2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns

Parameter Optimization of an Active Tags Based Pedestrian Tracking System on a Distributed Emulator

Junya NAKATA*‡1 Razvan Beuran*‡2 Tetsuya Kawakami†3 Takashi Okada‡*4 Ken-ichi Chinen‡*5 Yasuo Tan‡*6 Yoichi Shinoda‡*7
* *Hokuriku Research Center, National Institute of Information and Communications Technology*
2-12 Asahidai, Nomi, Ishikawa Japan
[1]*jnakata@nict.go.jp* [2]*razvan@nict.go.jp*
† *Panasonic System Solutions Company, Panasonic Corporation.*
4-3-1 Tsunashima-higashi, Kohoku, Yokohama, Kanagawa Japan
[3]*kawakami.tetsu@jp.panasonic.com*
‡ *Internet Research Center, Japan Advanced Institute of Science and Technology*
1-1 Asahidai, Nomi, Ishikawa Japan
[4]*tk-okada@jaist.ac.jp* [5]*k-chinen@jaist.ac.jp* [6]*ytan@jaist.ac.jp* [7]*shinoda@jaist.ac.jp*

***Abstract*—In this paper, we introduce a distributed emulator for a pedestrian tracking system using active tags that is currently being developed by the authors. The emulator works on StarBED, which is a network testbed consisting of hundreds of PCs connected to each other by Ethernet. The three major components of the emulator (the processor emulator of the active tag micro-controller, RUNE, and QOMET) are all implemented on StarBED. We present the structure of the emulator, how it functions and the results from the emulation of the pedestrian tracking system. We confirmed that the emulator is not only useful for the evaluation phase of the development, but also applicable for the parameter optimization of the pedestrian tracking system by making emulations with different parameters.**

***Keywords*-ubiquitous networks; active tag; radio emulation; distributed testbed; supporting software**

I. INTRODUCTION

As Panasonic Corporation (hereafter referred to as Panasonic) is developing a pedestrian tracking system using active tags, one requirement is to carry out a large number of experiments. Real-world experiments with wireless network systems, and active tags in particular, are difficult to perform when the number of nodes involved is larger than a few devices. Problems such as battery life or undesired interferences often influence experimental results. We are currently implementing a solution by developing an emulation system for active tag applications that runs the real active tag firmware within a virtual, emulated environment. Through emulation, much of the uncertainties and irregularities of large real-world experiments are placed under control. In the same time, using the real active tag firmware in experiments enables us to evaluate exactly the same program that will be deployed on the real active tags; this is a significant advantage compared to simulation. For performing the practical emulations we use StarBED, a network experiment testbed. In order to be able to use this testbed for active tag emulation we developed several subsystems, and integrated them with the existing testbed infrastructure. These subsystems were developed on the basis of existing tools that are already used on StarBED, namely the wireless network emulator QOMET [1], and the emulation support software RUNE [2].

Active tags were so far mainly studied through simulation, such as the work presented in [3]. Public domain wireless communication emulation research is currently mainly done in relation to Wireless LANs (WLANs). One can use real equipment, and hence be subject to potential undesired interferences. Two examples from this class that allow a controlled movement of wireless nodes are the dense-grid approach of ORBIT [4], or the more realistic robot-based Mobile Emulab [5]. An alternative which avoids undesired interferences and side effects is to use computer models for real-time experiments. TWINE [6] is an example from this class. TWINE is a wireless emulator that combines wireless network emulation and simulation in one setup, but only supports 802.11b WLAN so far. Our development started from an existing wireless emulator, QOMET, which uses similar concepts.

There are already a number of implementations of experiment tools for ubiquitous systems that could be used in conjunction with active tag devices. Some of these tools focus on the operating system level, such as TOSSIM [7], which is a TinyOS simulator aiming to simulate TinyOS applications accurately in a virtual environment. ATEMU [8] is able to emulate TinyOS applications at processor level; its flexible architecture has support for other platforms too. ATEMU is thus closer to our purpose, since our low-cost active tags do not use any operating system. We aimed to run in emulations the same firmware with the one used by the real devices. The manufacturer of the active tag processor, Microchip, only provides two alternatives for system development: real-time emulation in hardware

978-0-7695-3862-4/09 $26.00 © 2009 IEEE
DOI 10.1109/ComputationWorld.2009.39

using either the MPLAB REAL ICE In-Circuit Emulator, or the PICMASTER Emulator, or processor simulation using the MPLAB-SIM Simulator [9]. However none of these solutions are appropriate for our purpose; thus we developed our own cycle-accurate processor emulator running on PCs.

The pedestrian tracking system developed by Panasonic makes use of active tags so as to provide to a central pedestrian localization engine the information needed to automatically calculate the trajectory to date and the current position of the active tag wearer. Using the prototype of the pedestrian localization system, real-world experiments were carried out in March 2007, as reported in [10]. The experiment consisted in the orchestrated movement of 16 pedestrians both in indoor and outdoor environments. A system overview and experimental conditions will be presented later in this paper.

One of the important conclusions of the experiment was that it is very difficult to organize a real-world experiment for such applications of active tags. The number of people involved, and the accuracy of their movement following the predefined scenario, are only a few of the issues encountered. Nevertheless, the results of the above-mentioned experiment are currently being used as a basis for improving the prototype of the pedestrian localization system and extending it for use with very large groups of people, of the order of one thousand. The active tag emulation system that we designed and implemented plays an essential role at this point, since it makes it possible to continue the experiments in the development phase with ease and in a wide range of controllable conditions.

Based on the real-world experiment, we carried out the emulation to validate the emulator can reproduce the real-world experiment with enough accuracy. As a consequence of the evaluation of the results obtained by executing the emulation, it is confirmed that the emulator can re-create the real-world experiment and give useful data for the system development [11]. In this paper, several emulations with different system parameters are carried out to fit the system parameters to the condition in which the system is utilized.

The paper is organized as follows. Section II introduces our general approach to active tag emulation. This is followed by Sections III and IV, which describe the main components of the active tag emulation system: the wireless communication emulation, and the active tag processor emulation subsystems, respectively. Section V presents the preliminary real-world experiments carried out in order to validate the pedestrian localization system prototype. Section VI gives the details of the emulations executed. Section VII is dedicated to presenting emulation results obtained with the emulation system in the attempt to reproduce and extend the preliminary real-world experiments. The paper ends with sections on conclusions and references.

II. SYSTEM DESCRIPTION

The technique of emulation implies creating a virtual environment in which the movement, the communication, and the behavior of active tags are all reproduced. Emulation has two main requirements in the case of our project: (i) Emulate the wireless communication of the active tags; (ii) Emulate the active tag processor so that the same firmware used by the real devices can be tested in emulations. The conclusions of the real-world experiment using the pedestrian localization prototype system were used as guidance during the design and implementation of the emulation testbed. In addition, we used our previous experience with emulation systems, such as those presented in [12] and [13], as we built the wireless communication emulation implementation on QOMET [1], as discussed in Section III.

The emulation-support software RUNE (Real-time Ubiquitous Network Emulation environment) [2] is used to effectively run and manage the emulation, as it can be seen in the overview given in Figure 1. RUNE Master and RUNE Manager are modules used in all RUNE-based emulations for controlling the emulation globally and locally, respectively. The active tag module was specifically designed and implemented for this application. This module includes: (i) Active Tag Communication and chanel spaces, used to calculate and manage the communication conditions between active tags. These functions will be discussed in Section III; (ii) Active Tag Control space, which is powered by the active tag processor (PIC) emulator, and runs the active tag firmware to reproduce the active tag behavior, as it will be discussed in Section IV. The emulation itself is performed using standard PCs (running the FreeBSD operating system) that are part of the StarBED testbed. They are labeled as Execution Units in Figure 1.

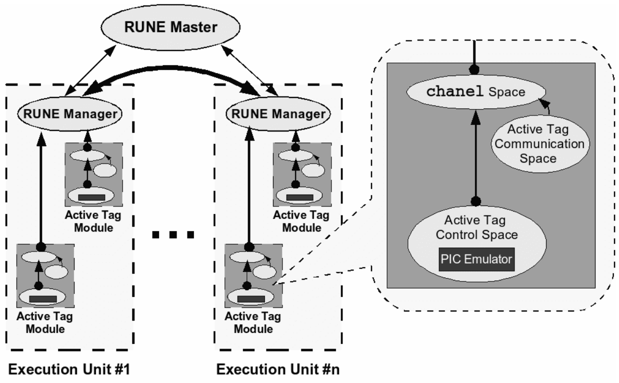


Figure 1. Overview of the active tag emulation system.

III. QOMET

The prototype of the pedestrian tracking system uses the AYID32305 active tags from Ymatic Corporation, also known under the name S-NODE [14]. They were nicknamed communication tags or c-tags in the framework of the current pedestrian localization project. S-NODEs use as processing unit the PIC16LF627A microcontroller. The wireless transceiver of the active tag operates at 303.2MHz, and the data rate is 4800bps (Manchester encoding), which results in an effective data rate of 2400bps. The electric field emitted by active tags is 500uV/m; according to the specification, this produces an error-free communication range of 3-5m.

One of the most important elements when using emulation for studying systems that use wireless communication is to be able to recreate with sufficient realism the communication between them. For the active tags used in our pedestrian tracking system this was accomplished by extending the WLAN emulator QOMET to support the wireless transceiver used by active tags.

QOMET uses a scenario-driven architecture that has two stages. In the first stage, from a real-world scenario representation we create a network quality degradation (ΔQ) description which corresponds to the real-world events (see Figure 2).

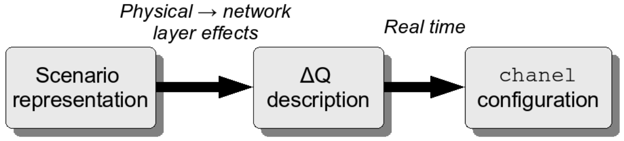


Figure 2. Active tag communication emulation.

The ΔQ description represents the varying effects of the network on application traffic, and the wireless network emulator's function is to reproduce them.

The CHANnel Emulation Library, chanel, is used to recreate scenario-specific communication conditions based on the ΔQ description (FER probabilities) computed by QOMET. Given that we emulate wireless networks, a second function of chanel is to make sure the data is communicated to all the systems that would receive it during the corresponding real-world scenario.

The active tag communication protocol used by the pedestrian tracking system was custom designed as a simple protocol based on time-division multiplexing. Each tag will select at random one of the available communication slots and advertise its identifier and the current time. Currently the number of available communication slots for advertisement messages is 9. There are additional communication slots that can be used on demand to transmit position tracking records from mobile tags to gateways.

This relationship was established based on measurements we made in an RF shielded room with the helicoidally shaped antenna, which was also used in the practical experiment, and 4-byte frames. By fitting a second degree equation on the measurement results we obtained a formula that approximates the relationship between distance and FER, as follows:

$$FER_4(d) = 0.1096d^2 - 0.1758d + 0.0371, \qquad (1)$$

where FER_4 is the frame error rate (the index shows it is based on 4-byte frame measurements) and d is the distance between the receiver and transmitter active tags. The above equation gives a goodness-of-fit coefficient, R^2, equal to 0.9588. Consult [15] for further details of the parameter derivation process.

Since the measurements were done using 4 byte data frames, the result of equation (III) must be scaled accordingly for other frame sizes, as given by:

$$\begin{aligned} FER'_4(d) &= 0.1096\Big(\frac{d}{C}\Big)^2 - 0.1758\frac{d}{C} + 0.0371, \\ FER_4(d) &= \begin{cases} 0, & if\ \frac{d}{C} < 0.5m \\ 1, & if\ FER'_4(d) > 1, \\ FER'_4(d), & otherwise \end{cases} \end{aligned} \qquad (2)$$

where FER represents the frame error rate for a data frame of x bytes, and H is the frame header size in bytes.

Slot collisions arising during the time-multiplexed communication are an additional and independent source of errors. However they are handled during the live emulation in the receiving procedure of the processor emulator.

The main role of chanel is to recreate scenario-specific communication conditions based on the ΔQ description (FER probabilities) computed by QOMET. Given that we emulate wireless networks, a second function of chanel is to make sure the data is communicated to all the systems that would receive it during the corresponding real-world scenario.

IV. PROCESSOR EMULATOR

One advantage of network emulation is that already-existing network applications can be studied through this approach to evaluate their performance characteristics. Although this is relatively easy for typical network applications that run on PCs, the task is complex when the network application runs on a micro controller. In order to execute the active tag application unmodified on our system, we emulate the active tag processor so that the active tag firmware can be run in our emulated environment without any modification or recompilation.

Processor emulation in our system had to take into account the following aspects that we implemented: (i) instruction execution emulation; all 35 PIC instructions are supported by our processor emulator. (ii) data I/O emulation; the only I/O access method used by the active tag application is USART (Universal Synchronous Asynchronous Receiver Transmitter). The application uses USART to interface with the active tag transceiver, and also with the back-end system in the case of gateway tags. (iii) interrupt emulation; all interrupts necessary for the active tag application, i.e., timer0, timer1, and timer2 are supported. We used a pseudo-DMA data transfer technique which is not implemented by the real device instead of emulating the active tag transceiver. It makes easier to integrate the active tag application and the peripheral components of the emulation such as the chanel space, etc.

When emulating active tag applications such as ours it is important to introduce cycle-accurate processor emulation. In our case active tags use the time information contained in messages to synchronize with each others autonomously. Incorrect time information may lead to artificial desynchronization problems and potentially communication errors, therefore it must be avoided.

V. PRELIMINARY TRIAL

The real-world experiment was carried out in March 2007 by Panasonic. Each experiment participant was equipped with an active tag based pedestrian localization system prototype (c-tag).

A group of 16 participants were provided with instructions regarding the path they should follow in the 100 x 300m experiment area. An example of instructions, as received by participant #1 is shown in Figure 4.

The real-world experiment also included a number of tags with known position. These tags are divided into two classes: fixed and gateway c-tags, denoted in Figure 4 by F0 to F3, and GW0 to GW2, respectively. The role of fixed tags is to provide specific information to the mobile c-tags that come in their vicinity to makes it possible to localize those tags. Gateway c-tags, in addition to c-tag communication, also allow information to be transferred between them and to the back-end system as shown in Figure 3.

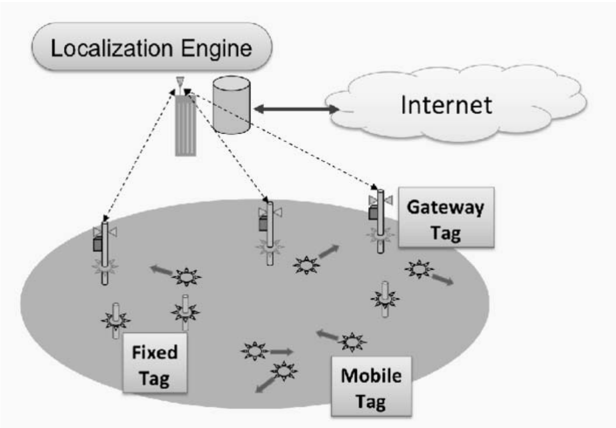
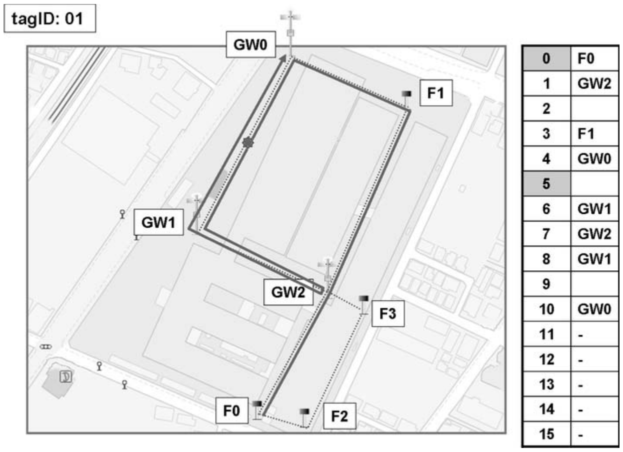


Figure 3. Pedestrian tracking system overview

The gateways are placed at 3 known outdoor locations. Gateways are also connected to the back-end servers; their data is used by the localization engine to determine the trajectories and positions of pedestrians.



0	F0
1	GW2
2	
3	F1
4	GW0
5	
6	GW1
7	GW2
8	GW1
9	
10	GW0
11	-
12	-
13	-
14	-
15	-

Figure 4. Pedestrian movement instructions as received by participant #1.

The real-world experiment was successful in the sense that data collected from the active tags could be used to localize the pedestrians in most cases with sufficient accuracy. The active tag localization approach does not use any GPS-like or triangulation system. Instead the logs of each mobile tag, as collected by gateways, are used. The c-tag logs contain information regarding the time at which other mobile or known-position c-tags were encountered, and their identifiers. This information is used to predict the trajectory of c-tag wearers and track their position.

VI. EMULATION ON TESTBED

In order to validate the emulation system we carried out emulations that reproduced a real-world 16 pedestrian experiment that took place in March 2007 using the prototype of the active tag based pedestrian localization system reported in [11]. The emulation results show the good agreement that exists between the motion patterns of virtual pedestrians, reproduced according to the real-world scenario, and the actual conditions that were recreated in our emulation.

After making the emulation reproduction, the emulation system has started to be utilized to optimize the parameters of the system that are hard to be changed in real-world. Most important parameters are the communication range and the duration of communication cycle. The communication between tags may become unstable when the communication range is too short. On the other hand, too long communication range increases the possibility of packet collision and power consumption. Similarly, more frequent sending of packet makes the system more robust, and at the same time, power-consuming. Those parameters are needed to be set carefully according to the experiment condition such as the density of location-known tags and mobile tags, the size of the area to be covered otherwise they affect the accuracy of the system output. It is, however, quite hard to change those parameters of the real-world system because changing the communication range requires to replace the transceiver or append an attenuator in the circuit. Even though it is possible to change the circuit of the tags, it is still difficult to repeat exactly the same process during the real-world experiment, especially in outdoor environment.

Introducing emulation can solve the previously mentioned problems. So we made two sets of emulations with different parameters based on the scenario of the preliminary trial.

The first set of experiments is to determine how many record packets are exchanged with different communication ranges and virtual processor clocks. In this emulations, changing the clock of the virtual processor also changes the duration of the communication cycle since the firmware of the tags uses software encoding when sending packets. Usually, only changing the clock of processor is not enough to change the the duration of the communication cycle in real-world system unless the transceiver supports higher data rate. In our emulation, the difference of data rate is handled properly by the chanel space.

The second set of emulations is to find out how the number of effective communication slots affects the number of record packets successfully exchanged while the clock of virtual processor changes.

Table I shows what conditions are covered by both sets of emulations.

Table I
THE NUMBER OF EFFECTIVE COMMUNICATION SLOTS WHEN VARYING THE COMMUNICATION RANGE AND VIRTUAL PROCESSOR CLOCK

Frequency [MHz] Range [m]	1	4	16
3	9	9	9
6	3 / 9	3 / 9	3 / 9
9	9	9	9
12	9	9	9
15	9	9	9

VII. RESULTS

The emulation results we present here were obtained by emulation on StarBED. The emulation shown in this section is based on the emulation carried out to reproduce the real-world experiment described in Section V. The motion of pedestrians is exactly the same as the instruction distributed to the real-world experiment participants.

For simplicity each active tag and the associated chanel component are run on one PC. The emulation setup follows the overview presented in Figure 1.

The initial position of the 16 virtual pedestrians and the locations of the 4 fixed c-tags and 3 gateway c-tags, the building topology, and virtual pedestrian movement were all described by converting the real-world experiment instructions to the QOMET XML-based scenario description. Time granularity used when computing communication conditions was 0.5s. RUNE was used to configure the host PCs according to the emulation description and run the emulation.

Figure 5 shows the result of the first set of emulations carried out to determine how many record packets are exchanged with different communication ranges and virtual processor clocks. In the figure, the y-axis shows the traced period, i.e., how much time is tracked, in other words, how much time during emulation the tags succeeded to exchange packets to other tags. If this value reaches 100%, the trajectory is completely tracked without the estimation process done by the back-end servers. As can be seen from Figure 5, the emulations with the virtual processor clock of 1MHz and 4MHz have a peak at the communication range of 12m, 16MHz at 6m respectively. This is because more frequent sending of packets provides a higher possibility of successful exchange of packets, and also less communication range leads to reduce the number of collisions under dense tags condition.

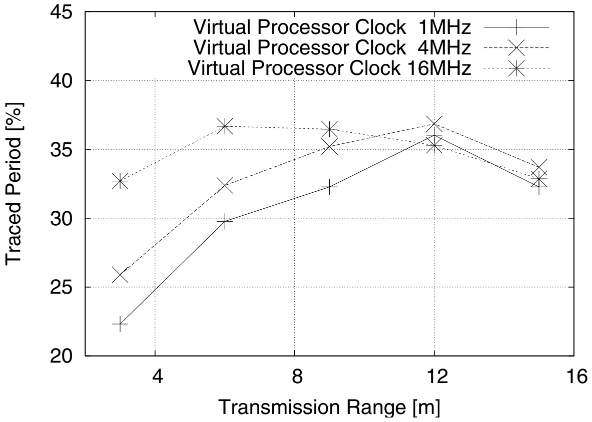


Figure 5. Traced period when varying the communication range and virtual processor clock

Figure 6 shows the result of the second set of emulations done in order to find out how the number of effective communication slots affects the number of record packets successfully exchanged while the clock of virtual processor changes. As Figure 6 shows, the difference of the traced period between different virtual processor clocks becomes bigger when the number of communication slots is limited. This is corresponding to the situation of more dense condition since less number of communication slots triggers frequent occurrence of collision.

VIII. CONCLUSION

In this paper, we presented an emulation system that we designed and developed for active tag applications. This emulation system is currently employed for the development phase emulations of a pedestrian localization system. By using our system it was possible to emulate the system

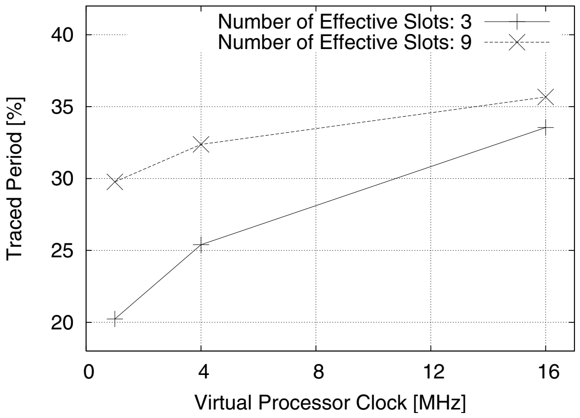


Figure 6. Traced period with reduced number of communication slots when varying virtual processor clock

with a wide range of different parameters that are hard to chang in the real system. This sort of emulation is useful to optimize the system parameters to fit the system to particular condition.

In this paper, we explained utilizing the emulation system in order to optimize the system parameters from mainly reliability point of view. As a future work, we are going to make emulations under more congested condition, and extend the emulation system so that the power-consumption of the system is taken into account. By doing this, the emulation can further improve the system, not only from reliability point of view, but also from sustainability point of view, which is quite important in a battery-driven system like our system.

REFERENCES

[1] R. Beuran, L. T. Nguyen, K. T. Latt, J. Nakata, and Y. Shinoda. Qomet: A versatile wlan emulator. In *AINA '07: Proceedings of the 21st International Conference on Advanced Networking and Applications*, pages 348–353, Washington, DC, USA, 2007. IEEE Computer Society.

[2] J. NAKATA, T. Miyachi, R. Beuran, K. Chinen, S. Uda, K. Masui, Y. Tan, and Y. Shinoda. Starbed2: Large-scale, realistic and real-time testbed for ubiquitous networks. In *The 3rd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities(TridentCom 2007), Orlando, Florida, U.S.A.*, 2007.

[3] A. Janek, C. Trummer, C. Steger, R. Weiss, J. Preishuber-Pfluegl, and M. Pistauer. Simulation based verification of energy storage architectures for higher class tags supported by energy harvesting devices. volume 32, pages 330–339, Amsterdam, The Netherlands, The Netherlands, 2008. Elsevier Science Publishers B. V.

[4] Rutgers University and Wireless Information Network Laboratory. *ORBIT - Wireless Network Testbed*. http://www.orbit-lab.org/. [accessed: Aug. 24, 2009].

[5] D. Johnson, T. Stack, R. Fish, D. M. Flickinger, L. Stoller, R. Ricci, and J. Lepreau. Mobile emulab: A robotic wireless and sensor network testbed. In *INFOCOM*. IEEE, 2006.

[6] J. Zhou, Z. Ji, and R. Bagrodia. Twine: A hybrid emulation testbed for wireless networks and applications. In *INFOCOM*. IEEE, 2006.

[7] P. Levis, N. Lee, M. Welsh, and D. Culler. Tossim: accurate and scalable simulation of entire tinyos applications. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 126–137, New York, NY, USA, 2003. ACM.

[8] J. Polley, D. Blazakis, J. McGee, D. Rusk, and J. S. Baras. Atemu: A fine-grained sensor network simulator. In *Proc. of the First IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004), Santa Clara, California, U.S.A.*, 2004.

[9] Microchip Technology Inc. *MPLAB*. http://www.microchip.com/mplab/. [accessed: Aug. 24, 2009].

[10] Y. Suzuki, T. Kawakami, M. Yokobori, and K. Miyamoto. A real-space network using bi-directional communication tags - pedestrian localization technique and prototype evaluation. In *IEICE Forum on Ubiquitous and Sensor Networks, techni cal report*, 2007.

[11] J. NAKATA, R. Beuran, T. Kawakami, K. Chinen, Y. Tan, and Y. Shinoda. Distributed emulator for a pedestrian tracking system using active tags. In *UBICOMM 2008: Proceedings of the The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, Valencia, Spain*, pages 219–224. IEEE Computer Society.

[12] R. Beuran, J. NAKATA, T. Okada, T. Miyachi, K. Chinen, Y. Tan, and Y. Shinoda. Performance assessment of ubiquitous networked systems. In *5th International Conference on Smart Homes and Health Telematics (ICOST2007), Nara, Japan*, pages 19–26, 2007.

[13] T. Okada, R. Beuran, J. Nakata, Y. Tan, and Y. Shinoda. Collaborative motion planning of autonomous robots. volume 0, pages 328–335, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[14] Ymatic Inc. *S-NODE specification*. http://www.ymatic.co.jp/SNODE2.html. [accessed: Aug. 24, 2009].

[15] R. Beuran, J. Nakata, Y. Suzuki, T. Kawakami, K. Chinen, Y. Tan, and Y. Shinoda. Active tag emulation for pedestrian localization applications. In *5th International Conference on Networked Sensing Systems (INSS 2008), Kanazawa, Ishikawa, Japan*, pages 55–58, 2008.