conclusion, in this work it has been realized a software that allows a mobile autonomous robot to manage the access to wireless local area networks by itself.

By this package, the robot is always able to communicate, if it is in a place covered by a wireless network. The program can manage either open or protected networks and execute specific operations to associate to each of them.

In Figure 4 we can see a snapshot of the program, running at the University during a test.

Fig. 4. The user interface of the program running.

In the monitor window, that is the main window, we can see the data regarding the environment, extracted on the last *scan* operation.

The antenna received signals from four WLANs; the list of available connections is ordered by the quality property. The best of these network is the one called "ATRIO", from which it was received a –65dBm power signal, which is an acceptable level, considering the fact that its associated icon is a green ball. The worst available connection is instead "SALACONSILIARE", with a –84dBm power signal; this link is not considered acceptable by the program settings (as we see from its red icon) and so it could not actually be chosen as a network to associate with. Anyway, at this moment there is no need to change the current connection, because that the robot is exploiting the network "ATRIO", to which it is connected receiving a –68dBm signal, as the monitor window shows at the bottom.

Icons on the left, then, indicate whether or not the found networks are protected: here for instance we can deduce that "ATRIO" and "AULAMAGNA" are WPA-protected networks, while "SALACONSILIARE" is a protected network too, but we do not have the rights to access to it; at last, "Vodafone" is a GPRS not protected network.

Exactly, "Vodafone" is a GPRS network that we are simulating by a program working in parallel, to test the robustness of our system, exploiting various types of communication protocols.

The data collected about the environment are not only printed on the screen and then "thrown away": if the system is enhanced with a localizer, every scan operation can be referred to a position in a global map of the working area. So the system has the capability to record the results of every scan, storing them in an XML–formatted file. XML standard was chosen just for its ease of use and its wide compatibility with many applications.

The monitor window also let the user directly order some of the program automatic features: the button `"Refresh"`, for instance, forces a *scan* operation, while the buttons on the right start the procedures of association and authentication on the respective networks.

These features, executed in manual mode, are clearly not related to robotics, but have been added to the program to help the user during configuration tests of the system.

We performed many tests on the system, making the robot move through the University departments: we verified that network roaming happens correctly from a WLAN to another when the transmission is difficult due to the distance to the associated access point, while, if no WLAN is found, the robot solves the situation connecting to a simulated GPRS network. When a WLAN becomes then available again, the program pulls down the connection to the GPRS network and authenticates itself on the WLAN just found. This behaviour derives from the settings we gave to the program, that tell the robot, for instance, to prefer a WLAN if available.

Network roaming is not exactly seamless, i.e. there could be a little packet loss during the handoff, but this was yet

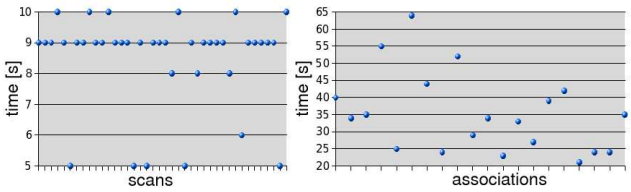

Fig. 5. Measurement performed during tests, regarding scan and association delays during the handover.

contemplated in the project: the way chosen to disassociate from a network and authenticate with a new access point implies a little delay, during which sent packets are unavoidably lost. This delay was measured as a maximum of 10s to scan the environment looking for new available connections, added to about 20÷40s to get a new IP address from the DHCP server, time that can sometime vary due to the congestion of the network and the capabilities derived from the DHCP handshake (e.g. lease time) [Droms, 1997].

Timing and features are indeed compatible with the needs of those robotics applications that need an endless communication between a community of devices, though a sporadic lack of connection can be admitted.

In conclusion, in this work, we realized such a general-purpose software that could fit with a very large number of other projects and, in general, could be useful in almost all applications involving mobile robots communicating through wireless networks.

Thanks to the variety of uses of wireless networks, this project is rich of possible future improvements: let us now describe the most interesting ones.

A. Extension to other wireless standards

Actually the program was intensively tested for WLANs' management and its right behaviour in presence of other kind of networks is granted by the simulation of a GPRS network. Naturally, the first desirable upgrade consists in extending all the features already implemented for WLANs to other wireless standards, such as, first of all, GPRS and UMTS.

The program core is already thought for it and therefore does not have to be modified. What it should be added regards the lower-level features, such as the driver management and some considerations about the characteristics of the signal quality, that surely differ from the ones made for WLAN signals. Of course, we'll have to upgrade the hardware too and substitute the WLAN PC card with one of the new cards that integrate UMTS, GPRS and WLAN accessibility.

B. Integration in DCDT

The Device Community Development Toolkit (DCDT) is a project carried out by the Department of Electronics for Automation of the University of Brescia, that consists in a message oriented middleware to enable asynchronous exchange of data, event notification, persistence, quality of service and ease of development, to implement a communication layer among different devices, making the underlying media transparent to the developer [Cassinis et al., 2001].

It would be very useful to extend the capabilities of the realized system by letting it interfacing with the DCDT. In this way, it could be possible for every device of the community to ask the robot for some information about its connection to the LAN, for instance to which network it is associated or its IP address, gaining the aim of a complete traceability of the robot.

Moreover, the implementation of the communication through the DCDT would guarantee the delivery of the messages between members of the community even if an adressee is not currently connected to any network (i.e. it is roaming to a better network), solving the problem of the loss of information during the temporary lacks of connection. In fact, the *post office* provided by the DCDT architecture keeps sent messages in memory until all the interested members have read them.

C. Context-aware implementations

For now, the system is completely independent of the context in which the robot is placed. On one side, this can be seen as a good characteristic of the software, because it is able to work in a fully unknown environment: it has no need of being hardly configured, because it gets by itself all the information it needs. But usually a mobile robot works in a limited area, often already known.

If we manage to mix the information about the localization of the robot and the data detected *in loco*, we can save all those information in maps, that can be used by the robot to either optimize the connection with the networks it foresees to find or viceversa change his route to reach better connections.

For example, consider the case represented in Figure 6:

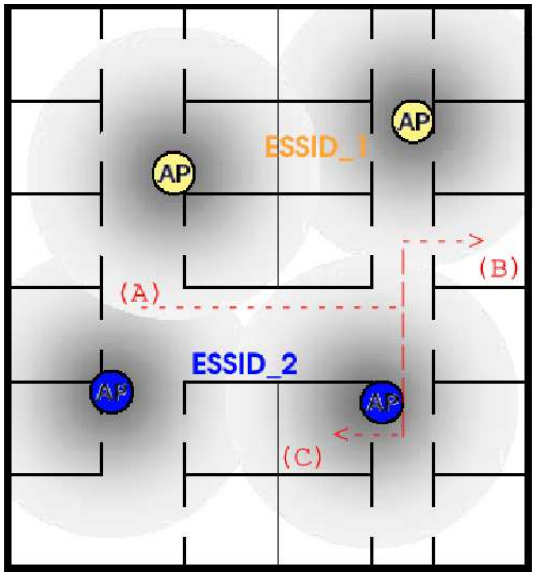

Fig. 6. Example of context-awareness: gradient maps obtained by mixing sensing and localization information.

here we have a typical scenario we can find in an enterprise. As we see, there are two WLANs, with different ESSID, every one including two access points. The robot is initially in the position marked with the "(A)" label and it is switched off, so not connected to any network; suppose then that someone must deliver two packs, one into the room marked as "(B)" and the second into the room marked as "(C)" and switches on the robot assigning to it this job.

If the robot had a gradient map of the wireless networks signal quality, it would have no doubt about which network to associate with: of course it would choose the one with ESSID_2, because it assures a better coverage on the route it is going to take. If the robot instead had not got these information, it would probably would choose almost randomly, considering that in the initial position "(A)" he receives from the two access points signals with about the same strength.

Context-awareness should be also useful for the robot when it needs to send a message quickly or to have an urgent information, but it is in a place without wireless networks' coverage. In this case, having a map of the wireless networks, the problem should be solved easily by moving to the nearest point with an available connection.

Moreover, the information on the environment could also be shared with other users or devices: the XML format of the file that stores the data learned during scans, already implemented in the realized software, ensures an easy expansion of the system. In particular, it would be very interesting to develop an application based on a web–server for the remote watching of the data, that could be mixed up to build interactive maps enjoying the capabilities of the Scalable Vector Graphics (SVG) language, that is directly compatible with the XML format.

As for the integration of the robot navigation maps with the data about the strength of the signals from wireless networks, a localization system is clearly needed. About this subject, a possible solution is proposed in [Cassinis et al., 2005b], where a system based on a webcam and active markers for outdoor and indoor robot localization is presented. This system is also integrated with DCDT, so the robot, like any other member of the community, should be able to know its position just asking it to the dedicated device.

D. Use of different interfaces

The current project involves only one network interface, both to communicate and to look for other available networks.

This choice makes it possible to install this system without any hardware addition, but causes some delay during the handoffs, because the interface can not be used to communicate during the scan procedures. Moreover, this characteristic limits the scan frequency, because the communication can not be continuously interrupted to analyse the environment.

Therefore, a possible future improvement of the system should be to provide it with a further network interface: the system should then exploit the first one to communicate with the associated network and the second one to continuously look for the available connections, measuring the power of their signals.

REFERENCES

[3com, 2003] 3com (2003). Deploying 802.11 wireless LANs.

[3GPP, 2002] 3GPP (2002). General Packet Radio Service (GPRS) Service description; Stage 2. TS 03.60, 3rd Generation Partnership Project (3GPP).

[Cassinis et al., 2001] Cassinis, R., Meriggi, P., Bonarini, A., and Matteucci, M. (2001). Device communities developement toolkit: an introduction. In *Eurobot*, pages 155–161, Lund, Sweden.

[Cassinis et al., 2005a] Cassinis, R., Tampalini, F., Bartolini, P., and Fedrigotti, R. (2005a). Docking and Charging System for Autonomous Mobile Robots. Technical report, Università degli Studi di Brescia. `http://www.ing.unibs.it/~cassinis/docs/papers/05_008.pdf`.

[Cassinis et al., 2005b] Cassinis, R., Tampalini, F., and Fedrigotti, R. (2005b). Active Markers for Outdoor and Indoor Robot Localiziation. Technical report, Università degli Studi di Brescia. `http://www.ing.unibs.it/~cassinis/docs/papers/05_009.pdf`.

[Cisco Systems, 1995] Cisco Systems (1995). Virtual LAN communications. `http://www.cisco.com/warp/public/cc/pd/wr2k/cpbn/tech/vlan_wp.pdf`.

[Droms, 1997] Droms, R. (1997). Dynamic Host Configuration Protocol.

[Droms and Arbaugh, 2001] Droms, R. and Arbaugh, W. (2001). *Authentication for DHCP messages*. Internet Engineering Task Force (IETF).

[Dundar and Puri, 2002] Dundar, B. and Puri, A. (2002). IP layer mobility across heterogeneous wireless networks. In *International Conference on Third Generation Wireless and Beyond*, Berkeley, CA, 94720, USA. University of California - Department of Electrical Engineering and Computer Sciences.

[IEEE, 1997] IEEE (1997). *LAN MAN Standards Committee of the IEEE Computer Society. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE.

[IEEE, 2003] IEEE (2003). *IEEE 802.11F Standard - IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation*. IEEE Wireless Communications.

[Kanter, 2002] Kanter, T. G. (2002). Hottown, enabling context-aware and extensible mobile interactive spaces. *IEEE Wireless Communications*.

[Krishnamurthy et al., 1998] Krishnamurthy, P., Pahlavan, K., Zahedi, A., Vallström, J., Talvitie, J., Pichna, R., and Ylianttila, M. (1998). Handoff in 3g non-homogenous mobile data networks. In *MTT-S European Wireless*, Amsterdam.

[Salkintzis et al., 2002] Salkintzis, A. K., Fors, C., and Pazhyannur, R. (2002). WLAN-GPRS integration for next-generation mobile data networks. *IEEE Wireless Communications*.

[Sällström, 2003] Sällström, F. (2003). IP mobility vs session mobility for wireless VPN.

[Shankar et al., 2001] Shankar, N., Arbaugh, W., and Zhang, K. (2001). A transparent key managenent scheme for wireless LANs using DHCP.

[Tampalini and Cassinis, 2005] Tampalini, F. and Cassinis, R. (2005). Fuzzy logic controller based on XML formatted files for behaviour–based mobile robots. Technical report, Università degli Studi di Brescia. `http://www.ing.unibs.it/~cassinis/docs/papers/05_011.pdf`.

[Tourrilhes, 1997] Tourrilhes, J. (1997). *Wireless Extensions for Linux*. `http://www/hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html`.

[Valkò, 1999] Valkò, A. G. (1999). Cellular IP: a new approach to internet host mobility. *SIGCOMM Comput. Commun. Rev.*, 29(1):50–65.