Robust Multi-hop Communication for Mobile Applications

Stefan Herbrechtsmeier[1], Mohamed El-Habbal[1],
Ulrich Rückert[1], Ulf Witkowski[2]

[1]Heinz Nixdorf Institute, University of Paderborn, Fuerstenallee 11, 33102 Paderborn, Germany, hbmeier@hni.upb.de

[2]Electronics and Circuit Technology, South Westphalia University of Applied Sciences, 59494 Soest, Germany, witkowski@fh-swf.de

This paper focuses on node placement strategy, communication protocols and hardware implementation of a robust mobile multi-hop network. As part of the GUARDIANS project a main disaster scenario of a large industrial warehouse on fire is assumed. In this scenario, black smoke may fill large space of the warehouse which makes it very difficult for the firefighters to locate themselves and orientate in the building. In our approach a multi-hop ad-hoc network communication system is able to provide a robust communication system. Focus of this paper is the distribution of robots to guaranteeing robust communication between the base station outside the scenario area and the fire fighter inside the building. Concerning the first point, the presence of smoke and the indoor nature of our scenario prevent localization using camera and GPS respectively. Therefore the performance and accuracy of other sensors were tested in order to examine their usability and reliability in such a scenario, like ultra-sound, laser and radio signaling. The second point is the evaluation of routing protocols and the development of a mobile communication gateway which is optimized for the mobile usage and therefore supports different techniques for energy saving. Its characteristic was tested in different scenarios like a complex building and a parking garage.

Introduction

A robust wireless communication system can significantly help in various scenarios to enable specific actions or services and to increase human's safety. In this paper we focus on a communication system that has been developed in the context of the GUARDIANS project that is funded by the European Union 6th Framework Program (project no: 045269). In the GUARDIANS scenario, black smoke is assumed to fill large space of industrial warehouses, which makes it very difficult for the firefighters to orientate in the building, find victims or find their way out to exit the building. The main idea of the project is to send a heterogeneous team of robots inside the scenario area to assist fire fighters by detecting fire sources and hazardous gases, by providing different sensory, localization and positioning data and by maintaining communication links between fire fighters working inside the scenario area and the base station located outside. Main objective of this paper is to give an overview of required techniques to form a network by distributing communication nodes and by integrating a gateway communication module as a technical basis. A demonstrator has been implemented to test the robustness of the communication link between a fire fighter and the base station. BeBot robots [1] are used in this demonstrator, where one of them is the leading robot representing the fire fighter, and a group of robots is following it. These following robots should change their role, stop moving and act as relay nodes at specific places to build up a communication chain between the fire fighter and the base station establishing a so called multi-hop communication. To test the quality of the communication link, the leading robot is equipped with a camera and transmits a video to the base station using multi-hop communication via the robots in the chain acting as relay nodes. The first part of the paper describes how the following behavior and distribution of robots is realized to build up the chain, while the second part of the paper presents the developed routing platform in detail and shows how the routing is done via the underlying gateway modules.

Distribution of infrastructure nodes

Overview

In order to realize a robust communication system a proper distribution of communication nodes is required. The aim is to distribute the robots and place them with pre-determined spacing between each other, and to build up a communication chain between the base station and fire fighter. In order to do that, the robots should have two behaviors; the first is to perform robot following and the second is to act as relay nodes. For the first behavior, each robot has to be able to follow and to keep track of its predecessor. The distance between the robots is relatively short (within 2 meters). Two thresholds are used to regulate this distance; the first threshold (TH1) is the lower threshold, while the second (TH2) is the upper one. If the distance is smaller than TH1 (30 cm) the robot should stop, and if it is larger than TH2 (2m) the robot should increase its speed rapidly. In between the speed is also adapted according to the distance. For the second behavior the robots should only be able to receive commands to stop following and to act as a communication relay node if the communication threshold (TH3) between each two nodes exceeds 10 m. So, to perform the following behavior, each robot should obtain the distance and angle between itself and the followed robot. To measure these two parameters and to perform the following behavior three methods were tested; the first is using ultrasonic sensors (US), the second is using laser with object recognition and the third is using laser with infrared photo detectors (IrPDs).

Ultrasonic-based system

In order for a robot to follow another robot, it needs to know two parameters: the distance and the angle to the followed robot. To do that using US, the following robot needs to be equipped with at least two sensor groups. The US sensors (right-side and left-side receivers) deliver two distances to the target. Since the distance between the receivers is fixed and known, the distance and angle from mid-point of following robot to the followed robot is calculated using triangulation as presented in figure 1.

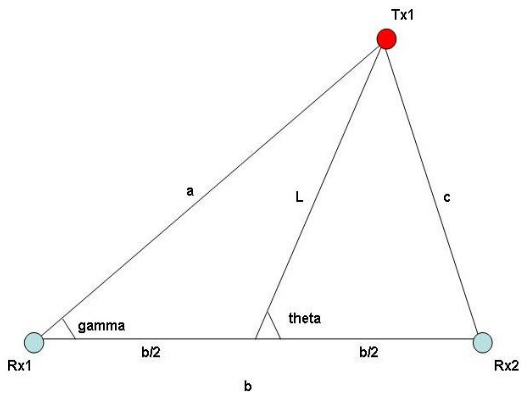

Figure 1: Angle and distance estimation of a target based on ultrasound measurements

Figure 2 shows the used structure for mounting the US sensors on a single robot, having four receivers and 3 transmitters (total of 7 sensors).

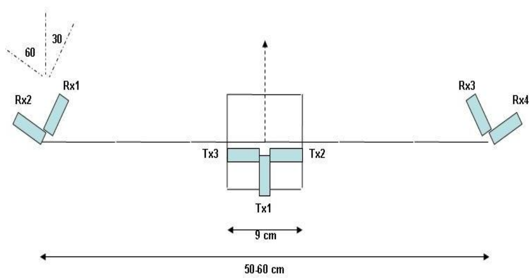

Figure 2: Structure for mounting US sensors on the BeBot robot platform

The separation between the right side and the left side sensors should be at least 50 cm to provide accurate results that are good enough for imple-

menting the robot following behaviour. The US receiver pins of Rx1 and Rx2 should be connected together (Rx_p1), in order to receive the fastest pulse possible without (or with minimum) reflections. The same goes for Rx3 and Rx4 (Rx_p2). The US transmitter pins of Tx1, Tx2 and Tx3 should be connected together (Tx_p1), in order to send pulses from all transmitters simultaneously and cover about 300 degree area.
Because the ultra-sonic signals cannot be (or are unlikely to be) encrypted, an ultra-sonic pulse sent by a transmitter (Tx) can be heard by any receiver (Rx) in range. Therefore, if multiple robots are performing ultra-sonic sensing, this should be organized and done sequentially, in order not to interfere or affect each other's measurements. So, for a robot to identify and measure the distance between it and another robot, it should ask the permission of the other robot by sending a request and waiting for an acknowledgement (ACK). Both robots should be synchronized so that the scanned robot (Rx) measures the time of flight (ToF) for the US pulse sent by the Tx (US_Tx pulse) to arrive at the Rx. Initially Bluetooth was planned to be used for synchronization, but after measuring the ToF, exactly the time for sending a request packet, which has a minimum size of 6 Bytes (5 Bytes header and 1 Byte data), till they are received by the Rx and the interrupt is entered, this time was found to be too large and not constant, about 12 -18 ms.

Therefore, other fast RF synchronization modules were used, which are the low power RF transceivers (TRx) working in the 433 MHz band. These modules have a low data rate, but due to their simplicity, sending a single pulse is quite enough and can be detected in a very short time. By measuring the Tof of the RF_Tx pulse, it appeared to take only 20 – 30 µs, which is about 1000 times faster than Bluetooth synchronization, and this time is short compared to the US ToF. This pulse, which is not encrypted of course, can still be heard by any RF TRx tuned to the same frequency. We thought of sending 1 byte addressing using this RF module, but it appeared to take too much time due to their low data rate, this is besides their weak robustness compared to wireless RF transmissions like Bluetooth and WiFi. So, we thought of using Bluetooth instead for addressing first, and then after receiving the ACK, the RF pulse is sent for synchronization, followed by the US pulse.

Laser-based with object recognition system
Another technique used for implementing the following behavior is using a Laser scanner that can recognize and identify objects. The used laser scanner is Hokuyo URG-04LX, having a scanning angle of 270 degrees, 1/3 degree resolution and range up to 4 m. The sensor delivers raw distance data and cannot recognize objects by itself like Radar sensors because of its very high resolution. Accordingly, in order to identify robots, special patterns needed to be mounted on the robots that are not likely to be found frequently in real scenarios. So, a star like pattern in chosen which the Laser scanner can recognize, and the corresponding output data pattern would look like a saw tooth, as shown in figure 3.

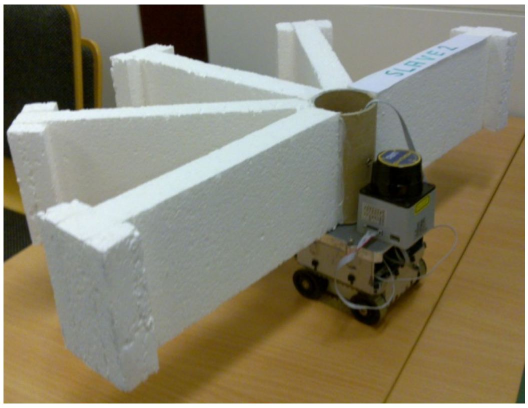

Figure 3: BeBot robot with star-like pattern

Laser-based with IrPDs system
A third technique for implementing the following behavior is also using Laser scanner, but with infrared photo detectors instead of patterns as in the previous method. Initially, as in the first method using ultrasonic, the scanning robot needs also to send a synchronization signal to the target robot right at the beginning of a Laser scan. So, again the fast 433 MHz RF modules where used for that purpose. After sending the fast synch signal, the target robot starts its timer only 20 us after the scanning robot starts its Laser scan. Once the IrPDs are hit on the target robot, it stops the timer and accordingly would accurately know the angle to the scanning robot, which is proportional to the time. After that the target robot sends the angle via Bluetooth to the scanning robot, which can search within a window of 20 degrees for the distance to the target robot. Figure 4 shows both the target robot on the left mounted with IrPD arrays and the scanning robot mounted with the Laser sensor on the right.

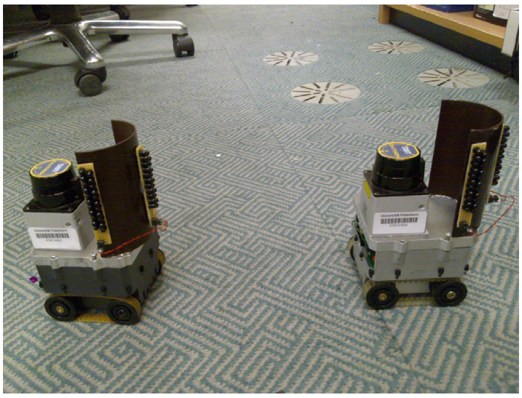

Figure 4: Scanning and target BeBot robots in IrPD method

Comparisons and results

Each of the three techniques has an advantage over the others. The "ultrasonic" method works more robust in smoke since RF and ultrasonic waves are more immune to smoke than laser working in the near infrared range, but ultrasound on the other side is not really robust in indoor scenarios, where delivered data of distance measurements are not stable and needs filtering from spikes resulting from multipath reflections. This will result in increasing the time for one scan to about 1 second for processing, filtering and averaging. Since this technique require simultaneous scans when more robots are used, it will make the following behavior real slow and probably fail due to the very low scan refresh rate. The error in distance measurement after averaging is about 10% and accordingly the error in angle estimation would reach 30%. The third method, which is "laser with IrPD method" also requires simultaneous scanning because the IrPDs cannot differentiate between laser rays coming from multiple sources, but since the Laser scanning time is 100 ms, and the second scanner can start scanning within this time interval right after the first robot is hit and scanned, and since the number of following robots is limited to 5 only in this scenario, the following behavior still works fine, with a scan refresh rate of about 500 to 700 ms. The resolution used for distance measurement is 6 cm and the error in angle estimation is less than 5%, which makes the following behavior run smoothly.

The second method, which is "laser with object recognition method" also run smoothly and has the advantage that all scanners can work simultaneously because the scanning robot does not require any feedback from the target robot. From this one could benefit a very high scan refresh rate of 10 times / second (only 100 ms), which is the maximum scanning rate the laser scanner can provide, and get rid of any limit on the number of following robots. The only problem is the large size of star pattern which makes it hard to move in sharp corners and also could make the scanning robot lose tracking of its target robot. Hence the third method was chosen for our scenario. Yet the performance of the second method can still be enhanced by shrinking the size of the pattern and adding reflectors on the edges.

Routing protocol

The main ad-hoc routing protocols are re-active and pro-active routing. The first one provides better bandwidth usage than pro-active routing, due to the limitation of unnecessary periodic updates, where a route is only searched for and established on demand. This makes the re-active routing more attractive to use in large networks. The main drawbacks of re-active routing is that it is not preferred for usage in high mobility networks or in radio-harsh conditions where links tend to break more often, like in our case of warehouses full of metal and excessive reflections that lead to instability in link quality. Re-active routing is not preferred in such conditions since link failures will trigger searching and establishment of new routes, which will introduce large time delays. Consequently, this will lead to another drawback, which is the unsuitability of re-active routing for real-time continuous data transmission, which is also required in our scenario especially for the continuous data transmission between base station and fire fighter.

According to the mentioned conditions for our scenario, pro-active routing is found to be more suitable. Most of the well-known pro-active routing protocols are originated from two families; distance vector family (DV) and link state family (LS). Link state protocols are more popular nowadays since they are newer and rely on routing based on link state parameters (more stable links) rather than relying on shortest distance only. Three main LS protocols had emerged; GSR (Global State Routing) [2], OLSR (Optimized Link State Routing) [3] and FSR (Fish-eye State Routing) [4]. GSR is the first implemented LS protocol, which is the simplest in implementation but not preferred in large networks. OLSR tends to reduce number of nodes used for message forwarding in the entire network by selecting special nodes called MPRs

(Multipoint Relays), see figure 5. For FSR, instead of reducing the number of nodes used for forwarding messages like OLSR, it reduces the number of forwarded messages themselves, by reducing the update rates for distanced nodes and increasing it for near-placed nodes.

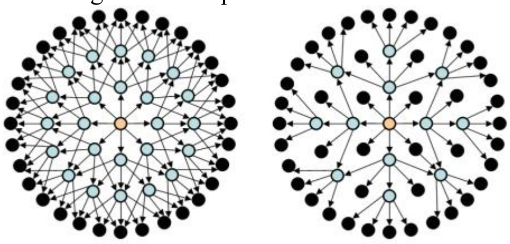

Figure 5: Comparison of link density between message forwarding in GSR (left) and OLSR (right)

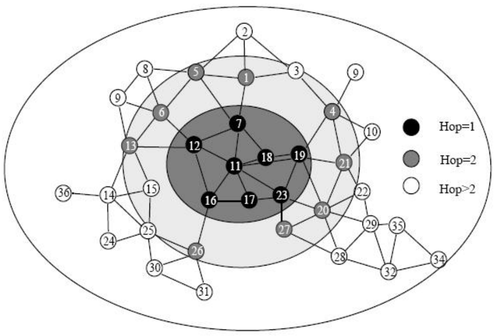

Figure 6: Different update rate zones in FSR

OLSR is the chosen protocol for our scenario. The used software driver is called "OLSR Daemon" which is widely tested and used, and runs under Linux systems supported by our main communication module (gateway) used in the project. Another advantage of using OLSR is that the code can be easily modified to switch to GSR or FSR. GSR is simpler and is thought to be more efficient in our case scenario since our network is mid-sized.

Communication gateway

The chosen routing protocol for mobile ad-hoc communication is implemented on a specially designed hardware. This so called mobile communication gateway is optimized for the mobile usage and therefore supports different techniques for energy saving. Some of these techniques are dynamic frequency and voltage scaling as well as dynamic power down of non-used hardware components. It is equipped with the new OMAP35xx processor, which delivers more than 1,200 Dhrystone MIPS at low power levels. The standard configuration supports the wireless communication standards Wifi and Bluetooth. Based on a modular concept it can be equipped with additional Ethernet or NanoLoc communication. The latter communication module offers distance measurement between wireless network nodes. Additionally the wired communication standards I²C, SPI, UART and high speed USB allows variable expansion of the gateway. Therewith it is possible to connect sensors, actors, robots or computers direct with the gateway and thereby with the communication network.

The gateway is implemented on a 58.4mm by 22.4mm sized 10 layers printed circuit board with about 3000 via holes. The minimum via hole size is 100 µm. The minimum track width and clearance are also 100 µm. The board used the new package on package technology which allows the assembly of the memory chip directly on top of the processor chip.

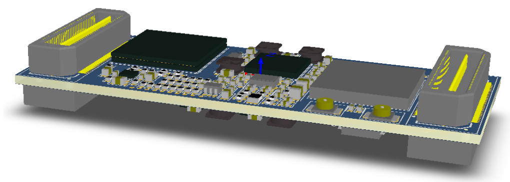

Figure 7: 3D Model of the printed circuit board of the gateway

The software environment of the gateway is a Linux operating system. It consists of adapted Linux kernel, the GNU C standard library and the device manager udev. The standard Unix tools were provided by the software application BusyBox. This combines tiny versions of many common Unix utilities in a single and executable. The software building is done via OpenRobotix [5]. This is an extension of the OpenEmbedded development environment which allows the creation of a fully usable Linux operating system. It generates cross-compiled software packages and images for the embedded target. The existing software branch was extended to contain the hardware special information, patches and additional software like the Player network server and drivers for the hardware.

The network communication is based on the standard internet protocol suite known as TCP/IP. This allows an abstract and common communication over different communication standards. Well-known implementations are Ethernet and IEEE 802.11 wireless LAN (Wifi). But there are several other implementations like the Bluetooth network encapsulation protocol (BNEP) and the USB communications device class (CDC) Ethernet. We implemented an additional driver for our NanoLoc module which enables network communication over IEEE 802.15.4 devices. The driver is based on the serial line internet protocol (SLIP) but expands this with multi point support and ranging functionality. All these implementations together enable network communication via different standards and interfaces.

The underlying routing of messages between the different interfaces and standards is managed by OLSRD [6]. This is a widely used implementation of the optimized link state routing protocol and has an active ongoing development. It discovers possible connections between the gateways, estimates the link quality and configures the routing accordingly.
The underlying IP protocol is configured to use the subnet 192.168.0.0/16. Thereby the subnet 192.168.0.0/24 is reserved for the Wifi interface of the gateways. The last number of the IP represents the number of the gateway. The subnet 192.168.x.0/24, in which the x is replaced by the gateway number, is reserved for the network connected to the USB interface of the gateway. This allows the usage of 254 gateways in one network. The number can be increased by using another not so human readably schema. The usage of different subnets for every gateway allows transparent communication over the mobile ad-hoc network. The setup of a port forwarding or similar configuration is not needed and arbitrary software or services can be used. An additional simplification is permitted by using dynamic host configuration protocol (DHCP) to automatically assign an IP address to the robots or base station connected to the gateway. This IP has the type 192.168.x.2. Again the x is replaced by the gateway number.
The integration of status information into the robot or base station is done via a player driver. This driver will be loaded by the player server on the robot and for example publish a topological map of the network. This map consists of network links between two gateways and the link

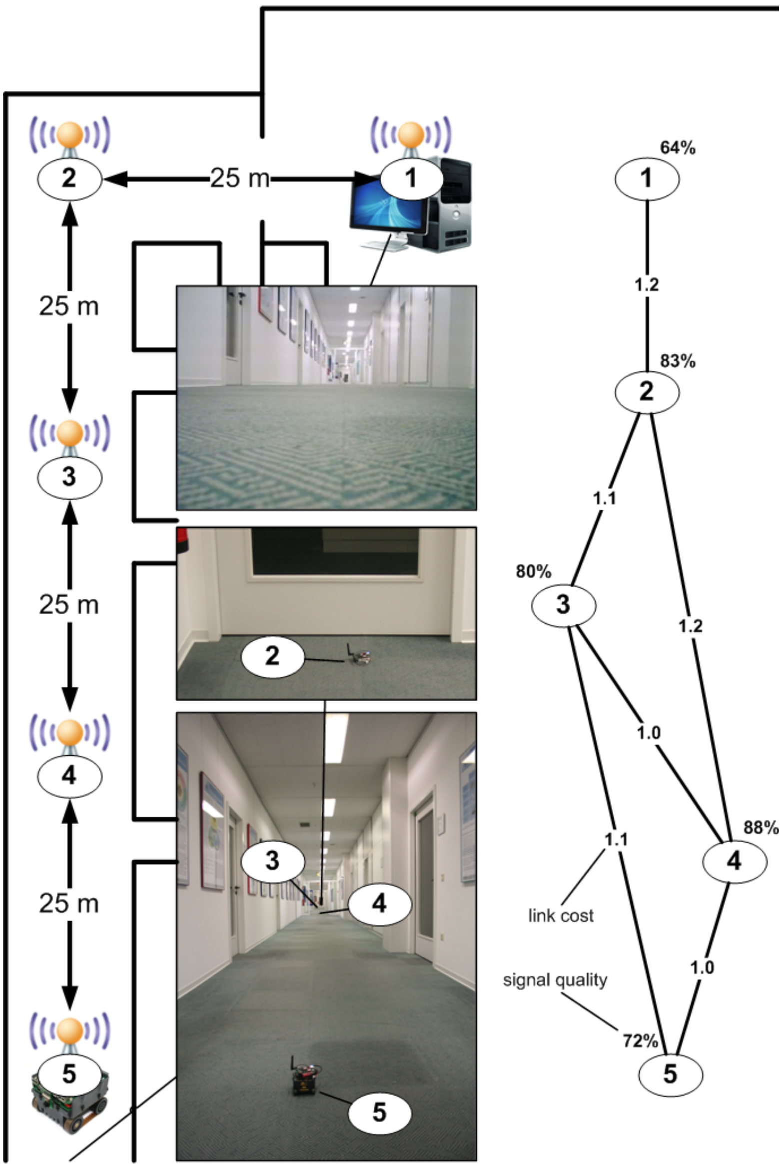

Figure 8: Set-up, images and topological map of test scenario 1 (long corridor in office building)

quality. The integration of the status information via player driver simplifies the usage and allows a remote monitoring of the information without additional knowledge about the gateway.

Tests and results

The communication was tested in two scenarios. Scenario 1 depicted in Figure 8 represents a test in a simple environment and took place in our institute building along a corridor.

The buildup of the test consists of four gateway modules placed along a corridor with equal distances of 25 m between each other. Gateway number 5 at the end of the corridor was connected to a BeBot miniature robot. An additional gateway number 1 was connected to a base station in a free space office next to the other end of the corridor and displays a live camera stream of the BeBot robot shown at top of Figure 8. The topological map on the right side of Figure 8 shows the links between the five gateways. The gateways are connected in a consecutive numbered line. Along the corridor the gateways can also establish good direct connections between their two hop neighbors. The signal quality was much better in the corridor as in the free space office around the base station. The tests were done with a fixed bit rate of 36 Mb/s via orthogonal frequency-division multiplexing (OFDM) modulation and 11 Mb/s via complementary code keying (CCK) modulation. The latter shows a more stable network during a changing environment caused by humans walking along the corridor but has the drawback of a lower throughput.

Scenario 2 depicted in Figure 9 represents a test in a harsh environment with much more metal and takes place in our institute parking garage. The test was done with two different buildups. Version a) used a fixed bit rate of 11 Mb/s via CCK and achieved distances up to 25 m between two gateways. During version b) with a fixed bit rate of 36 Mb/s via OFDM the distance declined down to 10 m. Also the link cost in the topological map, shown on the right side of Figure 9, increased, which stands for more packet losses and poor link quality.

The two scenarios show the tradeoff between the maximum distance between two gateways, bit

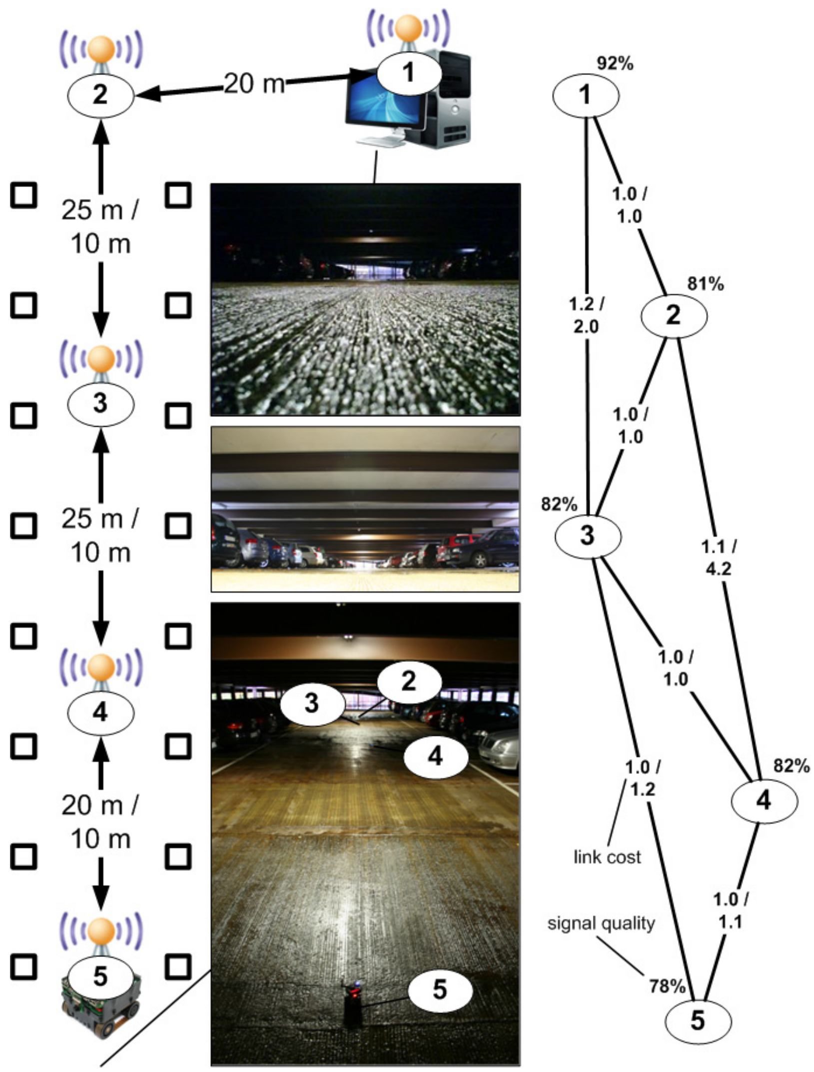

Figure 9: Set-up, images and topological map of test scenario 2 (parking garage)

rate, link cost and characteristic of the environment. By a reduction of the bit rate a distance of 25 m can be reached even in harsh environments. To avoid a general decreasing of the bit rate and thereby of the throughput a dynamic adjustment of the bit rate can be used. The topological maps show direct links between two hop neighbors which means an additional stability and reliability for the network, because of the redundancy.

Conclusion

Main aim of the GUARDIANS project is the development of a team of robots that supports human fire fighters and thereby increases the overall safety and extends the operational area of the fire fighters. One of the key issues in this purpose is the buildup and maintenance of a robust communication system providing communication between all team members. Therefore we have tested and compared different techniques for the distribution of infrastructure nodes. Additionally we have developed a mobile communication gateway which is optimized for the mobile usage and therefore supports different techniques for energy saving. The used network communication is based on the standard internet protocol suite and allows a multi standard communication. We have tested and verified the functionality of the network gateway in different environments like an office or a parking garage.

Acknowledgements

This work was supported by the Sixth Framework Program of the European Union as part of the GUARDIANS project (no. 045269, www.guardians-project.eu).

References

[1] S. Herbrechtsmeier, U. Witkowski, and U. Rückert: BeBot: A Modular Mobile Miniature Robot Platform Supporting Hardware Reconfiguration and Multi-standard Communication. In: Progress in Robotics, vol. 44, 2009.

[2] C. Tsu-Wei, and M. Gerla. Global state routing: a new routing scheme for ad-hoc wireless networks. IEEE International Conference on Communications, ICC 98, 1998; and Global State Routing: http://wiki.uni.lu/secan-lab/Global+State+Routing.html

[3] P. Jacquet, P. Muhlenthaler, A. Qayyum, A. Laouiti, and L. Viennot, T. Clausen. Optimized Link State Routing Protocol (OLSR), RFC 3626. http://www.olsr.net/

[4] G. Pei, M. Gerla, and T.-W. Chen. Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks. Proceedings of ICC 2000, New Orleans, LA, 2000

[5] OpenRobotix: Openembedded based open source linux distribution for mini robots: http://openrobotix.berlios.de/

[6] OLSRD – An adhoc wireless mesh routing daemon: http://www.olsr.org/