# Supporting Distributed Applications for Swarm of Robots within Smart Environments: The Way of EU Project DustBot

Francesco Chiti, Romano Fantacci, Giovanni Collodi, Gianfranco Manes, Luca Bencini, David Lund, Bassem Ammar, Ioannis Katsaros, Alistair Doswald, Stephan Robert, Peter Sollberger

***Abstract* — This paper deals with a communication infrastructure needed to allow a swarm of robots performing dust cleaning and garbage collection task in an urban area. It outlines the required communication links, analyses them regarding security, describes in details the implementation and figure out some performance test results.**

***Index Terms*— Wireless Sensor Networks, Mobile Sink, Swarm of Robots, Communication Protocols, Security.**

## I. Introduction

THIS paper deals with a communication infrastructure for a swarm of robots that is about to be setup for the EU project DustBot (FP6-IST-045299-STREP).

The DustBot project is aimed at designing, developing, testing and demonstrating a system for improving the management of urban hygiene based on a network of autonomous and cooperating robots, embedded in an Ambient Intelligence infrastructure.

The robots will be able to operate in partially unstructured environments and to vacuum-clean them from rubbish and dirt. They will be able to transport small quantities of home garbage, collected on demand from citizens, at their doors. By using preloaded information on the environment and inputs from on-board and external sensory systems, and by taking advantage of the benefits provided by the Ambient Intelligence (AmI) platform, the robots will be able to move with a proper level of autonomy to carry out their tasks.

The communication infrastructure is a vital part of the DustBot project due to the fact that the robots should be able to fulfil their job autonomously in an urban environment. The communications infrastructure is needed to efficiently deliver sensed information and commands. This allows the remote user to effectively interact with the robot network and to monitor a particular area, but also makes the robots capable of autonomously coordinating themselves and to perform complex tasks.

The whole communication is based upon different wireless standards that are fully integrated and allow a seamless handover between them. Important issues aside from the required functionality are the security concerns that have to be considered. This is due to the fact that the used communication technologies are spread in the public domain and are often exposed to attack attempts which may compromise the desired functionality. A failure of the communication channel could result in an unexpected reaction of the robots which may cause possible danger to the public. This has to be avoided in any case.

The communication infrastructure that is implemented covers all aspects from the maintenance access to the information transfer between the robots, the landmarks and the backend system (AmICore). The system is designed in a highly modular way with implementation of different functionalities in separate modules for simple and efficient expandability and extensibility as well as the handling of the inherent complexity.

A first chapter will introduce the various communication links and explain their missions. The next chapter presents the results of a security risk analysis, which then is followed by design explanations. Finally several conclusion regarding possible development and testing procedures are drawn.

## II. Communication Infrastructure Overview

The communication infrastructure is split into several parts. As shown in Figure II-1 there are several different communication needs. In the following subchapters each of these communication needs will be described in more detail.

Authors are with the Department of Electronics and Telecommunications, University of Florence, via di S. Marta 3, 50139 Florence (Italy) (corresponding author to provide phone: +39-055-4796270; fax: +39-055-494569; e-mail: name.surname@unifi.it).

I. Katsaros, D. Lund and I. Katsaros are with HW communications Ltd, Parkfield, Lancaster, LA1 4TZ (UK).

Alistair Doswald is with the Institute for information and Communication Systems (IICT), Haute Ecole d'Ingénierie et de Gestion, Vaud, Rte de Cheseaux 1, 1401 Yverdon (Switzerland).

Peter Sollberger is with the Competence Center Electronic of the Lucerne University of Applied Science and Arts, Technikumstr. 21, 6048 Horw (Switzerland).

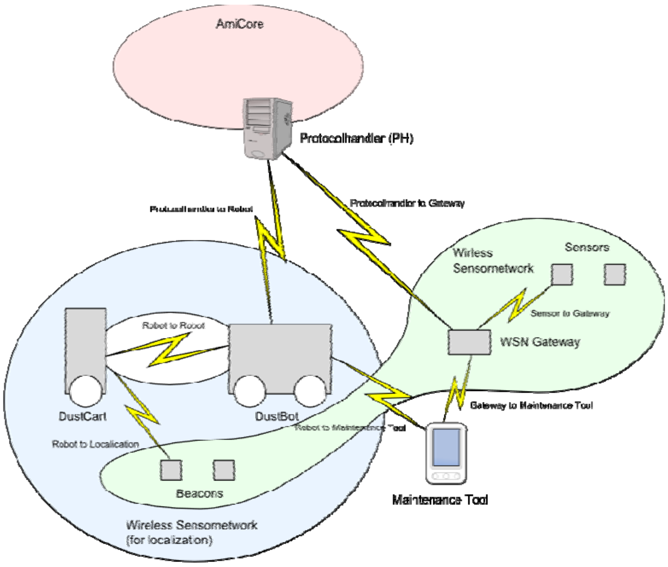

**Figure II-1: Communication Infrastructure Overview**

The robot itself needs to maintain several different communication channels:
A) from/to other robots.
B) from/to the Protocol Handler (PH), which is the defined entry/exit point of the ambient intelligence core (AmICore),
C) from/to beacon nodes for localization and
D) from/to a supervision and maintenance tool.

DustBot robots are also part of a larger wireless sensor network (beacon nodes also serve as sensor nodes) and the information from that network must also be sent over the Internet via the Protocol Handler to be processed by the AmICore. A WSN Gateway computer communicates on one side with the wireless sensor network, and on the other with the Protocol Handler. As such, we decided that the network characteristics of the WSN Gateway will be similar to that of the robots.

The WSN communication channel between the sensor nodes is no more within this paper.

### *A. Robot to Robot*

To allow robots to directly exchange data when in range, an IEEE802.11 connection will be used. This however is only possible with a separate network card and with fixed network settings, including those settings for encryption.

### *B. Robot to Protocol Handler*

The purpose of the Robot to Protocol Handler communication link is to connect the robot to the core of the ambient intelligence system (AmICore), passing by the Internet. The link allows information from the robots sensors to be processed by the ambient intelligence, and instructions from the ambient intelligence to reach the robots.

The main requirement for the link is to maintain as well as possible a continuous TCP/IP, no matter where the robot may move, which is a problem when the robots move from one network to another. Another requirement is that we may only install minimal infrastructure, if any, to support the link.

For the DustBot project, we chose to use Mobile IPv4, which allows us to use heterogeneous network technologies while maintaining the necessary TCP/IP link. Mobile IPv6 would also be a possibility, but it’s less interesting with the current network infrastructure (problems include longer packet size, IPv6 mobility over IPv4, NAT traversal). For the physical link, we use a solution that integrates 2G/ 3G communication (HSUPA, HSDPA, UMTS, EDGE and GPRS) and WLAN. Using the mobile phone network enables us to have widespread Internet access without installing any infrastructure, while WLAN is cheap and easy to install, and some public places have a good coverage.

### *C. Robot to Localization*

The localization of a robot moving around a certain operation area is accomplished by means of interacting with a Beacons Network (BN) which is comprised of fixed nodes and are aware of their position (absolute coordinates). In particular, a robot is able to associate the BN Nodes (BNNs) falling within its coverage radius and to send them a *beacon message*; upon the reception of which BNNs reply with a *ranging message* (either in the Receive Signal Strength Indicator (RSSI) or ultrasonic domain). Finally, the robot collects these packets and estimates its own position through a 2D trilateration algorithm implemented in the integrated *localization engine*.

### *D. Robot to Maintenance Tool*

The Maintenance Tool is a mobile device that is used to configure and supervise the robot on site. A wireless link is preferred since supervision then can also take place if the robot is moving. As this is a short range connection and there is typically only one maintenance tool connected at time, Bluetooth will be used. However, the supervision and maintenance application may also be used remotely via AmICore. Therefore, this application will use a TCP/IP protocol. And since Bluetooth offers the Personal Area Networking (PAN) profile the robot gets an IPv4 connection over Bluetooth to the maintenance tool.

## III. SECURITY RISK ANALYSIS

### *A. Robot to Robot*

#### *1) Authentication and Public Key Infrastructure PKI*

Since the Robots are going to be connected in an ad-hoc IEEE 802.11 network the implications for the security of the link are numerous and must be carefully addressed to see what is in scope for the particular proposal and what should not be considered at all. There are several issues that are fundamental to the security of an ad-hoc network purely because of its topology and architectural properties. Key management is certainly an issue particularly for the case where authentication and hence some form of PKI is required. In a scenario where robots are freely joining and leaving the network, a mechanism needs to be in place that authenticates new nodes upon joining the network. This can be easily achieved if the robots are pre-configured with their own certificates and accompanying private keys. Private keys can be frequently redistributed either when the robots are in the docking stations or via the network using a predefined key management mechanism. The fact of having a PKI in place makes the task of encrypting traffic between robots and from the robots to the AmiCore an easy one. Established protocols, primitives and PKI schemes can be used such as the X.509 framework. For the case of the DustBot network the AmiCore can be the network entity in charge of administering certificates, key-pairs and also maintaining the revocation lists.

### 2) *Routing and security*

Since routing is of vital importance to an ad-hoc network, the security of the routing mechanisms are also of vital importance. Given that the robots will be running the established AODV (Ad-hoc On demand Distance Vector, RFC3561) routing protocol and should we establish that protecting the routing mechanism is in scope, the secure version (SAODV) of the AODV protocol, provides an ideal solution

## B. *Robot to Protocol Handler*

### 1) *Standard security issues*

As the communication link between the robot and the Protocol Handler is in many ways a standard wireless communication link, it suffers from all the risks of passing through a public network. The two different modes of wireless communication (WLAN and 3G) have very different associated security handling.

- The security of the 3G communication is completely out of our control. The security on connection and within the operator’s network must be trusted for issues such as man-in-the-middle attacks and denial of service. As for the radio signal itself, it is at risk of denial of service through a phone jammer. While such a device is illegal in most countries, it remains available, and it is quite cheap (can be found under 300$ for a 10m block).
- The WLAN connection may be problematic as well. The robot can connect to any available public network, and will choose to use it as long as the available QoS is sufficient. This means that anyone with an access point can open it up to capture the robots signal path (but not to effect a DoS, as the robot will choose another network if this one does not allow it to contact its Home Agent). As such, man-in-the-middle type attacks are easily possible. With enough WLAN devices, an attacker can also create a DoS on the WLAN, by covering all available channels, while masquerading it as legal WLAN usage (such an attack can also block Bluetooth communication).

Although it is possible to make sure that the privacy of the information carried on the link is secure through encryption, it is impossible to guarantee availability of the link. A denial of service attack directly against the physical layer for WLAN and 3G communication can prevent all communication. In such cases, it may be worth considering the zigbee connection to relay a message, but in any case the AmICore should report any robot that has no signal for an extended period of time.

### 2) *Mobile IP security issues*

The security issues that affect Mobile IP depend not only on the protocol itself [1] and its extensions (such as Mobile IP Traversal of Network Address Translation (NAT) [3]), but also on the manner in which the protocol is implemented. There are a few general security concerns, but some are specific either to the mobile node or to the home agent. As the mobile IP installation we intend to set-up is limited to the robots, and does not need to dynamically take into account new additions to the network, the associated security infrastructure is relatively simple.

Most security concerns for the mobile IP protocols are outlined in the RFCs that define them. General security concerns for the basic mobile IP protocol are:

- Authentication: by default, mobile IP messages are authenticated using HMAC-MD5 with a minimum key size of 128 bits. All authentication associations are available (Mobile Node (MN)-Home Agent (HA), MN-Foreign Agent (FA), FA-HA), however the MN-FA authentication does not guarantee protection against replay attacks. This should not be a problem as we do not intend to use FAs, but will become a serious flaw if we do. Without authentication, the protocol is subject to a host of attacks, an example being the depletion of resources through register messages.
- Privacy: this may be of concern. Data analysis can be prevented though SSL encryption, but we will probably have to deal with traffic analysis. The only way we can prevent this from being a concern is if the information acquired through traffic analysis cannot be used for an attack, meaning that the end systems have to be sufficiently secure.
- Replay protection: protection against replay attacks is of critical importance. Mobile IP defines two methods to prevent replay: timestamps and nonces. However, since latency will change radically depending on the link type, it may be difficult to choose a good time-before-reject. Nonces may be the preferred solution.
- Use of gratuitous ARP and proxy ARP: on the home network ARP is used to bind the link layer address either to the mobile node or to the home agent. As the protocol has no authentication, a machine on the home network can use the same mechanism to redirect traffic to it (although as almost all networks use ARP anyway, this is probably not a problem).
- The tunnel itself is unauthenticated, meaning that validation of the data from a correspondent node must be done with end to end authentication/encryption, even though tunnel messages will only be accepted if they match the registration details (source IP, etc…).

#### a) *Mobile node security risks*

As long as the general security risks are addressed, the mobile node itself suffers from one security risk: a redirection (man in the middle) attack through a rogue Foreign Agent (FA). Normally a mobile node should use a foreign agent if it’s present on a network, but this means trusting that agent. We may choose not to use an unauthenticated foreign agent, but we'll lose the use of potential useful foreign agents. Besides, an attacker could achieve the same goal far more simply by using an access point.

##### *b) Home agent security risks*

All security risks to the home agent, aside from the general mobile IP risks, stem from the fact that it must hold a public IP address. Being directly reachable from the internet, the machine itself and all communicating software are susceptible of being attacked. As such, the security of the server the HA resides on is a priority.

The use of Mobile IP Traversal of NAT[3] adds an important extra security risk: the passage through a NAT means that the source IP address is different than the one displayed in the authentication message. The home agent will use the difference to determine that the mobile node is behind a NAT, and will add the source port/address to its table. However, as this information is not authenticated, an attacker could modify it to route all information to a machine of his choice.

##### *c) Security features implemented in Dynamics mobile IP*

The open-source software that our mobile IP implementation is based on respects all security features of RFC3344. It provides both nonces and timestamps against replay protection, and it supports all required authentication methods, as well as SHA-1 and HMAC-SHA1 (extra algorithms are not forbidden by the RFC). The software also has key delivery mechanisms, but if we intend to use them, they should be updated to the latest RFCs.

### *C. Robot to Localization*

The use of IEEE 802.15.4 for the Robot to Localization link allows for the usage of standardised mechanisms for the protection of the data carried over it. The ZigBee standard comes with a fairly robust end-to-end security architecture that suffices the DustBot requirements. For encryption the AES algorithm is used and for key exchange the Elliptic Curve MQV algorithm is used.

The end-to-end encryption between any two nodes is achieved by encrypting data using the pre-negotiated key but no authentication mechanism is in place for directing traffic through nodes that have not been authenticated. For the time being addressing such threats is out of scope since introducing rogue sensors in order to deploy replay attacks and exploit the lack of an authentication mechanism is not going to provide an attacker with any significant information regarding the location of the robot. A threat that might be more realistic is that of comparing ciphertext for the purpose of deducing information about the location of the robots, but again the feasibility of this attack would depend upon the plaintext space and the protection mechanisms in place for that (adequate padding, key renewals etc.).

### *D. Robot to Maintenance Tool*

As the underlying access technology for the particular link is going to be Bluetooth the security mechanisms and potential security risks associated with Bluetooth need to be examined. There are several refereed articles[4][5] that describe attacks and vulnerabilities of the protection mechanisms used by the protocols but most are based upon weak versions of the original specification. It is therefore necessary to examine reports of security threats and choose the protocol version that suffers the least from these vulnerabilities. In order to avoid the lesser of two evils policy, additional mechanisms need to be in place to strengthen the security of the Bluetooth link. It is also important to avoid attacks that are not aimed at the link itself but at the maintenance device and the robot. In other words user authentication needs to be in place for the trained personnel that are going to be operating it and vital configuration information for the robot needs to be safely stored. This will be easily achieved by password protecting the devices used for maintenance and by making sure that the vital configuration information stored in the robot's memory is also encrypted.

### *E. Other equivalent Links*

The security risks for the remaining communication links are equivalent to the risks for the four types mentioned above. The remaining links are; the Gateway to PH link and its risks are almost equivalent to the Robot to PH link. The Gateway to Maintenance tool link is the same as the Robot to Maintenance Tool link.

## IV. Communication Infrastructure Design

### *A. Robot to Robot*

After setting up all dedicated network cards for the Robot to Robot communication, network traffic in the form of a heartbeat signal must be generated by each robot.

The heartbeat message is an UDP packet containing the name of the robot itself and the IP addresses of its surrounding robots. If a robot receives such a heartbeat message from another robot that contains its IP address, it may establish a connection oriented TCP/IP link.

### *B. Robot to Protocol Handler*

DustBot will maintain as well as possible a permanent TCP/IP link to the Protocol Handler. The structure to maintain that connection can be described in two parts: the physical connection to the access point (AP) or antenna, and the Mobile IP protocol that allows the data stream between the DustBot and the Protocol Handler to be maintained. The connection will not be managed by single unified software, but rather by several software components each handling a specific part of the connection.

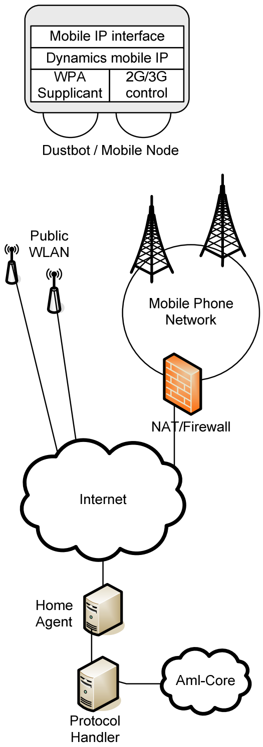

**Figure IV-1: Robot to Protocol Handler Infrastructure design.**

Maintaining the physical connection is a matter of deciding when to pass from an AP/antennae to another, and if multiple connections are possible, which one to choose. Considerations are latency, bandwidth, the quality of the link and cost. While the 2G/3G connection will almost certainly have better coverage, it has the disadvantage of being slower and more costly then passing through WLAN, since using 2G/3G networks is usually charged on the quantity of traffic sent of received (at least in Europe).

The open-source software component that manages the WLAN roaming is WPA supplicant, which allows the dustbots to roam between unsecured public networks and the secured APs that may be installed for the project, while of course avoiding other secured APs. A custom interface for WPA supplicant is being developed to blacklist APs that are unsecured, but do not route to the Protocol Handler. In the case of multiple available APs, the software will prefer APs installed for the DustBot, and then make a decision based on the quality of the link.

The software that manages the 2G/3G link has yet to be written, but it is planned to work as follows: it makes sure that a connection to the PH is always established (through a keep-alive signal), and selects the link technology that provides the best bandwidth/latency, as long as it has a minimal link quality (to be determined through testing). The data link layer is handled by the Point-to-Point Protocol (PPP).

The connection used for the actual data is determined by the Mobile IP software. The full description of the Mobile IPv4 protocol can be found in RFC 3344 [7], but a brief description will be given here. The Mobile IP protocol allows a mobile computer (the DustBot in our case), known as a Mobile Node (MN) to retain a permanent IP address (called home address) even when receives a new address (called care-of address) as it moves to a new network. In order to manage this, the mobile node relies on a server called Home Agent (HA), which is informed whenever the mobile node acquires a new address, which it associates to the MN. The MN communicates with Internet machines using its home address, either directly or by using the HA as a relay, in which case the data is sent over an IP-over-IP tunnel. Traffic to the MN is intercepted by the HA, then tunneled to the care-of-address of the MN. Of course, the home address must point to the network the HA resides in. An optional Mobile IP server, called the Foreign Agent (FA), may be installed within networks to handle MNs. FAs can assign separate addresses to the Mobile Nodes, serve as the end point for tunneling (resulting in less information to be sent over the wireless link), provide finer security management and enable faster handovers between networks.

Several other RFCs extend the Mobile IP protocol to increase its functionality, security and performance. The most important of those for the DustBot project is RFC 3519, which describes a modification to allow Mobile IP to work even when the FAs and/or MNs are behind NATs by using IP-over-UDP tunnels. This is necessary as public WLANs and 2G/3G networks usually are behind a NAT.

For the Dustbot project we will likely only use an HA and MN software, as the networks that the DustBot pass through will rarely be administered by us. The software for HA and MN (and FA where possible) is based on the open-source Dynamics Mobile IP. It is being modified to support IP-over-UDP tunnels (as described in RFC 3519) and faster handovers. The decision of which connection to use, WLAN or 2G/3G, is taken by a control interface for the Mobile IP software. When both links are available, WLAN is preferred unless it shows to be performing significantly worse then the 2G/3G connection. The software will attempt to anticipate when a temporary interruption will occur in one of the links (for example a break in the WLAN connection due to the passage from one AP to another) and if possible start using the other before the break occurs. The resulting communication and handovers should unbroken unless no connection whatsoever is present.

### C. *Robot to Localization*

The proposed architecture is depicted in Figure IV-2, where mobile robots, BNNs and other wireless nodes are shown.
In particular, four cases are presented which are related with the primitives of IEEE 802.15.4 standard:

- *set up phase*: mobile robot sends an ASSOCIATION.REQUEST message (a) to the BNNs within its coverage area, which reply with a ASSOCIATION.RENSPONSE message such that mobile robot can identify them (b).
- *polling phase*: mobile robot selects the better BNNs (e.g., basing on the link quality) and send to them a LOCALIZATION.REQUEST message (c). BNNs can acknowledge mobile robot with an optional

LOCALIZATION.RENSPONSE message notifying that they are ready to send an ultrasonic ranging message.

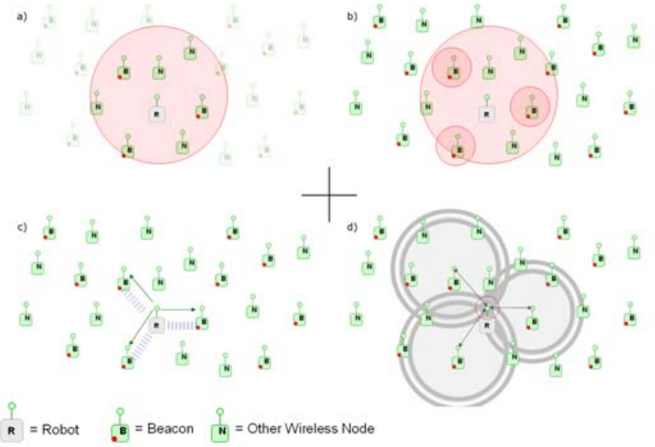

**Figure IV-2: The Beacons Network system architecture**

### D. *Robot to Maintenance Tool*

Bluetooth offers different defined profiles for different use cases. One such profile is the Personal Area Networking (PAN) profile. This profile offers two Bluetooth devices to get an IPv4 connection over Bluetooth.

#### 1) *User Scenarios*

There are general two scenarios possible. One is a group ad-hoc network and the other is a network access point. Both scenarios are shown within the Figure IV-3: Possible Bluetooth Connection for PAN BlueZ.

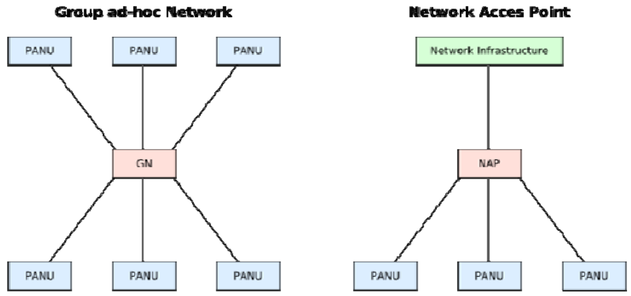

**Figure IV-3: Possible Bluetooth Connection for PAN BlueZ**

BlueZ [6] is the Bluetooth stack on Linux which will be used on the Robot and the GW. Each BlueZ node can have one of the following three roles:

- PAN user (PANU)
- Group ad-hoc Network (GN) controller
- Network Access Point (NAP)

The PAN user is a client of a PAN. It will connect to a GN or a NAP. The maintenance tool will be a PAN user. The GN is the central controller of a Group ad-hoc Network which allows up to seven nodes to connect to this ad-hoc network. The NAP is a router, proxy or bridge between the Bluetooth network and an existing network infrastructure such as the internet or a LAN. The NAP provides its service for up to 7 nodes. The robot will provide a GN since there is no need to connect to a further infrastructure. The difference between a GN and a NAP is mainly that the NAP has some additional rules to allow forwarding the network requests.

#### 2) *Usage within Dustbot*

The Bluetooth PAN connection will be used between the robot/gateway and the maintenance tool. The robot/gateway will take the role as the Group ad-hoc Network (GN) controller.

The maintenance tool will take the role of a PAN user (PANU) since it will access the robot/gateway to get an IP connection to the robot/gateway. The IP address will be provided from the GN/NAP to the PAN with a DHCP server.

## V. Conclusion

This paper showed that several different communication links may be used to allow a swarm of robots performing tasks in urban environment. The benefit of using all these various technologies lays in having a more robust infrastructure with redundant channels.

The security risk analyses pointed out possible attacks and countermeasures and hopefully alerts readers to also consider communication security when dealing with robot communication. In addition, it is shown that the costs for a secure communication infrastructure is acceptable and the available bandwidth is adequate for the needs of robot performing tasks in urban environment.

## Acknowledgment

This work was supported in part by the EU STREP FP6-IST-045299-STREP ”DustBot”.

## References

[1] C. Perkins, “RFC 3344-Mobility Support for IPv4”, 2002, http://www.faqs.org/rfcs/rfc3344.html

[2] C. Perkins, “IP Mobility Support for IPv4”, *RFC 3344*, 2002.

[3] H. Levkowertz, S.Vaarala, “RFC 3519-Mobile IP Traversal of Network Address Translation (NAT) Devices”, 2003, http://www.faqs.org/rfcs/rfc3519.html

[4] Markus Jakobsson, Susanne Wetzel, Security Weaknesses in Bluetooth, Lecture Notes in Computer Science, (2001).

[5] Y. Shaked and A. Wool, Cracking the Bluettoth PIN, In Proc. 3$^{rd}$ USENIX/ACM Conf. Mobile Systems, Applications, and Services (MobiSys), pages 39-50, Seattle, WA, June 2005.

[6] BlueZ – Official Linux Bluetooth protocol stack, Max Krasnyansky, Marcel Holtmann, http://www.bluez.org