Emulation System for Active Tag Applications

Razvan Beuran[*,1,2], Junya Nakata[1,2], Takashi Okada[2,1], Tetsuya Kawakami[3],
Ken-ichi Chinen[2,1], Yasuo Tan[2,1], Yoichi Shinoda[2,1]

[1] Hokuriku Research Center,
National Institute of Information and Communications Technology,
Nomi, Ishikawa, Japan
[*] razvan@nict.go.jp
[2] Japan Advanced Institute of Science and Technology,
Nomi, Ishikawa, Japan
[3] Panasonic System Solutions Company,
Panasonic Corporation,
Kohoku-ku, Yokohama, Japan

Abstract—**In this paper we present an active tag emulation system that is used to carry out live experiments with active tag applications. Its two main components are the wireless communication emulator QOMET, and an active tag processor emulator. Experiments are performed using the experiment-support software RUNE on StarBED, the large-scale network testbed at the Hokuriku Research Center of the National Institute of Information and Communications Technology, in Ishikawa, Japan. We illustrate the use of our emulation system with a detailed study of a pedestrian localization application. Emulation is used during the development of the pedestrian localization system so as to perform a wide range of experiments easily and in a repeatable manner.**

Keywords-emulation, active RFID tags, pedestrian localization

I. Introduction

Emulation is an experiment technique that is being used more and more frequently for network-related research. We are currently developing an emulation system for active tag applications. Our goal is to make it possible to carry out realistic experiments during the development of active tag applications by running the active tag firmware within a virtual, emulated environment. Real-world experiments with wireless systems, including active tags, are difficult to organize and perform. Issues such as battery life or undesired interferences often influence experimental results. Through emulation, much of the uncertainties and irregularities of real-world experiments are placed under control. Using the real active tag firmware in experiments enables us to evaluate exactly the same program that will be deployed on the real tags – a significant advantage compared to simulation.

Active RFID tags are employed nowadays in distribution industry, for factory automation, asset tracking, etc. As opposed to passive RFID tags, which are used almost exclusively for identifying purposes, the active communication characteristics of active RFID tags allow for a much wider range of applications in addition to simple identification. One of such applications is that of determining the location of moving objects or people, which is an important issue related to cases such as disaster situations, public surveillance, etc.

The active tag emulation experiments we present are done on StarBED, the large-scale network experiment testbed at the Hokuriku Research Center of the National Institute of Information and Communications Technology (NICT), located in Ishikawa, Japan. In order to be able to use this testbed for active tag emulation, we developed specific subsystems on the basis of existing tools that are already used on StarBED, namely the wireless network emulator QOMET [1], and the experiment-support software RUNE [2].

To the best of our knowledge this may be the first attempt to emulate active tags so as to perform live experiments in realistic conditions. Active tags were so far mainly studied through simulation, such as the work presented in [3]. We created an experimental platform for active tags in which emulation plays an essential role at two points:

- Recreate live wireless communication conditions between active tags, so that information exchange is similar to that occurring in reality;
- Emulate in real time the active tag processor, so that the same firmware that is developed for the active tag prototype system can be run and tested through controlled live experiments.

Public domain wireless communication emulation research is currently mainly done in relation to Wireless LANs (WLANs). One can use real equipments, and hence be subject to potential undesired interferences. Two examples from this class (that allow a controlled movement of wireless nodes, so that mobility effects can be studied), 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.

This work was supported in part by the Japan Science and Technology Agency (JST) under a grant of 2007 3rd round of the Innovating Industry-University Collaboration Seeds project.

Regarding the emulation of the active tag devices themselves, there are already a number of implementations of experiment tools for ubiquitous systems. Some of them 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 intended to run by emulation the same firmware with that used by the real devices. Microchip, the manufacturer of the active tag PIC processor (microcontroller) used in our prototype, only provides two alternatives for system development: real-time emulation in hardware using either the MPLAB REAL ICE In-Circuit Emulator, or the PICMASTER Emulator, or processor simulation using the MPLAB-SIM Simulator [9]. Since none of these solutions were appropriate, we developed our own real-time processor emulator running on standard PCs.

The Panasonic Corporation is developing a pedestrian localization system that employs active tags, and carried out some real-world experiments, as reported in [10]. The results are being used as a basis for improving the prototype of the pedestrian localization system. 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. The system was first briefly presented in [11]. As a case study, in this paper we discuss more thoroughly the real-world experiment and the active tag pedestrian localization system that we emulated, as well as demonstrate the applicability of active tag emulation through several experimental results.

II. System Overview

One solution to carry out experiments in a repeatable manner and in a wide-range of controllable conditions is to use the technique of emulation. This implies creating a virtual environment in which the movement, the communication, and the behavior of active tags are all reproduced. The requirements that differentiate emulation from simulation are:

- Emulate the wireless communication of the active tags in live experiments;
- Emulate the active tag processor so that the same firmware used by the real devices can be tested in emulation experiments.

The pedestrian localization system prototype was used as guidance during the design and implementation of the emulation testbed. In addition, we put to use our previous experience with emulation systems, such as those presented in [12, 13]. A key element in our approach is to base the wireless communication emulation implementation on QOMET, as it will be discussed in Section 3.

The other important component is the experiment-support software RUNE [2]. RUNE (Real-time Ubiquitous Network Emulation environment) was designed to support emulation of large ubiquitous networks, having features such as: emulate the surrounding environment, support real-time concurrent execution of numerous nodes, provide multi-level emulation layers, etc. The basic element of the logical structure of a RUNE-driven emulation is the *space*. A space is an entity that behaves as one of the emulated devices, depending on its function. Spaces can emulate: nodes, i.e., physical devices, such as the active tags in our experiment; environments, such as the thermal field in a room; networks, i.e., network communication channels, such as the active tag wireless network. Spaces are connected with each other by *conduits*. A conduit represents an error-free communication channel between two spaces, implemented transparently to the user. Spaces can be executed on the same host or on different hosts. In order to emulate real communication channels, in which loss, delay and bandwidth limitations occur, an appropriate network space needs to be used in-between the communicating node spaces.

An overview of the system that is currently used for active tag emulation experiments, and for the development of the pedestrian localization prototype, is given in Fig. 1. Although we discuss here a particular case of active tag emulation related to pedestrian localization, the system is generic enough to be used for other applications as well. By using specific communication models and processor emulators, any other active RFID tag can be emulated.

RUNE Master and RUNE Manager in Fig. 1 are modules used in all RUNE-based experiments for controlling the experiment globally and locally, respectively. The active tag module was specifically designed and implemented for this application, and includes:

- Active Tag Communication and *chanel* (channel emulator) spaces, used to calculate and manage the communication conditions between active tags;
- Active Tag Control space, which is powered by the active tag processor (PIC) emulator, and runs the active tag firmware to reproduce active tag behavior.

The experiment itself is performed using PCs running the FreeBSD operating system that are part of the StarBED testbed. They are labeled as Execution Units in Fig. 1.

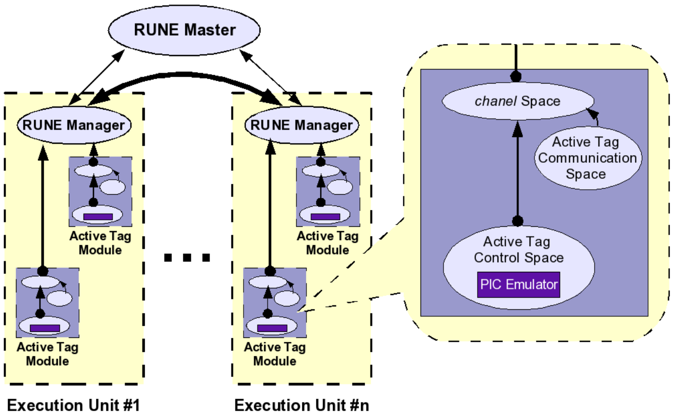


Figure 1. Active tag emulation system overview.

III. Wireless Emulation

The active tags employed in the pedestrian localization experiment use wireless communication to exchange information with each other. Therefore, one of the most important elements when using emulation for studying systems such as that discussed in this paper is to be able to recreate with sufficient realism the communication between the emulated active tags. This was accomplished by extending the WLAN emulator QOMET to support the wireless transceiver used by active tags. This task was facilitated by the modular architecture of QOMET.

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 Fig. 2). This is done by modeling the effects at the different layers of the communication protocol, from physical to network layer. This step can be done either off-line, if conditions are known in advance, or in real time, during the emulation process.

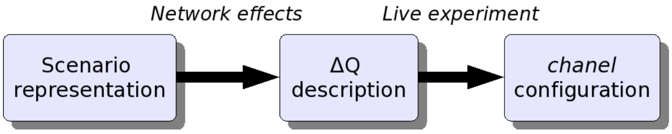


Figure 2. Scenario-driven two-stage wireless emulation.

The ΔQ description represents the varying effects of the network on application traffic, and the wireless network emulator's function is to reproduce them. Hence we apply in real time the ΔQ description calculated in the first stage to the traffic communicated during the emulation process, so as to replicate the user-defined scenario.

QOMET was adapted to the present project by adding two specific components to the already existing framework:

- A model of the active tag wireless transceiver communication, so as to compute the active-tag specific ΔQ description;
- A module to apply in real time the computed ΔQ description to the non-IP traffic exchanged by active tags.

The active tag communication model was described in detail in [11]. This model is used to obtain the ΔQ description that corresponds to the active tag communication conditions in a certain user-defined scenario. The conversion is from real-world scenario to physical layer, and then data link layer, as required by the QOMET framework. Thus, our model establishes the relationship between the distance between tags (a real-world scenario parameter) and the Frame Error Rate (FER, a data link layer parameter). This conversion is done based on measurements we made in an RF shielded room.

Given the time-multiplexed communication (see Section V.A), there is an additional and independent source of errors: slot collision. This is taken into account in real time during the live experiment in the transmitting and receiving procedures of the active tag control space. When two or more data frames are transmitted and/or arrive during the same time slot, it is considered that a collision effect has taken place, and all the respective frames are discarded. Other parameters directly reproduced during the live data transmission/reception by the active tag control space are the communication delay and bandwidth.

As active tags do not generate IP traffic, we could not use a wired-network emulator for introducing network layer effects to traffic, as previously done when using QOMET. Consequently, we implemented our own emulation system, named *chanel* (CHANnel Emulation Library). This module is inserted between the Active Tag Control Space and its connection to other spaces using conduits (cf. Fig. 1). The advantage of this integration is that it doesn't matter for the emulation system whether RUNE spaces are executed on the same PC or on different PCs, since communication itself is handled transparently by RUNE conduits. The main role of *chanel* is to recreate scenario-specific communication conditions based on the ΔQ description (FER probabilities) computed by QOMET. This function is similar to that of any wired-network emulator, such as *dummynet* [15].

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. This is done by using the ΔQ description to decide the conditions for the communication between the current active tag and the other active tags in its transmission range, and forward the data through conduits to the appropriate tags. It should be noted that, since transmission on Ethernet takes considerably longer than wireless transmission, we had to decrease the execution speed of the entire emulation process, so that timing of events is not affected. At the moment there is a ten times slower-than-real execution speed, but we are working on improving this figure.

IV. Processor Emulation

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 to do for typical network applications that run on PCs, the task is complex when the network application runs on a special processor. In order to execute the active tag application on our system, we emulate the 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:

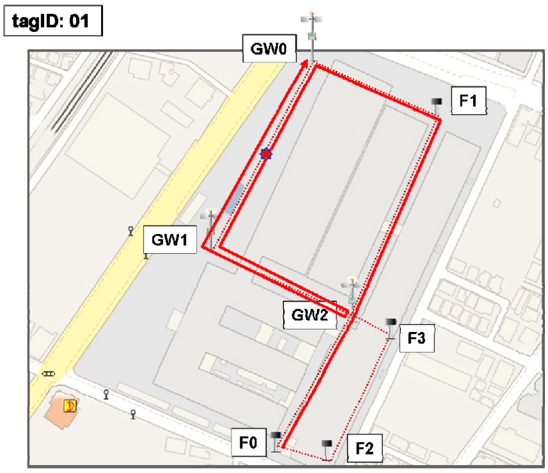
- Instruction execution emulation; almost all 35 PIC instructions are supported by our processor emulator;
- Data I/O emulation for USART (Universal Synchronous Asynchronous Receiver Transmitter);
- Interrupt emulation of all interrupts necessary for the active tag application.

Note that some features couldn't be reproduced mimetically. For instance, the random number generator algorithm of the real tags was implemented using an undocumented feature of the processor. Since this undocumented feature could not be reproduced in the emulator, we had to use a workaround and use the PC pseudo-random number generator instead of the real tag algorithm.

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 potential communication errors, therefore it must be avoided. According to the results shown in [11], good accuracy is obtained for more than 10 instances running in parallel on 1.13 GHz systems, and up to about 45 instances on a 2.33 Core2 system, a performance level which is sufficient for our needs.

V. Real-World Experiment

The real-world experiment was carried out by Panasonic Corporation as follows. A group of 16 participants were provided with instructions regarding the path they should follow in the 100 x 300 m experiment area. The instructions indicate the experiment topology (left-hand side), as well as the time in minutes when each known location should be reached (right-hand side). An example of instructions, as received by participant #1 is shown in Fig. 3.



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 3. Pedestrian movement instructions as received by participant #1.

The experiment lasted 15 minutes, and followed a scenario in which a disaster situation leads to the necessity of area evacuation. After 5 minutes of unrelated walking patterns for each individual or couple of pedestrians, it was considered that an event took place, such as a fire or earthquake, which required an evacuation procedure. Following this event all pedestrians were instructed to proceed to an assembly point located in the upper left corner of the area (GW0 in Fig. 3). Participants also received a stopwatch, to try to ensure they follow as closely as possible the indicated scenario.

The real-world experiment also included a number of tags with known position. These tags are divided into two classes: fixed and gateway tags, denoted in Fig. 3 by F0 to F3, and GW0 to GW2, respectively. Fixed tags cannot communicate information except by using the tag protocol and wireless network. They are placed at 4 known locations, both outdoors and indoors. The role of fixed tags is to provide specific information to the mobile tags that come in their vicinity to make it possible to localize those tags. Gateway tags, in addition to tag communication, also allow information to be transferred between them and to the outside world by using the IEEE 802.11j standard. The gateways are placed at 3 known outdoor locations and are connected to each other in an ad-hoc style network. Gateway data is used by the localization engine to determine the pedestrian trajectories.

A. Active Tag Specifications

The active tags used for the pedestrian localization system by the Panasonic Corporation are the AYID32305 tags from Ymatic Corporation [14]. They were nicknamed c-tags (communication tags) in this project. Other active tag devices may be integrated into our emulation system in the future.

The c-tags use as processing unit the PIC16F648A microcontroller, which has 4096 word flash memory, 256-byte SRAM and 256-byte EEPROM. Its frequency is configured at 4 MHz for this application. The wireless transceiver of the active tag operates at 303.2 MHz, and the data rate is 4800 bps (Manchester encoding), which results in an effective data rate of 2400 bps. The electric field emitted by c-tags is 500 μV/m; according to Ymatic, this produces an error-free communication range of 3-5 m. The range varies depending on the antenna used. Three choices of antennas are available: embedded loop, helicoidally shaped or flat spiral antenna. Measurements in an RF shielded room where carried out to determine the range corresponding to each antenna.

The active tag communication protocol was custom designed by Panasonic Corporation, and it is based on the concept of time-division multiplexing. Each tag will select at random one of the available communication slots and advertise its identifier and the current time. The random selection may induce reception errors if two c-tags choose the same slot. Currently the number of available communication slots for advertisement messages is 9. Message exchanges take place at intervals of about 2 seconds. There are additional communication slots that can be used on demand to transmit position tracking records from mobile tags to gateways.

B. Method and Discussion

The data collected from active tags during the real-world experiment could be used to localize the pedestrians in most cases with sufficient accuracy. The active tag localization approach doesn't 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 position and trajectory of c-tag wearers. The basic equation used to calculate the position P_x of a pedestrian at time t_x is:

$$P_x = P_i + \left(P_j - P_i\right)\frac{t_x - t_i}{t_j - t_i}, \quad (1)$$

where P_i and P_j are the known positions of the pedestrian (from logs) at moments of time t_i and t_j, with $t_i \leq t_x \leq t_j$ [10]. This simple equation assumes linear motion, which is close enough to reality when considering general localization applications.

While experimenting with the prototype active tag based localization system under various circumstances, a series of problems were identified. The wireless communication between the prototype active tags was not always reliable; in addition, battery depletion was relatively fast and caused signal to weaken during and between experiments. A new prototype is being built using a better transceiver and a more efficient processor. Orchestrating a real-world experiment using even as few as 16 people was a cumbersome task (performing the 15 minutes experiment required a few hours of preparation). The persons involved needed to be briefly trained on how to read the scenario instructions, and how to behave during the experiment. However, due to the human factor involved there were several occasions on which participants didn't accurately follow the instructions. This has consequences on experimental result interpretation, and on the repeatability of the tests. Hence we decided to use emulation for performing with ease a large number of experiments during system development phase.

VI. Experimental Results

The experimental results we present here were obtained by emulation on StarBED. For simplicity each active tag and the associated *chanel* component are run on one PC. The experimental setup follows the overview presented in Fig. 1. The experiments shown use exactly the same base conditions as the real-world experiment described in Section 5. They demonstrate the applicability of the emulation approach in the development of active tag systems. Note that we previously validated our system by showing in [11] the good agreement between the trajectory of the pedestrian and the positions localized by the system.

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

First of all, emulation was used to analyze the communication protocol of the active tags, since it provided an insight into the timing of the messages sent and received by active tags. One issue we were able to identify by emulation experiments is related to the time synchronization between active tags. The algorithm used by mobile active tags to synchronize their clock based on the time received from neighboring tags was faulty. Actually, better performance was obtained in emulation experiments by not using the faulty time synchronization. The synchronization algorithm is currently being reviewed by the firmware developers for improvement.

One of the strong arguments regarding the usage of an emulator is the fact that various conditions can be easily investigated without the need of expensive real-world experiments. To illustrate this aspect, we performed several tests in which the purpose was to discover what transmission range would provide best localization results. Transmission range is dependent on the antenna and transmission power of active tags; therefore it can be configured in reality as well. In Fig. 4 we show the results of these tests for 5 different transmission ranges. For each range 3 tests were carried out, and we used error bars to show the variation of the results. Although we only performed a relatively small number of tests for each set of parameters, this is enough to illustrate the usability of our experimental platform. A more thorough investigation is of course needed in order to draw definitive conclusions.

We compute the localization error as the mean of the distances at each moment of time between the trajectory tracked by the pedestrian localization system, and the emulated pedestrian trajectory in the virtual environment. As it can be seen, results indicate that a range of 9 m is the most appropriate for the investigated scenario. This is because shorter ranges lead to communication being unsuccessful on occasions, hence to localization errors. Meanwhile, larger ranges decrease the accuracy in identifying the position of a tag, since the area in which tags can communicate with each other is rather large.

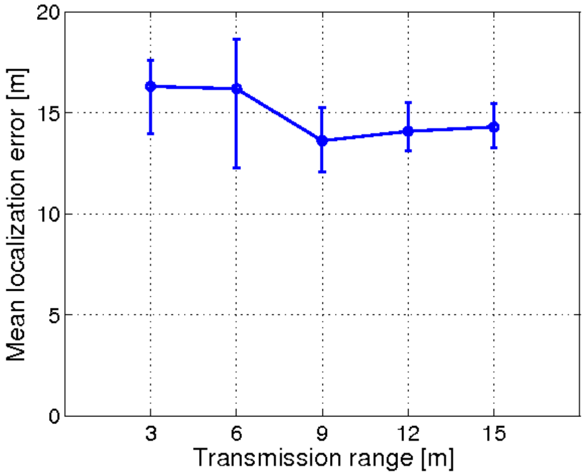


Figure 4. Mean localization error for several transmission ranges.

Another series of experiments was performed to study the effect of the number of slots used in the time-multiplexed communication on overall localization performance. The prototype used 9 slots for communication, preceded and followed by one guard slot (hence a total of 11 slots). During emulation, for two communication ranges, 3 m and 9 m respectively, we configured the active tag firmware to use 3, 6, and 9 communication slots (see Fig. 5). The advantage of using a smaller number of slots is that the active duration is decreased, and therefore battery life is potentially increased by 30% (when using 6 slots), or even by more than 50% (when using 3 slots). The disadvantage is that, when several tags want to communicate with each other, a small number of slots leads to a high collision rate, and impedes information exchanges.

Fig. 5 shows that there is a performance gain when increasing the number of slots, but this gain is not so evident for the 3m range, since the communication opportunities are rather low compared to the case of 9 m range. For 9 m range, the advantage of using more than 3 slots is quite visible, and an optimum performance level seems to be reached for 6 slots.

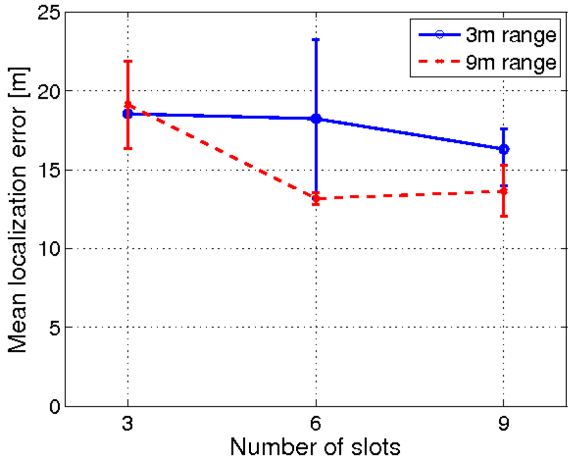


Figure 5. Effect of the number of slots used in time-multiplexed communication on localization performance.

VII. Conclusions

In this paper we presented an emulation system that we designed and developed for active tag applications. This emulation system is currently employed in the development phase of a pedestrian localization system by Panasonic Corporation. By using our system it was possible to simplify the testing procedures of the localization engine, and identify several firmware implementation issues. However, these are not the only benefits of emulation. Using our experimental platform it is also possible to conveniently perform tests in a wide range of controlled experimental conditions, thus being able to investigate various design choices.

In order to show the applicability of the emulation approach, we presented results of tests following the real-world 16 pedestrian experiment that used the prototype of the active tag based pedestrian localization system. The examples shown demonstrate how one can easily use our system to carry out the validation of the localization system, as well as explore the parameter space of the system, for instance in order to determine optimum values of certain system parameters.

Our future work has several main directions: improve the scalability of the system so as to enable experiments of pedestrian groups as large as 1000; improve the realism of the wireless communication emulation by using more accurate 3D models for topology and electromagnetic wave propagation. We also plan to combine a behavioral pedestrian motion model with a GIS-based urban area description so as to create a realistic pedestrian trajectory generator for large-scale urban experiments with our emulation system.

References

[1] R. Beuran, L. T. Nguyen, K. T. Latt, J. Nakata, Y. Shinoda, “QOMET: A Versatile WLAN Emulator”, IEEE International Conference on Advanced Information Networking and Applications (AINA-07), Niagara Falls, Ontario, Canada, May 21-23, 2007, pp. 348-353.

[2] J. Nakata, T. Miyachi, R. Beuran, K. Chinen, S. Uda, K. Masui, Y. Tan, Y. Shinoda, “StarBED2: Large-scale, Realistic and Real-time Testbed for Ubiquitous Networks”, TridentCom 2007, Orlando, Florida, U.S.A., May 21-23, 2007.

[3] A. Janek, Ch. Trummer, Ch. Steger, R. Weiss, J. Preishuber-Pfluegl, M. Pistauer, “Simulation Based Verification of Energy Storage Architectures for Higher Class Tags supported by Energy Harvesting Devices”, 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD 2007), Lubeck, Germany, Aug. 29-31, 2007, pp. 463-462.

[4] Rutgers University, Wireless Information Network Laboratory , “ORBIT – Wireless Network Testbed”, http://www.orbit-lab.org/.

[5] D. Johnson, T. Stack, R. Fish, D. Montrallo Flickinger, L. Stoller, R. Ricci, J. Lepreau, “Mobile Emulab: A Robotic Wireless and Sensor Network Testbed”, IEEE INFOCOM 2006, Barcelona, Spain, April 23-29, 2006.

[6] J. Zhou, Z. Ji, R. Bagrodia, “TWINE: A Hybrid Emulation Testbed for Wireless Networks and Applications”, IEEE INFOCOM 2006, Barcelona, Spain, April 23-29, 2006.

[7] P. Levis, N. Lee, M. Welsh, D. Culler, “TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications”, Proc. of the First ACM Conference on Embedded Networked Sensor Systems (SenSys’03), Los Angeles, California, U.S.A., November 5-7, 2003.

[8] J. Polley, D. Blazakis, J. McGee, D. Rusk, J. S. Baras, “ATEMU: A Fine-grained Sensor Network Simulator”, Proc. of the First IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004), Santa Clara, California, U.S.A., October 4-7, 2004.

[9] Microchip Technology, Inc., http://www.microchip.com/.

[10] Y. Suzuki, T. Kawakami, M. Yokobori, K. Miyamoto, “A Real-Space Network Using Bi-Directional Communication Tags – Pedestrian Localization Technique and Prototype Evaluation”, IEICE Forum on Ubiquitous and Sensor Networks, technical report, October 30-31, 2007.

[11] R. Beuran, J. Nakata, Y. Suzuki, T. Kawakami, K. Chinen, Y. Tan, Y. Shinoda, "Active Tag Emulation for Pedestrian Localization Applications", 5th International Conference on Networked Sensing Systems (INSS08), Kanazawa, Ishikawa, Japan, June 17-19, 2008, pp. 55-58.

[12] R. Beuran, J. Nakata, T. Okada, T. Miyachi, K. Chinen, Y. Tan, Y. Shinoda, “Performance Assessment of Ubiquitous Networked Systems”, 5th International Conference on Smart Homes and Health Telematics (ICOST2007), Nara, Japan, June 21-23, 2007, pp. 19-26.

[13] T. Okada, R. Beuran, J. Nakata, Y. Tan, Y. Shinoda, "Collaborative Motion Planning of Autonomous Robots", 3rd International Conference on Collaborative Computing (CollaborateCom 2007), White Plains, New York, U.S.A., November 12-15, 2007.

[14] Ymatic, Inc., http://www.ymatic.co.jp.

[15] L. Rizzo, “Dummynet FreeBSD network emulator”, http://info.iet.unipi.it/~luigi/ip_dummynet.