

A Multi-purpose Wireless Network Emulator: QOMET

Razvan Beuran^{†,*}, Junya Nakata^{*,†}, Takashi Okada^{*,†}, Lan Tien Nguyen^{*},
Yasuo Tan^{*,†}, Yoichi Shinoda^{*,†}

[†] *National Institute of Information and Communications Technology, Ishikawa, Japan*

^{*} *Japan Advanced Institute of Science and Technology, Ishikawa, Japan*

E-mail: razvan@nict.go.jp

Abstract

In this paper we present the design and usage of the multi-purpose wireless network emulator that we develop, QOMET. QOMET implements a two-stage scenario-driven approach to wireless network emulation. We give here the details of the improved model and additional functionality that make it currently possible to use QOMET to convert a real-world scenario to a sequence of network-condition descriptors for a wide range of experiments. QOMET was used on a real network testbed to perform experiments involving emulated 802.11 a/b/g WLAN technology over wired networks for several types of scenarios. We shall illustrate this in the final part of the paper, with emphasis on networked robot emulation.

1. Introduction

As wireless network technologies, such as WLAN, are more and more widely used for communication, it is imperative to be able to study in controlled conditions the behavior of systems that use them.

Simulation and sometimes analytical modeling are currently used on an extensive scale to study WLAN systems. One of the most widely used tools are ns-2 [1], with its extensions to the wireless domain, and OPNET [2], a commercial simulator that focuses on wireless applications. However, since analytical modeling is based on abstract models, it doesn't allow objective measurements of system performance, but only rough predictions of general system behavior. Simulation may be closer to reality, but it is still relatively abstract, given that only models of real systems interact with each other in logical time during a simulation experiment.

If one is to perform tests using real WLAN equipment, several problems occur, since the wireless medium is difficult to control, and undesired

interferences can often perturb experimental results. This is equally true when performing in-house tests, or when using public wireless testbeds such as that at Emulab [3]. If mobility is also to be studied, controlled movement of wireless nodes has to be orchestrated, which is a cumbersome task with a high management overhead, or impossible if using a fixed testbed. Some ways to tackle this problem are the dense-grid approach of ORBIT [4], or the more realistic robot-based Mobile Emulab [5]. An additional concern for real tests is naturally that of the costs incurred.

A solution for studying wireless systems that gained popularity in recent years is the use of emulation. Note that in the context of Emulab, for example, the term "wireless emulation" is used to refer to performing tests on a wireless testbed (as opposed to a real-world wireless environment). However, in our view, wireless emulation is not possible if using real wireless devices, unless the wireless communication conditions can be fully controlled (for instance by using shielded RF chambers). Since full control of the conditions is an essential element of the concept of emulation, we prefer to only use "wireless emulation" to refer to the approach of running real traffic through computer models of wireless networks, with experiments being effectively performed over a wired network. This acceptance of the term "wireless emulation" will be used throughout the present paper.

WLAN emulation combines the advantages of real-world experiments and simulation. As in the case of real-world tests, WLAN emulation allows researchers to use the same applications that are used in practice by target users. Hence, their observations are readily applicable to practical situations. Moreover, as in the case of simulation, the wireless medium effects can be controlled through emulation so as to study user-defined scenarios with ease and free of interferences.

Previous approaches to WLAN emulation *per se* are oversimplified in general. Some emulators (e.g.,

Seawind [6] or Empower [7]), make it possible to introduce network layer effects, such as bandwidth limitation, delay, packet loss. However these effects are directly provided by the user who configures the emulator; this means that the relationship between these effects and reality becomes the user’s task, and is possibly not accurately defined. There are also attempts to develop emulators that recreate by themselves network conditions that correspond to real events, such as W-NINE [8] or the wireless-network emulation extension of SDNE [9]. Both these implementations start from a description of node positions and movements. However the accuracy of the conditions they recreate is relatively low because of the simplicity of the models used. For example W-NINE uses tables to associate IP throughput to received signal levels, loss probability is considered to be either 0 or 1, etc. A more realistic design is that of TWINE [10] that uses WLAN behavior computer models for real-time experiments. However TWINE needs to combine emulation with simulation in order to achieve scalability.

QOMET (the acronym stands for “Quality Of applications in transforMing network Environments Testbed”) is intended to be a versatile WLAN emulator that accurately reproduces in a wired network the WLAN conditions that correspond to a user-defined scenario. Our approach is inspired by [8] and [9] in the sense that it is a two-stage scenario-driven design. However, in order to achieve emulation realism, the models that we developed follow closely the IEEE specifications related to the 802.11 a/b/g standards [11-14]. Currently we further extend the functionality of QOMET to support emulation of active tag systems.

QOMET can be used either standalone, to study predefined user scenarios, or in a library form, integrated into more complex systems. Its modular architecture makes QOMET easily extendable to the emulation of other wireless environments. The novelty of our work also appears at several levels. It consists first of all in the quality degradation view we take on networks, and the emphasis we lay on emulation realism through the use of more accurate and yet tractable models that are described in Section 2. In Section 3 we present an overview of QOMET’s usage for practical experiments over wired networks, using the case study of networked robot emulation as an illustrative example. This is followed (Section 4) by experimental results obtained for such robot scenarios. The paper ends with a section of conclusions and future work, followed by acknowledgments and references.

2. QOMET architecture

An overall view on our approach and the basic WLAN model we used at that moment for emulating 802.11b networks in predefined static environments was previously presented in [15]. We provide here the design outline and a summary of the current features of QOMET. See Section 2.1 for a discussion of the improved models and added functionality that make it currently possible to run a wide range of experiments.

The scenario-driven architecture we propose 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. Next, the ΔQ description is converted into an emulator configuration that is used during the effective emulation process to replicate the user-defined scenario in a wired network (see Figure 1).

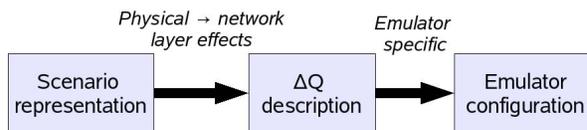


Figure 1. Two-stage scenario-driven approach to WLAN emulation

The following is a summary of the features of the current QOMET implementation. QOMET enables emulation of 802.11a/b/g WLANs using a model based on receiver-sensitivity thresholds (that are available from most manufacturers), and that includes effects of background noise. Support for 802.11g stations operating in compatibility mode (i.e., when in the presence of 802.11b stations) is also present. Our model takes into account interference between neighboring nodes, either resolved through the CSMA/CA (Collision Sense Multiple Access with Collision Avoidance) mechanism, or regarded as noise if the interfering signal is too weak to be detected. At the moment we extend the functionality of QOMET to support emulation of active tag systems.

The QOMET experiment scenario consists of an XML-based description of the WLAN nodes, topology elements, motion patterns, and communication environment properties. For WLAN nodes one can specify the WLAN adapters and their properties (transmitted power, antenna gain, etc.). The virtual environment topology and objects are defined by the user, and the communication environment can be computed dynamically, depending on nodes’ position, so as to increase the realism of emulation. The motion types that are currently available are: random-walk motion, linear motion, and circular motion.

We provide libraries to compute ΔQ parameters in real time, and also to configure a wired-network

emulator and run a QOMET emulation experiment. At the moment only the *dumynet* wired-network emulator [16] running on FreeBSD is supported, but the libraries can be extended to other operating systems. We also provide a timer library that ensures that *dumynet* configuration is changed at accurate time intervals during the emulation process.

2.1. WLAN emulation model

The WLAN emulation model in QOMET is an aggregation of several models used at the various steps of the conversion of a scenario representation to the network ΔQ description which is needed to recreate those scenario conditions. The initial models that we used in our previous implementation were presented in [14]. Transforming a real-world scenario description to meaningful 802.11 physical layer parameters is one of the most important steps in ensuring emulation realism. Therefore we reviewed and extended our initial models to follow more closely physical reality by using IEEE 802.11 specifications [11-14].

In order to calculate the effects of real-world scenario events on the physical layer of a WLAN station, it is necessary to determine first the signal attenuation due to the distance between communicating stations, interposed obstacles, etc. For this purpose we continue to use the reference log-distance path-loss model [17]. This model gives the received power, P_r , expressed in dBm (decibel-milliwatt), as function of the received power at the distance of 1 m, P_{r0} , and the distance, d , between receiver and transmitter:

$$P_r(d) = P_{r0} - 10 \cdot \alpha \cdot \log_{10}(d) - W + X_\sigma. \quad (1)$$

The communication environment is described by the parameters α (the path-loss coefficient), σ (shadowing parameter; the standard deviation of the zero-centered Gaussian distributed random variable X_σ), and W (wall attenuation; equal to zero for indoor environments). Values of these parameters for real environments can be obtained from studies such as [18, 19].

The approach we presented in [15] for computing the frame error rate (FER) corresponding to a received power strength had the disadvantage of requiring two separate models: one based on WLAN adapters receive sensitivities (which didn't include noise), and another one based on manufacturer-measured bit error rate (BER) versus signal-to-noise ratio (SNR) dependency. Since the latter data is only available from a small number of manufacturers (e.g., Intersil [20]), this limited the number of real adapters we could emulate.

As a consequence we decided to extend the first model so as to include noise. For this purpose we used

the information in [21] regarding the procedure of measuring receive sensitivity. Based on sensitivity values provided for each adapter by manufacturers, and a negative exponential model, we can determine the frame error rate (FER) corresponding to a received power strength. This P_r -threshold-based model computes FER as function of the adapter-specific receive sensitivity threshold for the current operating rate, S , the received power, P_r , the background noise, N , and the thermal noise N_{th} , as follows:

$$FER = FER_S \cdot e^{\gamma(S - (P_r - N) - N_{th})}, \quad (2)$$

where γ is a constant to be determined by calibration for each adapter (at the moment we use the default value 1), and FER_S is the frame error rate when P_r reaches the threshold S . According to [11] and [21], FER_S equals 0.08 for 1024-byte frames. The difference ($P_r - N$) in Equation (2) represents the Signal-to-Noise Ratio (SNR). N_{th} is the thermal noise at room temperature, and approximately equals -100 dBm if considering the 22 MHz bandwidth of 802.11b/g networks. Note that the FER from Equation (2) must be limited at 1, since the result represents a probability.

The background noise N in Equation (2) has two sources. One is the man-made electro-smog from devices operating in the same frequency band (for instance microwave ovens or cordless phones when speaking about the 2.4 GHz band of 802.11b/g). This parameter can be configured by the QOMET user. The other noise source are the signals generated by other WLAN devices. Such signals are treated as noise if the signal strength of the received power is inferior to the lowest receive-power sensitivity threshold of the emulated adapter. This interference type can cause frame errors at reception that will lead to retransmissions, hence delays, and possibly packet loss. In the opposite case the signal is considered to be properly detected by the receiver, and therefore induce transmission delays due to the use of the CSMA/CA mechanism. This is the second effect of interference, and it is dealt with separately at the data link layer to network layer conversion, as discussed in [15].

In order to determine the power of the received signal from interfering WLAN sources we use first Equation (1). However, interfering sources can transmit on different channels that the channel on which the receiver listens. In this case the inter-channel attenuation must be computed. For this purpose we used the requirements in the IEEE 802.11 specifications regarding the transmit spectral mask [10, 11]. Figure 2 shows our model's attenuation characteristics depending on channel distance (i.e., the absolute difference between channel ids). Note that in 802.11a networks effectively-used channels are at a

minimum distance of 4, which reduces considerably interferences, since attenuation will always exceed 30 dB. For 802.11b/g WLANs, distance between different channels can be 1, making it a more difficult situation.

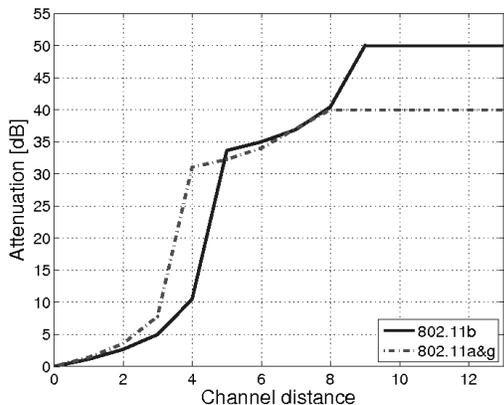


Figure 2. Inter-channel attenuation model

2.2. Discussion

The modular architecture of QOMET allows for continuous improvements and extensions. Since our initial WLAN model [15] didn't require any changes for the physical to data link layer, or data link layer to network layer conversions, those modules remained unchanged when improving QOMET.

Although WLANs are at the moment probably the most used wireless technology for IP networks, other technologies started being used in dedicated applications. Such an example is the wireless communication between RFID devices. In this context we currently implement an extension to QOMET dedicated to emulation of active tag communication for pedestrian localization applications. This extension is facilitated considerably by the architecture of QOMET, since we are able to reuse several of its features (e.g., scenario description, motion emulation), and only need to create the appropriate modules for conversions of parameters between the different emulation layers.

3. Networked-robot emulation

Various experiments were performed so far using QOMET. VoIP experiments with reference to rescue worker communication in emergency conditions were presented in [14]. The emulation of ubiquitous home environments with application to room temperature control was described in [24, 25]. Currently we perform active tag emulation experiments for pedestrian localization applications. All these experiments show how convenient it is to use QOMET for analyzing various real-world scenarios. We present here as a case study the most complex experiment to

date, which involved the emulation of more than 100 WLAN nodes, each representing a networked robot.

Experiments for evaluating such research are difficult to perform, since the cost of real autonomous robots is high. To make possible such tests we propose to use emulation for the evaluation of large-scale autonomous networked robot systems. In this approach, the robot behavior implementation – including aspects of motion planning and communication with other robots – is run on a standard PC. Various modules are connected to this implementation with the goal of allowing the implementation to “behave” as it would do in a real-world environment (see Figure 3). The WLAN emulation module ensures that the robot can communicate with other robots using a real protocol implementation in conditions similar to those that would occur in a real WLAN environment. A GPS emulation module provides the robot coordinates in the virtual space to the robot behavior implementation, exactly as a GPS card would do so in a real environment. The robot can be equipped with various other sensors, each requiring a specific emulator to be present. For instance, we can use a thermal field emulation module to emulate temperature variation in the virtual space; to enable the use of visual sensors, a visual environment emulation module is required, etc.

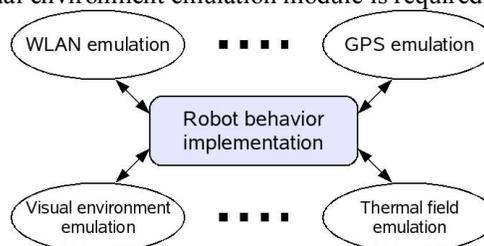


Figure 3. Robot-system emulation architecture

4. Experimental results

In order to perform QOMET-based experiments we used StarBED, the large-scale network experiment environment at the National Institute of Information and Communications Technology, Hokuriku Research Center, in Ishikawa, Japan, and the experiment-support software RUNE [27]. The motion planning algorithm whose performance was tested is based on a Probabilistic Roadmap Planner algorithm, and was adapted to dynamic environments as described in [28].

Although we carried out experiments with more than 100 emulated robots, we shall present here a smaller scale example. The virtual space in which the robots are considered to be located is the basement of a building in which 10 robots are assigned different tasks at various locations. They must travel from the initial

starting position to their individual destination by making sure no collisions occur with the obstacles present (building pillars) or with the other robots. The virtual topology is depicted in Figure 5.

In the emulated environment robots are initially located on a 15x15 meter grid. For Robot $\#n$, its source and destination are denoted by $S\#n$ and $D\#n$, respectively. Robots are considered to have a radius of 1 m and a constant speed of 0.5 m/s. Obstacles also have a radius of 1 m; they are represented by dark discs in Figure 4. The robots are each assigned a priority. A robot's priority is equal to its id; therefore the robot with the highest priority is Robot #10. The maximum distance from which robots can visually detect obstacles and other robots (the range of visual sensors, represented by an omni-directional camera) is 10 m.

Regarding the WLAN emulation, we considered that robots are equipped with 802.11b transceivers. In the experiment the QOMET library computes in real time the characteristics of WLAN communication at each processing step (currently 250 ms). Then actual network quality degradation is enforced using *dummysnet* [16], by applying to the network the previously calculated WLAN characteristics. This takes place in real time on the testbed. The emulated WLAN environment had the following main parameter values: $\alpha = 5.6$, $\sigma = 3.1$, $N = -100$. This corresponds to bad reception conditions, and results in a communication range of approximately 18 m.

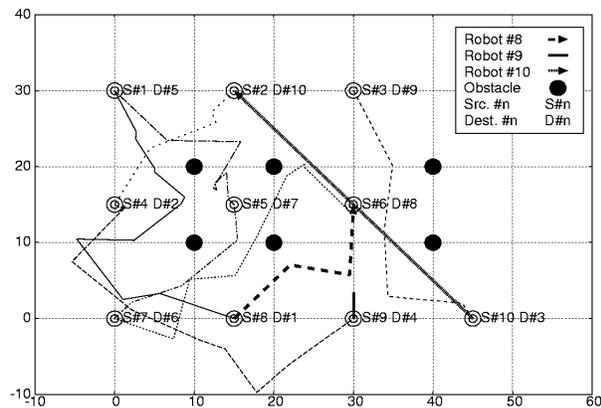


Figure 4. Robot initial positions and traveled trajectories

In Figure 4, for clarity reasons, we emphasized only the trajectories of three robots, those with ids 8, 9, and 10. Analyzing the trajectories of Robot #8 and Robot #10 one can notice that their initial trajectories starting at $S\#8$ and $S\#10$, respectively, lead them to collision at position (30,15). However, when Robot #8 approaches the point of coordinates (22,7) it enters in the communication range of Robot #10. As a consequence, they are able to exchange information about each

other's trajectory. This makes Robot #8 (the lower priority robot) to re-plan its trajectory, so as to avoid the collision. The higher priority robot, Robot #10, continues moving on its initial trajectory all the way to its destination at (15,30).

Another remark about Figure 5 concerns Robot #9. After leaving from the starting point $S\#9$, located at (30,0), the robot suddenly halts after about 3 m. This is caused by the fact that its trajectory interfered with the trajectories of other robots, and the robot must stop to "think" in order to find a new trajectory; this triggered an implementation bug. We managed to identify the cause of this problem, and fixed it. When many robots interact with each other in real time and in complex scenarios, implementation issues that would otherwise go unnoticed can be revealed.

We shall not discuss here the trajectories of all robots, but one can see in Figure 4 how the obstacles and other robots' trajectories influence the path of each robot. For example Robot #3, starting at (30,30) needs to change its trajectory from the initially planned one when it observes the obstacle located at (40,10).

5. Conclusions

In this paper we presented the improved multi-purpose wireless network emulator that we develop, QOMET. Our approach makes it possible to transform a user-meaningful real-world representation of a wireless network environment (termed "scenario representation") into a network quality degradation description (the " ΔQ description"). This description is sufficient to subsequently configure a wired-network emulator and effectively reproduce an environment that corresponds accurately at network level to the emulated WLAN scenario. Currently QOMET provides mainly support for 802.11a/b/g WLAN networks in realistic topologies in which the effects of the dynamic surrounding environment and inter-node interference issues are taken into account.

The assessment of networked systems through real-time emulation makes it possible for users to objectively evaluate software implementations of such networked systems under realistic conditions. These software implementations are afterwards readily deployable on real target systems, therefore their evaluation is an essential tool in reducing the cost of developing networked systems.

We illustrated the practical use of our WLAN emulation approach through a detailed analysis of a case study of robot motion planning, for an algorithm based on Probability Roadmap Planner, and adapted to dynamic environment conditions. The experimental results demonstrated how one can assess the properties

of such an algorithm through emulation in complex scenarios. Our experiments helped us identify and correct several robot motion planning implementation issues.

As future work regarding the development of QOMET we intend to add features to it so as to make it possible for users to define scenarios in a more realistic way, including streets and buildings. More advanced ad hoc network features, such as routing will also be included in future versions.

6. Acknowledgments

We would like to acknowledge the contributions of Khin Thida Latt, Toshiyuki Miyachi, Saber Zrelli, Assoc. Satoshi Uda, and Assoc. Ken-ichi Chinen.

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.

7. References

- [1] University of Southern California, Information Sciences Institute, "The Network Simulator - ns-2", <http://www.isi.edu/nsnam/ns/>.
- [2] OPNET Technologies, Inc., "OPNET network simulator", <http://www.opnet.com/>.
- [3] University of Utah, School of Computing, "Emulab - Network Emulation Testbed", <http://www.emulab.net/>.
- [4] Rutgers University, Wireless Information Network Laboratory, "ORBIT - Wireless Network Testbed", <http://www.orbit-lab.org/>.
- [5] D. Johnson, T. Stack, R. Fish, D. Montralio 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, pp.1-12.
- [6] M. Kojo, A. Gurtov, J. Manner, P. Sarolahti, T. Alanko, K. Raatikainen, "Seawind: a Wireless Network Emulator", *Proc. of MMB 2001*, Aachen, Germany, September 2001.
- [7] P. Zheng, L.M. Ni, "EMPOWER: A Network Emulator for Wireline and Wireless Networks", *Proc. of IEEE Infocom 2003*, San Francisco, U.S.A, April 2003.
- [8] T. Perennou, E. Conchon, L. Dairaine, M. Diazet, "Two-Stage Wireless Network Emulation", *Proc. of WCC2004*, Toulouse, France, August 2004.
- [9] M. Bateman, C. Allison, and A. Ruddle, "A Scenario Driven Emulator for Wireless, Fixed and Ad Hoc networks", *Proc. of PGNet2003*, Liverpool, U.K., June 2003, pp. 273-278.
- [10] 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.
- [11] ANSI/IEEE Standard 802.11, 1999 Edition (Reaffirmed 2003), <http://standards.ieee.org/getieee802/download/802.11-1999.pdf>.
- [12] ANSI/IEEE Standard 802.11a, 1999 Edition, <http://standards.ieee.org/getieee802/download/802.11a-1999.pdf>.
- [13] ANSI/IEEE Standard 802.11b, 1999 Edition, <http://standards.ieee.org/getieee802/download/802.11b-1999.pdf>.
- [14] ANSI/IEEE Standard 802.11g, 2003 Edition, <http://standards.ieee.org/getieee802/download/802.11g-2003.pdf>.
- [15] 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.
- [16] L. Rizzo, "Dummysnet FreeBSD network emulator", http://info.iit.unipi.it/~luigi/ip_dummysnet.
- [17] T.S. Rappaport, "Wireless Communications: Principles and Practice", *Prentice Hall PTR*, 2nd edition, 2002.
- [18] Defence Information Systems Agency (DISA), "WLAN Site Survey Addendum to the Wireless Security Technical Implementation Guide", *U.S.A. Department of Defense*, October 2005.
- [19] D. B. Faria, "Modelling Signal Attenuation in IEEE 802.11 Wireless LANs - Vol. 1", *Technical Report TR-KP06-0118*, Kiwi Project, Stanford University, July 2005.
- [20] Intersil, "HFA3861B: Direct Sequence Spread Spectrum Baseband Processor", *Intersil data sheet*, February 2002.
- [21] Intersil, "Measurement of WLAN Receiver Sensitivity", *Intersil technical brief*, February 2000.
- [22] A. Kameron, L. Monteban, "WaveLAN-II: A high-performance wireless LAN for the unlicensed band", *Bell Lab Technical Journal*, 1997, pp. 118-133.
- [23] P. Gupta, P.R. Kumar, "Capacity of wireless networks", *technical report*, University of Illinois, U.S.A, 1999.
- [24] 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 (CD proceedings).
- [25] J. Nakata, R. Beuran, T. Miyachi, K. Chinen, S. Uda, K. Masui, Y. Tan, Y. Shinoda, "StarBED2: Testbed for Networked Sensing Systems", 4th International Conference on Networked Sensing Systems (INSS07), Braunschweig, Germany, June 6-8, 2007, pp. 142-145.
- [26] D. Hsu, J.-C. Latombe, R. Motwani, "Path Planning in Expansive Configuration Spaces", *International Journal of Computer Geometry and Applications*, Vol. 9, No. 4/5, 1997, pp. 495-512.
- [27] 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.
- [28] T. Okada, J. Nakata, R. Beuran, 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.