Active Tag Emulation for Pedestrian Localization Applications

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

[1] National Institute of Information and Communications Technology, Ishikawa, Japan
[2] Japan Advanced Institute of Science and Technology, Ishikawa, Japan
[3] Matsushita Electric Industrial Co., Ltd., Kanagawa, Japan

***Abstract*—Active tags are currently used for various tasks in transport and distribution industries, for factory automation or asset tracking. Pedestrian localization is another application of active tags, for which we develop a technique and a practical system. Following preliminary real-world experiments with a system prototype, we continued to develop our active tag based pedestrian localization technique by an emulation approach that is discussed in this paper. We use emulation during development so as to be able to perform large-scale experiments with the pedestrian localization system easily and in a repeatable manner. The experimental results show the good agreement that exists between the real-world pedestrian scenario and the measurements made with the emulated system.**

***Index Terms*—active tag, wireless network emulation, processor emulation, pedestrian localization**

I. Introduction

Pedestrian localization is an important issue related to several circumstances, such as disaster situations, public surveillance, etc. In the case of evacuation procedures, for instance evacuation of a school following an earthquake, it is important to be able to determine quickly and easily whether evacuation was completed successfully or some persons are still within the disaster perimeter. Identifying automatically the present location of the potential victims is also vital.

We are currently developing a system for pedestrian localization that employs active tags. This application can make use of the data communication and processing features 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 some real-world experiments were carried out in March 2007 within the perimeter of the Panasonic System Solutions Company, Yokohama plant, as reported in [1]. The experiment consisted in the orchestrated movement, both in indoor and outdoor environments, of 16 pedestrians wearing localization system active tag based prototypes nicknamed communication tag, or c-tag. The real-world experiment also included a number of tags with known position: fixed c-tags and gateway c-tags. Gateway c-tags can transfer information between them and to the outside world using the 802.11j standard. The data provided by them is used by the pedestrian localization engine.

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.

While experimenting with the prototype active tag based localization system under various circumstances, a series of problems were identified, such as: (i) battery depletion was relatively fast and caused signal to weaken during and between experiments; (ii) orchestrating a real-world experiment using even as few as 16 people is a troublesome task (a 15 minutes experiment needed hours of preparation).

As a consequence we designed and implemented an alternative experiment platform using emulation, intended for the system development phase. The necessary emulation subsystems are based on existing tools, namely the wireless network emulator QOMET [2], and the experiment support software RUNE [3]. To the best of our knowledge this may be the first attempt to emulate active tags so as to perform experiments in realistic conditions.

Active tags were so far mainly studied through simulation (e.g., [4]). There exist already a number of experiment tools for ubiquitous systems. Some of them focus on the operating system level, such as TOSSIM [5], which is a TinyOS simulator. ATEMU [6] is able to emulate TinyOS applications at processor level, and supports other platforms too. The manufacturer of the active tag processor (microcontroller) used in our prototype, *Microchip*, only provides two alternatives for system development: real-time emulation in hardware, or processor simulation [7]. Since none of these solutions are appropriate for our purpose, we developed our own real-time processor emulator running on standard PCs.

Wireless communication emulation is currently mainly done in relation to WLANs. TWINE [8] uses computer models for performing real-time experiments so as to avoid undesired interferences and side effects. TWINE is a wireless emulator that combines wireless network emulation and simulation in one setup, but only supports 802.11b WLAN so far. Hence we extended QOMET, which uses a similar approach, to meet active tag communication emulation requirements.

II. Emulation System Overview

To make it possible to carry out experiments in a wide-range of controllable conditions, and in a repeatable manner, we use the technique of emulation. This implies creating a virtual environment in which the movement of pedestrians, the communication, and the behavior of active tags are reproduced in real time. The conclusions of the real-world experiment using the pedestrian localization prototype system that were mentioned in the previous section were 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 [9, 10]. A key element in our approach is to base the wireless communication emulation implementation on QOMET [2], as it will be discussed in Section III.

The other important component is the experiment-support software RUNE (Real-time Ubiquitous Network Emulation environment) [3]. The basic element of the logical structure of RUNE-driven emulation is the *space*, a module behaving as one of the emulated devices. Spaces are connected with each other by elements called *conduits*.

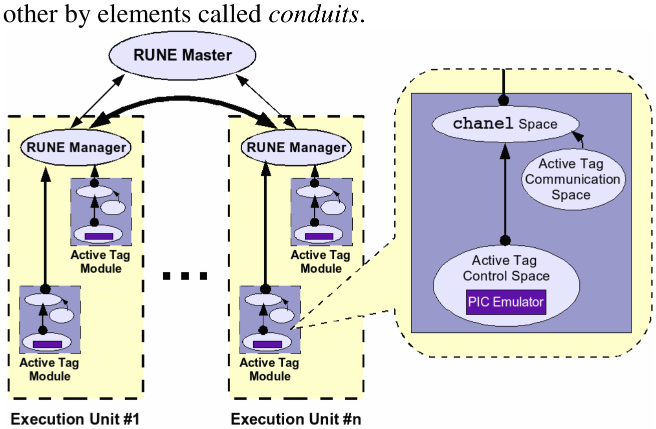

Fig. 1. Active tag emulation system overview.

An overview of the system that we designed for active tag emulation experiments and the development of the pedestrian localization prototype is given in Fig. 1. RUNE Master and RUNE Manager 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. This module includes: (i) Active Tag Communication space and `chanel` space, managing inter-tag communication; (ii) Active Tag Control space, powered by the active tag processor (PIC) emulator, executing active tag firmware. The experiment itself is performed using standard PCs that are part of StarBED, the large-scale network experiment testbed at the Hokuriku Research Center, National Institute of Information and Communications Technology (NICT), Ishikawa, Japan [11].

III. Wireless Communication Emulation

The active tags employed in the pedestrian localization experiment use wireless communication to exchange information with each other. The wireless transceiver of the prototype system 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 energy emitted by S-NODE active tags is 500 µV/m; according to Ymatic, this gives an error-free communication range of 3-5 m depending on the antenna used [12]. We extended the WLAN emulator QOMET [2] to support the wireless transceiver used by active tags. This task was facilitated by the modular architecture of QOMET, as we could borrow many of the already existing components.

A. Active Tag Communication Model

Our current model for active tag communication establishes the relationship between the distance between two nodes (a real-world scenario parameter) and the Frame Error Rate (FER, a data link layer parameter). This conversion is done based on measurements we carried out in an RF shielded room. The results used below are those obtained with a helicoidal-shaped antenna, the one used in the practical experiment, and 4-byte frames. By fitting a second degree equation on the measurement results we obtained the following equation:

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

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. Note that the result of equation (1) represents a probability, therefore needs to be adjusted if less than 0 or larger than 1.

FER given by equation (1) must be "scaled" accordingly for frame sizes other than 4 bytes, as given by:

$$FER_x = 1 - (1 - FER_4)^{(H+x)/(H+4)}\,, \qquad (2)$$

where FER_x represents the frame error rate for a data frame of x bytes (usually 7), and H is the frame header size (6 bytes).

The frame error rate induced by slot collision due to time-multiplexed communication can be approximated by the following simplified relation:

$$FER_S = n/(N_{slots})^2\,. \qquad (3)$$

Considering that FER_x is equal to 1 for out of range transmitters, the number n can be computed at each moment of time as the cardinal of the set of c-tags, E, for which the frame error probability due to distance when received by the current tag is inferior to 1:

$$n = |E|,\ E = \{e \mid FER_x < 1\}\,. \qquad (4)$$

Finally, the overall frame error rate, FER, can be computed by taking into account the fact that the two error causes discussed above are independent, as follows:

$$FER = FER_x + FER_S - FER_x \cdot FER_S\,, \qquad (5)$$

The communication model for active tags presented here reproduces the measurements we carried out. To increase realism we intend to extend it for the new generation of active tag prototypes using a log-distance path loss model, similar to the approach in [2].

For active tags, FER is the only parameter that needs to be computed. FER is equivalent to packet loss in the absence of a network layer. Communication delay and bandwidth limitation are directly recreated during data transmission & reception by the active tag control space and the PIC emulator.

B. Communication channel emulation library

Given that the active tags are not generating IP traffic, we could not use a wired-network emulator for introducing network layer effects to traffic, as previously done when using QOMET [2]. As a consequence we implemented a channel emulation system, named `chanel` (CHANnel Emulation Library). This module is inserted between the space emulating the c-tag (Active Tag Control Space in Fig. 1) and its connection to the other spaces using conduits. The advantage of this integration is that it becomes transparent from the point of view of emulation 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` [13]. 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 wireless scenario.

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 for typical network applications that run on PCs, the task is complex when the network application runs on a special processor. In order to run the active tag application unmodified on our system, we emulated the active tag processor so that the same firmware that runs on active tags can be run in our emulated environment without any modification nor recompilation.

The prototype system uses AYID32305 tags from Ymatic Corporation, also known under the name S-NODE [12]. S-NODEs use as processing unit the PIC16LF627A microcontroller, which has 1024 word flash memory, 224 byte SRAM and 128 byte EEPROM; its frequency is 4 MHz.

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.

In addition, processor emulation had to take into account the following aspects that we implemented:

- Instruction execution emulation; all 35 PIC instructions are supported by our processor emulator, except for one of the processor states, *sleep*, since this state is not needed for the active tag application;
- 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 it as the interface to the active tag transceiver. Our processor emulator supports not only access to the whole memory area, but also serial-parallel conversion so that `chanel` can send/receive sequences of bytes (although the real data is sent/received in bit-by-bit manner by the real active tag application). This was necessary so as to optimize data transfer at PC level.
- Interrupt emulation; all interrupts necessary for the active tag application, i.e., *timer0*, *timer1*, and *timer2* are supported. At the moment, other interrupts, such as comparator interrupt, USART TX/RC interrupt, and so on, are not supported.

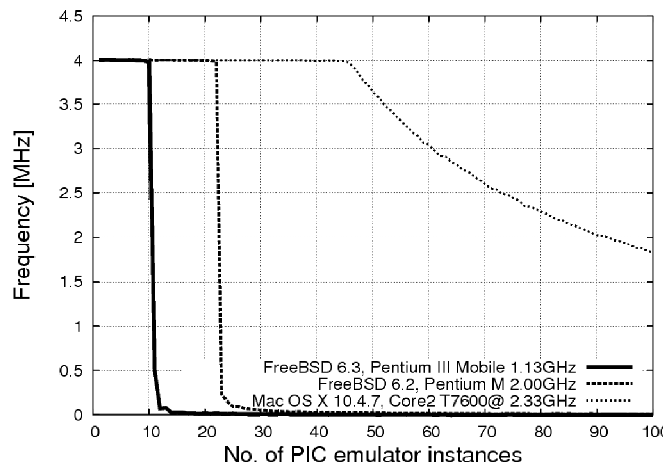

Fig. 2. PIC emulator frequency accuracy depending on the number of instances for 3 different computing platforms.

Besides the available functionality such as that mentioned above, one of the main concerns regarding a processor emulator is how accurately the execution speed is reproduced, especially in the case when running multiple instances of the emulator. In Fig. 2 we show how accuracy changes depending on the computing platform used and the number of instances of the PIC emulator that are run in parallel. We remind that the intended execution frequency is 4 MHz. The figure shows that, given a sufficiently powerful platform, good accuracy is obtained for up to about 45 instances. We are still investigating the cause of the rapid performance degradation for the tests ran on FreeBSD 6.3 and 6.2 systems with more than 10 or 22 PIC emulator instances, respectively. A possible cause may be the thread scheduling algorithm of the FreeBSD operating system.

V. EXPERIMENTAL RESULTS

The experimental results we present here were obtained by emulation on StarBED in a setup following the overview presented in Fig. 1. The initial positions of the 16 pedestrians and their movement, the locations of the 4 fixed c-tags (F0 to F3) and 3 gateway c-tags (GW0 to GW2), and the building topology were described by converting the real-world experiment information to QOMET scenario description. The time granularity used when computing communication conditions and during real-time execution was of 0.5 s. RUNE was used to configure the host PCs and to run the experiment.

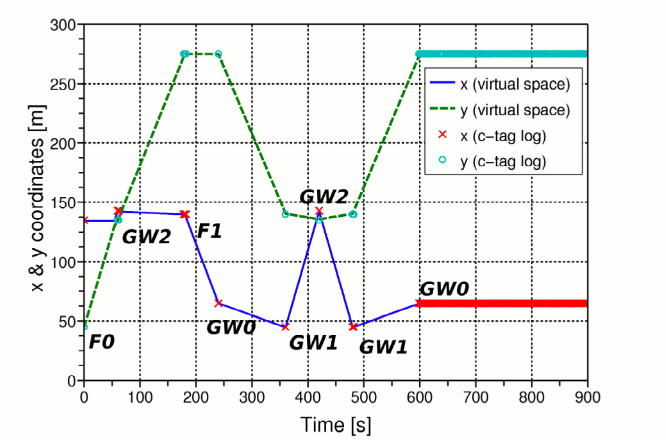

Fig. 3. Trajectory of the c-tag in the virtual space, and the position tracked using c-tag logs collected in the emulation experiment for pedestrian number #1.

In Fig. 3 we plot position localization data concerning pedestrian #1. The figure depicts with continuous line the trajectory of the pedestrian (x & y coordinates versus time), as it took place in the virtual environment of our emulation system. This trajectory followed the real-world experiment scenario that we carried out in March 2007. With "x" and "o" marks is represented the position on the virtual space x & y axes of the fixed c-tags or gateways that communicated with the pedestrian mobile c-tag at various moments of time. This time information is determined using the logs of the communication with the corresponding c-tag, which contains the time and the id of the encountered active tags. This log allows us to identify very easily the location of the c-tag at several moments of time, and to calculate its trajectory. To facilitate reading we annotated the plot with the names of the encountered fixed or gateway c-tags.

One can see the good agreement between the trajectory of the pedestrian (continuous lines) and the positions localized by the system ("x" and "o" marks). Communication between mobile nodes and fixed nodes only takes place when the mobile node is in the virtual vicinity of a fixed node. This proves that mobile node motion in the virtual space and wireless communication condition recreation (using QOMET and `chanel`), as well as c-tag execution (using RUNE) are perfectly synchronized and in accordance with our expectations from the real-world experiment.

VI. CONCLUSION

We presented an active tag emulation system with application to pedestrian localization. This system is currently employed by Panasonic for the development of the pedestrian localization system to simplify the development and testing procedures of the localization engine. In our experimental platform emulation plays an essential role at two points: (i) recreate in real time the wireless communication conditions between active tags; (ii) emulate in real time the active tag processor. The experimental results show the good agreement that exists between the virtual conditions reproduced through emulation, and the actual real-world experiment conditions.

Our future work has several directions: improve the scalability of the system to enable emulation experiments with pedestrian groups as large as 1000; improve the realism of the wireless communication emulation by using more accurate 3D models for topology and electromagnetic propagation; create a realistic pedestrian trajectory generator for large-scale experiments.

REFERENCES

[1] 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*, tech. report, October 30-31, 2007.

[2] R. Beuran, L. T. Nguyen, K. T. Latt, J. Nakata, Y. Shinoda, "QOMET: A Versatile WLAN Emulator", *IEEE 21st Int. Conf. AINA-07*, Niagara Falls, Ontario, Canada, May 21-23, 2007, pp. 348-353.

[3] 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.

[4] 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", *Conf. on Digital System Design Architectures, Methods and Tools (DSD 2007)*, Lubeck, Germany, Aug. 29-31, 2007, pp. 463-462.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] T. Miyachi, K. Chinen, Y. Shinoda, "Automatic Configuration and Execution of Internet Experiments on an Actual Node-based Testbed", Proc. of Tridentcom 2005, Trento, Italy, February 2005, pp. 274-282.

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

[13] L. Rizzo, "Dummynet FreeBSD network emulator", *http://info.iet.unipi.it/~luigi/ip_dummynet*.