Electronic Communications of the EASST
Volume 37 (2011)

Workshops der wissenschaftlichen Konferenz Kommunikation in Verteilten Systemen 2011 (WowKiVS 2011)

RoombaNet - Testbed for Mobile Networks

Mohamed A. Hail, Jan Pinkowski, Torsten Teubler, Maick Danckwardt, Dennis Pfisterer, Horst Hellbrück

12 pages

Guest Editors: Horst Hellbrück, Norbert Luttenberger, Volker Turau
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

RoombaNet - Testbed for Mobile Networks

Mohamed A. Hail[1], Jan Pinkowski[1], Torsten Teubler[1], Maick Danckwardt[2], Dennis Pfisterer[2], Horst Hellbrück[1]

Lübeck University of Applied Sciences[1], University of Lübeck/Institute of Telematics[2]

Abstract: The design and deployment of wireless networks needs careful planning including various tools for analysis, simulation and evaluation. Therefore, development of software to support deployment of wireless networks has been subject of intensive research for several years. In particular the evaluation of the influence of mobility remains a challenging task. For deployment of mobile communication networks operators perform simulations and measurements during the planning process with large efforts. In the past the research community based their decisions on development of new protocols on simulations exclusively. While network simulators provide fast investigation of huge and also mobile networks they rely on theoretical models which are often considered as inaccurate and too optimistic. Therefore, more and more real wireless network environments called testbeds are established worldwide most of them with static nodes. Testbeds dedicated towards mobile networks remain a challenge as the effort to build such a network increases with mobility. The work here presents an approach for a fully automated real-world mobile network testbed where nodes are piggybacked on mobile robots. The platform with up to 30 mobile nodes and additional 30 static nodes can emulate various scenarios especially suited for pedestrian scenarios or for slow car movements. In this paper we introduce this testbed which is integrated into the larger Real-World GLab Internet testbed facility. We provide first details of the hardware and software components and provide first evaluations as well as present application examples.

Keywords: Mobile Wireless Networks, Ad-Hoc networks, Testbed, Automated Test

1 Introduction

The Internet as we know it today will change significantly in the next years with new devices and applications that enrich and enlarge the network. In the year 2017 the survey of the wireless world research forum [Dav08] expects seven trillion of wireless devices. A significant part of these devices will be sensor nodes besides the already well known wireless helpers of humans like laptops, tablets or smart phones. Cars will be equipped with wireless devices and form mobile networks so called Vehicular Ad-Hoc networks (VANETs) within the future Internet. Consequently, we expect to see also mobile sensor networks that will form multi-hop Ad-Hoc networks. Therefore, one of the future challenges for the established Internet technologies are the integration of Ad-Hoc networks like mobile wireless sensor networks or mobile Ad-Hoc networks (MANETs) that consist of a large number of devices with very specific constraints. Here, Ad-Hoc communication means multi hop communication among wireless devices which

is a challenging task with respect to devices limitations. However, multi-hop is necessary to transfer data over a larger distance.

In the past, multi-hop Ad-Hoc networks were investigated mainly by simulation. However these simulations hardly rely on abstract models and results are only as accurate as these underlying models. This problem is addressed by the Future Internet research community and testbeds for Future Internet activities are deployed like (GLab [BFF+10], Wisebed [BCD+10]). However, these testbeds lack support for mobility.

Our work introduces a testbed based on off the shelf iRobot Roomba [iC10a] housekeeping robot that piggybacks devices with wireless interfaces like WLAN enabled PC based system and sensor nodes. Roomba empowers a simple and effective mobility approach for wireless nodes to form mobile Ad-Hoc Networks. The robots are easy to control including a built-in default mode to return to their charging unit. They move independently and are fully controllable in an efficient way via a serial interface.

Future plans foresee an integration of RoombaNet in RealWorld-GLab. Since RoombaNet is in a quite early stage of development and topics like deployment or possible scheduling of timeslices to the remote users remain to be clarified. The maximum runtime of a test is limited by the battery charge of the robots and will be between 1 to 1.5 hours. We target to offer a testbed with a good availability although our mobile nodes need to be recharged after a test run.

The rest of the paper is structured as follows: In Section 2 we discuss related work. Section 3 introduces our approach and provides more details of the testbed implementation including two mobility models that we implemented already. In the evaluation section we will provide first results and demonstrate the first application. The paper will conclude with a short summary and a description of the next steps to complete our testbed.

2 Related Work

The idea to use iRobot Roombas as a mobile platform is not completely new. There are existing works with background in robotics as well as WSN applications [RMR08].

The work mentioned in [RBS10] describes a middleware approach for networked robots. This work is different from our project because it targets the application level and the interaction between objects. The described testbed seems to be highly complex and uses advanced robots with a high degree of freedom. However, building such a middleware on top of our mobile WSN platform is out of the scope of this project.

An interesting approach for our ongoing research is described in [SD10]. The authors suggest a mobile base station that collects data from static sensor nodes and introduce a suitable data collection protocol. For evaluation purpose a Roomba robot serves as mobile base station. With this approach one can eliminate complex routing mechanisms but it is questionable if this procedure can applied to every real world scenario. In our opinion arbitrary nodes should be considered as mobile in future scenarios and a flexible testbed needs to cover this issue.

Recent work to mobile Ad-Hoc networking testbeds [BIXD10] investigates throughput, packet loss and hidden terminal problem in a mobile Ad-Hoc network. During evaluation nodes are moved manually placed on office chairs.

A Table of testbed projects worldwide can be found in [Coo10]. The listed mobile testbeds

use human volunteers, cars or transit buses mainly. One testbed MiNT [DRSC05] also uses Roomba as a mobile platform. 802.11b serves as wireless technology for the test and a non-interfering technology for controlling the robots (wired connection or infrared). In our testbed we use 802.15.4 enabled sensor nodes which is proposed as the wireless networking standard in WSN whereas the control channel is WLAN and vice versa. The operating channels of 802.15.4 will be chosen appropriately to avoid interference with the WLAN.

At Skymesh [SKM+06] blimps form an Ad-Hoc network and serve as a mobile access point for Internet connectivity. However, this complex outdoor assembly is not suitable for use as a frequently available mobile testbed like our RoombaNet.

Another related field of research is localization within mobile WSNs. As stated in [AK09] WSNs have to be aware of their location for context awareness and navigation. In our testbed nodes are passive with respect to their movement so issues of navigation are a matter of the underlying mobility model and will be considered there. The spatial context of the collected sensor data is also a matter of the application and thereby out of the scope of this work in the first step where we foresee tests for routing and data dissemination protocols.

[CFK+10] describes Roombas as carriers for mobile sensor nodes. The work provides neither technical details nor evaluation of the mobile WSN. In [RHLG10] the authors describe a testbed with mobile nodes that make use of the LEGO Mindstorms platform to carry a sensor node within a stationary part of the sensor network. To enable reproducibility of experiments on the testbed the LEGO Mindstorms robot follows a track using a light sensor and the sensor node send out beacon messages to locate the nodes position on the track with the RSSI values of received messages.

Based on the experiences from the project AutoNomos (http://www.auto-nomos.de) Lego Mindstorms as well as the other related work does not fulfill the reliability requirements of a fully automated testbed where we perform remote protocol evaluation without local human interaction.

3 RoombaNet

In this section we describe our automated testbed for mobile networks. The implementation and deployment of RoombaNet is not yet finished. We start to explain the approach following the steps when using the testbed. We begin with a basic overview of the functionalities followed by a description of the main parts of our testbed. Afterward the rest of the section describes the mobility aspects of RoombaNet.

3.1 Testbed Overview

When the testbed user has finished the scenario generation, the configuration will be transferred from the testbed server to the mobile controllers of the nodes before the test run as shown in Figure 1 and Figure 2.

In the scenario manager general test parameters, the mobility, the software, network traffic and logging of data can be configured and sent to the mobile node via WLAN. Basic configuration parameters like number of nodes, start and stop time are first included in the scenario description.

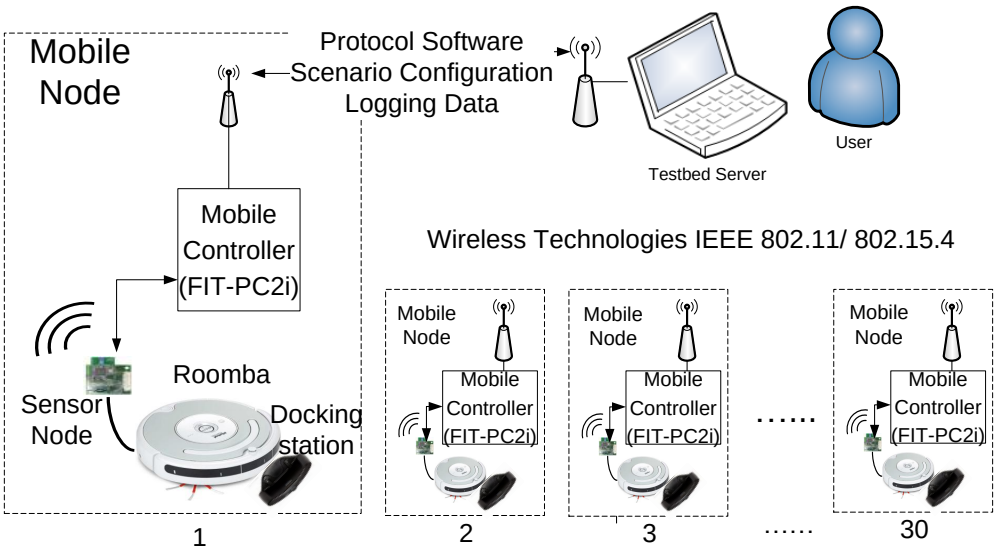


Figure 1: Testbed Overview

There are several options for a mobility configuration. The scenario can include mobility traces or configuration for autonomous movement according to mobility models as we will describe in Section 3.4. Important parameters like average speed and movement time for mobility models can be generated using random processes on the mobile nodes directly.

One major advantage compared to previous work is that the sensor node can be fully programmed and controlled via the FIT-PC2i. Test program, driver and firmware can be transferred to the mobile node and are controlled from the software manager and client software manager. The firmware programmer is integrated in the mobile controller FIT-PC and programs the firmware on the sensor node via USB/JTAG interface.

As a mobile controller we selected a FIT-PC2i which is a tiny (27mm x 115mm x 101mm) low power Intel Atom PC running windows 7 or Linux operation system Placed on top of the Roomba the PC controls the evaluation of the mobile networks. In the second step one or several test runs are started after having received the configuration, protocol, firmware and application software via WLAN. The test run starts with a message from the testbed server. Figure 5 illustrates a test run on a floor at our institute. The protocol software will be executed on one of the wireless devices. The wireless device can either be the FIT-PC itself or a connected sensor node see (Figure 1).

During a test run we collect and store data from the wireless device to the FIT-PC for logging and enabling subsequent analysis and evaluation of the test results. The data collected from the sensor nodes is stored on the FIT-PC. Finally, log data is received and processed via logging manager.

At the end of the test run the Roombas autonomously return to their docking station to recharge themselves automatically. This enables them to be ready for the next test run while the logged data will be collected from the FIT-PC2i and processed by the server.

The implementation of the software components has been started in Java to allow switching

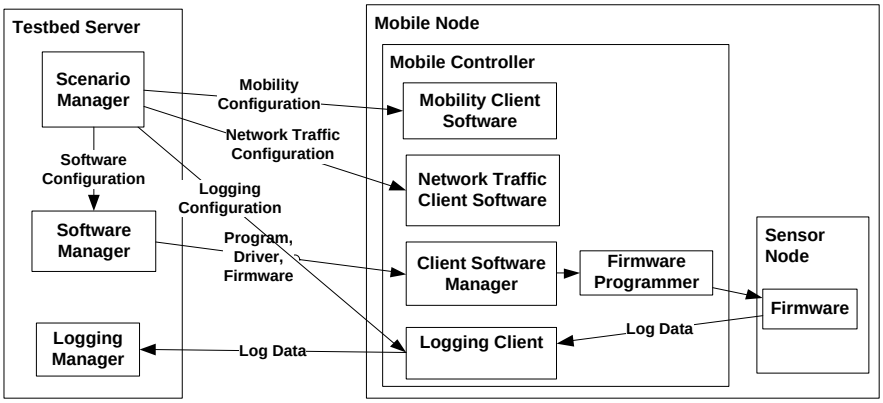


Figure 2: Interaction of Functional Blocks (Software)

the operating system in a flexible way.

3.2 HW Components of the Testbed

In this part we describe the main hardware components of RoombaNet. We start with the Roomba which is a mobile vacuum cleaner robot from the company ”iRobot” with a diameter of 34 cm, a height of 9.4 cm and a maximum driving speed of maximum 500 mm/s (1.8km/h). Various sensors on the robot like wheel-drop, bumper or infrared sensor detect obstacles and prevent it from having accidents like falling down the stairs. With the help of its infrared receiver and sophisticated algorithm, the Roomba is able to return to its docking station. The Roomba provides control and manipulation by using an external device via an Open Interface (OI). A PC or microcontroller can act as a controller by sending commands via a RS232 to the Roomba. The 7-pin mini-DIN connector of the Roomba provides pins for the power supply, switching the baud rate of the RS-232 and TXD, RXD lines. Detailed specification of the interface and the commands is available at [iC10b].

Our first prototype consists of a sensor node called TriSOS which is placed on top of the Roomba, as illustrated in Figure 3 and Figure 4. With the mobile controller FIT-PC we have an additional 802.11g interface for test runs as illustrated in Figure 3.

3.3 Basic Movement API for Mobility Models

The Roomba OI provides functionalities including simple drive commands and request commands for sensor values. To compose the correct byte sequence specific knowledge of the Roomba is needed for the command codes, their special parameters and restrictions. Another challenge is that the Roomba does not stop by itself during a motion. Once a driving command is executed it will drive continuously if not stopped manually. So the direction and distance is controlled by Roomba’s odometry data. In summary, there is no simple way to move the robot in a human intuitive way like ”turn 90° to the right”. Therefore, we implemented a software

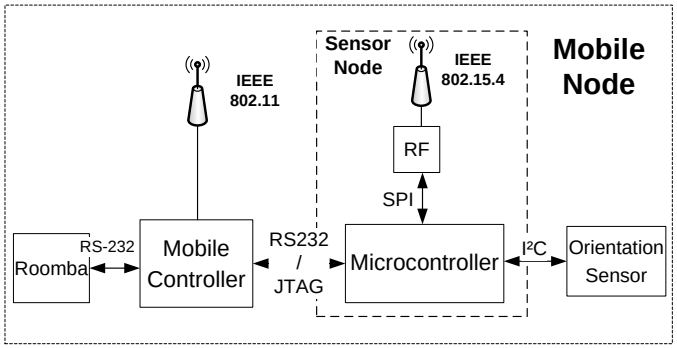


Figure 3: Block Diagram of Mobile Node

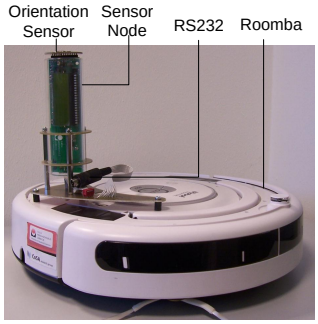


Figure 4: Roomba with mounted Sensor Node

driver that encapsulates these specific Roomba details from the user providing the following new interface:

```
move(uint16_t distance, uint16_t speed);
turn(int16_t angle, uint16_t angular_speed);
```

- *distance* in mm the Roomba will drive
- *speed* of the drive wheels in millimeter per second (mm/s)
- *angle* degree the Roomba will turn
- *angular_speed* the Roomba will turn in degree per second

The distance is specified in mm which determines how far the Roomba will move. The direction is given as an angle between -360° and 360° where negative values will result in a left-turn by the Roomba. This interface is more intuitive and matches the human expectations for controlling in a better way compared to the basic Roomba Open Interface (OI) specification. Additionally, we can build complex movements from these basic movement patterns. This is shown in the next part of this paper.

3.4 Basic Available Mobility Models

The movement of the mobile nodes plays a central role in the planning of the mobile network and the evaluation of protocols and applications of Ad-Hoc networks. In this section we introduce two basic mobility models that are already implemented in RoombaNet.

Random-Direction (RD)

This model generates a random unpredictable motion of the network nodes. In the basic variant the RD nodes move with a constant speed from a starting point in random direction α for a period t [ZD97]. The direction α of the nodes is uniformly distributed between 0° and 360° (right-turn) or between 0° and −360° (left-turn) by a random process for each new motion step.

Gaussian-Markov-Model (GM)
The Gaussian-Markov-Model has been introduced for the evaluation of prognosis-algorithm for the position of any subscriber in mobile radio network [LH99]. The difference between the GM and the RD is that motion steps in GM are smaller and the newly selected speed and direction depends on the previously selected values. In our algorithm the speed is fixed and the direction is calculated according to the formula: $\alpha_n = k\alpha_{\text{n-1}} + (1-k)\mu + \sqrt{1-k^2}\alpha_{\text{xn}}$, where $1 \leq \text{k} \leq 0$, μ is the average value of α (normally equal 0), and x_n is a Gaussian random process with $\mu = 0$ and standard deviation σ_x.

Decreasing k results in a more random movement of the nodes. If k = 0, the direction of the motion does not depend on the past α and the resulting behavior is equivalent to the RD model. If k approaches the value 1, then α changes very slowly.
The two above mobility models are implemented in our testbed and evaluated in the following Section 4.1.

Figure 5: RoombaNet Test Run

4 Testbed Evaluation

In this section we provide the first preliminary evaluation results for our testbed. We start with a basic analysis of which scenarios can be tested with RoombaNet and provide evaluation of the mobility of the nodes. We will conclude the section with an exemplary application in Section 4.3.

The idea behind RoombaNet is comparable to engineers that build a reduced size model of their complex system. They start with analysis and simulations. Afterwards they build a smaller model for the first test runs until they finally build the first prototype. Even if it is not 100% functional it can still provide important results that lead to efficient improvement on the final product. RoombaNet allows for fast and cheap test runs before testing on the final version of protocols for mobile networks in the same way.

Our testbed can be used on realistic simulations of these network protocols. We start our assessment of possible scenarios with the speed and dimensions of the Roomba in Table 1 as

well as the transmission range that we can adjust.

Scenarios	Size [cm]	Speed [km/h]	Transmission Range [m]	Scaling Factor
Roomba	34 x 34	0-1.8	ca. 30-300 (802.15.4/802.11)	
Pedestrian	ca. 100 x 100	ca. 3-6	ca. 30-300 (802.15.4/802.11)	1:3
Car	ca. 300 x 180	ca. 30-100	ca. 300-1000 m (802.11p)	1:8

Table 1: Roomba Down Scaling

Due to the maximum speed of 1.8 km/h and physical size, the Roomba would be most suitable for a pedestrian scenario when scaling by 1:3 as we will show here. An average adult pedestrian needs about 1m x 1m space on the ground when walking; height doesn't matter. A typical speed for pedestrians is between 3-6 km/h. The scaling factor of 1:3 for Roomba to pedestrians fits quite well with a scaled resulting speed of maximum 5.4km/h. Even the transmission range of our nodes fits into this scenario when reducing the transmission power. One unresolved issue in our first implementation for pedestrian scenarios is how to avoid collisions like humans can do. For robots it is not that easy. There are basically four ways of detecting an obstacle by touching. A robot can use acoustical sensors like ultrasonic, optical sensors like infrared or image recognition.

Although the shape of cars is fundamentally different to that of the Roombas, we predict good chances to using RoombaNet for beginning evaluations of VANETs. The scaling factor here is about 1:8 which results in a scaled maximum speed of 8x1.8km/h=15km/h. In the case where we have a mismatch we can "slow down" the time base of our test runs by a scaling factor of 2-7 to emulate the typical speed of 30-100km/h for city or freeway scenarios. A typical range in the current IEEE 802.11p standard is about 300-1000m. In our testbed as introduced in Section 3.1, we can choose 802.11g or 802.15.4, wireless technology could be used and we adjust the transmission range on a large scale interval.

4.1 Basic Mobility Models

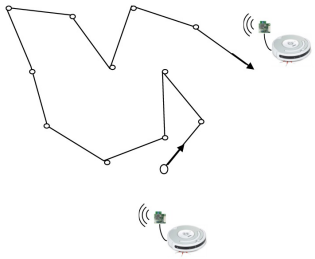
Two mobility models namely Random-Direction (RD) and Gauss-Markov (GM), as described in Section 3.4 are already implemented in our testbed and demonstrate basic mobility. Figure 6 shows a typical movement of the random direction model. A random process provides values for the direction α. The other parameters velocity v and time t_{mov} are fixed in this scenario.

In Figure 6 a Roomba randomly rotates between -360° and 360° (all directions).

Figure 7 shows the motion of the Gaussian-Markov-Model resulting in a smooth path compared to the RD model.

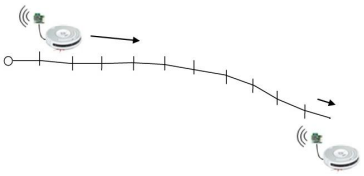
4.2 RoombaNet Positioning Accuracy

Since the positioning and especially the turning movement of the Roombas is inaccurate and changes with the surface like carpets or ceramics we apply an additional orientation sensor to improve the positioning of the nodes and thereby improve the accuracy of the nodes movement.



	α
1	-049
2	-079
3	-142
3	-289
4	060
5	046
6	-013
7	118
8	052
9	-121
10	-271
11	023

Figure 6: Random Direction Model



	α
1	000
2	004
3	-004
3	-001
4	002
5	005
6	000
7	007
8	006
9	-005
10	-003
11	004

Figure 7: Gaussian-Markov-Model

For RoombaNet we decided to use the low power sensor AMS0805WAH. The orientation sensor is connected to our sensor node and can be read by the mobile controller using RS232 connection. For RoombaNet the important value of the orientation sensor is the azimuth value which is the horizontal difference between the North Pole with a range between 0° and 360°. The sensor provides us with the orientation of a Roomba within the scenario which is a very helpful value.

First evaluation results promise a substantial improvement of the positioning with this additional sensor. However, node position might still drift away, so that absolute positioning is impossible with long test runs. We need to quantify this effect but with a test run less than 2 hours we are confident that we can cope with this deficiency. Some of the mobility models do not need absolute positioning like Gaussian-Markov.

4.3 Exemplary Application

One useful case for a mobile testbed is the evaluation of swarm logic, a part of the organic computing. Swarm logic analyzes the swarming behavior, e.g. of insects, and tries to adopt these motion patterns to mobile systems like mobile sensor networks. Example patterns for such a swarming behavior are:

- gather - all nodes move to one central point and group as close as possible.
- spread - the nodes move away from each other but stay in communication range.
- random walk - move around randomly but stay in the testbed area.
- synchronized moving - all nodes move around in a synchronized way and hold the distances between all neighbors.
- move home - the nodes move to a central point e.g. to a home base.

These motion patterns are useful for self-organizing applications like discovering an unknown area, optimizing a monitoring area or automatically recharging mobile nodes. In addition to the basic mobility models swarm or bio-inspired mobility models or protocols like Phero-Trail [VLG08] can be realized. This localization and routing protocol is designed to operate in a synchronized moving sensor network with some nodes circulating in the group.

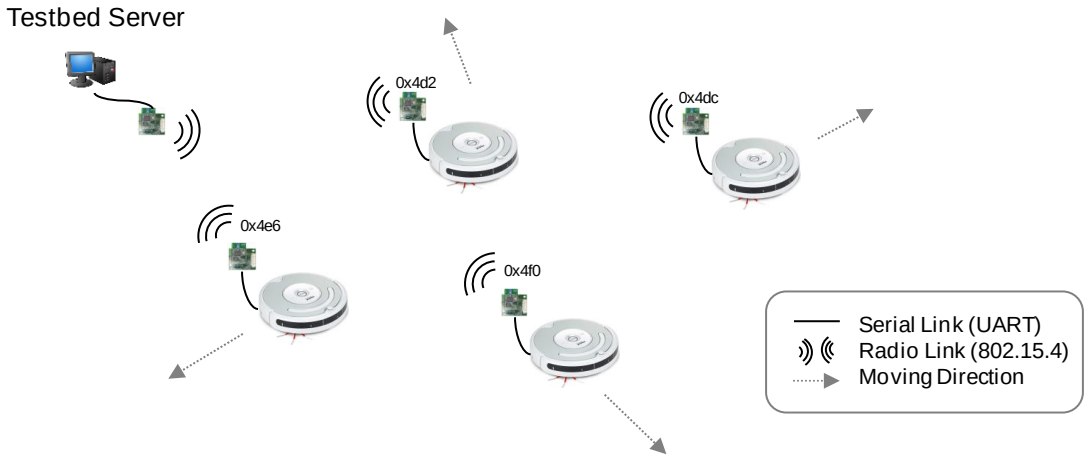


Figure 8: Sample movement - spread

The swarm logic has two fields of activity that can be evaluated in a mobile testbed as we describe it here. The first one is the design of the basic motion patterns like ”spread” to enable the nodes to react as a swarm. The second field is to use these patterns and evaluate specialized protocols for such moving sensor networks.

We have implemented the two motion patterns ”gather” and ”spread” as a proof of concept for the RoombaNet. On the other hand we developed a Roomba control server application to navigate Roomba robots from a central point. With this software it is possible to navigate a single robot or a set. It allows the user to see all necessary information like battery status or sensor values from the Roomba. In Figure 8 each sensor node is able to control its own robot and to read the sensors using the serial link control protocol as described above. The Roomba control application uses the wireless sensor network to forward these messages to the corresponding node and to inject the command or sensor data request to the Roomba robot to form the spread operation.

5 Conclusion and Future Work

In this paper we described the testbed RoombaNet designed for fully automated evaluation of mobile network protocols. We presented our testbed following the steps of a test run, starting from the configuration at the testbed server, the description of the FIT-PC equipped with 802.11g to receive the protocols and the movement configuration. The FIT-PC moves the Roomba according to a mobility model like Random Direction or Gaussian-Markov-Model. An additional orientation sensor will help to improve the positioning accuracy significantly in future. Mobility models, connection between mobile controller, Roomba and sensor node as well as orientation sensor are implemented and tested. Furthermore a programming device to program the software is integrated between the FIT-PC2i and sensor node and communicated via USB. The testbed

can be used for pedestrian scenarios and for evaluation of car to car communication as shown in the general evaluation of the testbed.

In the next step we will finalize our implementation of testbed server (scenario manager, logging manager software manager) and mobile node (mobility client software, client software manager and logging client). Additionally, we will integrate RoombaNet into the larger GLab testbed. RoombaNet within GLab will be available for prospective protocol designers that can rent this testbed for their own evaluation of new protocols. In the future, we will implement and evaluate mobile network protocols in different environments in this testbed.

Finally, we will extend the choice of mobility models and patterns to achieve a more human like behavior when two Roombas head towards each other and implement more specific setups, e.g. traffic light scenarios.

Acknowledgements: This work was funded by the Federal Ministry of Education & Research of the Federal Republic of Germany (Förderkennzeichen 01BK0905, GLab) and (Förderkennzeichen 17PNT017, DataCast).

Bibliography

[AK09] I. Amundson, X. Koutsoukos. A Survey on Localization for Mobile Wireless Sensor Networks. In Fuller and Koutsoukos (eds.), *Mobile Entity Localization and Tracking in GPS-less Environnments*. Lecture Notes in Computer Science 5801, pp. 235–254. Springer Berlin / Heidelberg, 2009.

[BCD+10] T. Baumgartner, I. Chatzigiannakis, M. Danckwardt, C. Koninis, A. Krller, G. Mylonas, D. Pfisterer, B. Porter. Virtualising Testbeds to Support Large-Scale Reconfigurable Experimental Facilities. In Silva et al. (eds.), *Wireless Sensor Networks*. Lecture Notes in Computer Science 5970, pp. 210–223. Springer Berlin / Heidelberg, 2010.

[BFF+10] D. Bimschas, S. Fekete, S. Fischer, H. Hellbrück, A. Kröller, R. Mietz, M. Pagel, D. Pfisterer, K. Römer, T. Teubler. Poster Abstract: Real-World G-Lab: Integrating Wireless Sensor Networks with the Future Internet. In *TridentCom 2010: The 6th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*. Berlin, Germany, may 2010.

[BIXD10] L. Barolli, M. Ikeda, F. Xhafa, A. Durresi. A Testbed for MANET: Implementation, Experiences and Learned Lessons. *IEEE Systems Journal*, 2010.

[CFK+10] I. Chatzigiannakis, S. Fischer, C. Koninis, G. Mylonas, D. Pfisterer. WISEBED: An Open Large-Scale Wireless Sensor Network Testbed. In Akan et al. (eds.), *Sensor Applications, Experimentation, and Logistics*. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 29, pp. 68–87. Springer Berlin Heidelberg, 2010.

[Coo10] Cooperating Objects Network Of Excellence. http://www.cooperating-objects.eu/testbed-simulation/testbed-federation/testbed-directory/. Website, Juli 2010.

[iC10a] iRobot Corporation. http://www.irobot.com/. Website, October 2010.

[iC10b] iRobot Corporation. iRobot Roomba 500 Open Interface (OI) Specification. November 2010.

[Dav08] K. David. *Technologies for the Wireless Future: Wireless World Research Forum, Volume 3*. Wiley Publishing, 2008.

[DRSC05] P. De, A. Raniwala, S. Sharma, T. cker Chiueh. Mint: A miniaturized network testbed for mobile wireless research. In *In The 24 th Annual Joint Conference of the IEEE Computer and Communications Societies*. Pp. 2731–2742. 2005.

[LH99] B. Liang, Z. J. Haas. Predictive Distance-Based Mobility Management for PCS Networks. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. Pp. 1377–1384. 1999.

[RBS10] J. Rashid, M. Broxvall, A. Saffiotti. Extending a Networked Robot System with Tiny Devices and Everyday Objects. *Workshop on Information Technology, Skovde, Sweden*, 2010.

[RHLG10] O. Rensfelt, F. Hermans, L.-A. Larzon, P. Gunningberg. Sensei-uu: a relocatable sensor network testbed. In *WiNTECH '10: Proceedings of the fifth ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*. Pp. 63–70. ACM, New York, NY, USA, 2010.

[RMR08] J. Reich, V. Misra, D. Rubenstein. Roomba MADNeT: a mobile ad-hoc delay tolerant network testbed. *SIGMOBILE Mob. Comput. Commun. Rev.* 12(1):68–70, 2008.

[SD10] O. Soysal, M. Demirbas. Data Spider: A Resilient Mobile Basestation Protocol for Efficient Data Collection in Wireless Sensor Networks. In Rajaraman et al. (eds.), *Distributed Computing in Sensor Systems*. Lecture Notes in Computer Science 6131, pp. 393–408. Springer Berlin / Heidelberg, 2010.

[SKM+06] H. Suzuki, Y. Kaneko, K. Mase, S. Yamazaki, H. Makino. An Ad Hoc Network in the Sky, SKYMESH, for Large-Scale Disaster Recovery. In *VTC Fall*. Pp. 1–5. 2006.

[VLG08] L. F. M. Vieira, U. Lee, M. Gerla. Phero-Trail: a bio-inspired location service for mobile underwater sensor networks. In *WuWNeT '08: Proceedings of the third ACM international workshop on Underwater Networks*. Pp. 43–50. ACM, New York, NY, USA, 2008.

[ZD97] M. M. Zonoozi, P. Dassanayake. User Mobility Modeling and Characterization of Mobility Patterns. *IEEE Journal on Selected Areas in communications* 15(7):1239–1252, September 1997.