Performance Assessment of Ubiquitous Networked Systems

Razvan Beuran[†,*], Junya Nakata[*,†], Takashi Okada[*], Toshiyuki Miyachi[*,†], Ken-ichi Chinen[*,†], 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@jaist.ac.jp

Abstract. In this paper we present the methodology we propose for the performance assessment of ubiquitous networked systems. Our methodology relies on emulation of both the ubiquitous nodes and the network interconnections between them. This approach makes it possible to conveniently perform experiments with a large number of nodes in controlled network conditions. We illustrate our approach with experimental results obtained on a large-scale testbed, StarBED2. Experiments are performed using the support software RUNE, and the WLAN emulator QOMET.

Keywords: Ubiquitous networked systems, home networks, WLAN emulation, assisted living.

1 Introduction

Pervasive computing is one of the most promising ways to improve quality of life in the near future. Using interconnected ubiquitous systems it becomes possible to embed intelligence in all the devices of a home environment, therefore opening the way for a better management of the properties of the home environment and assisted living. Deploying ubiquitous networked systems in a home environment raises however numerous questions regarding the performance of such systems, and that of the applications they support, the safety of the deployment, etc. Therefore it is necessary to test such systems in advance so as to validate their behavior. Although real-world tests are undoubtedly mandatory, this is particularly decisive solely in the final stages of the development cycle; using *only* real-world tests would be very costly, and is practically impossible at the very beginning of development.

The methodology we propose relies on emulation of the ubiquitous nodes in a test environment that also supports integration of real ubiquitous computing systems. In this way it is possible to study how devices under design interact with existing real systems before they are effectively produced. The approach that we propose makes it possible to conveniently perform experiments with a large number of ubiquitous nodes in network conditions that are also controlled through the use of emulation. This allows studying system behavior, and assessing objectively ubiquitous networked system performance in quasi-real conditions.

Software simulators let users make experiments using abstracted network elements on one computer. It is the most popular approach to evaluate network technologies, and ns-2 [1] is a well-known simulator example. The cost of making experiments with software simulators is low, but target systems must be described under specific modeling schemes. Software simulators run in logical time, which is a significant drawback if one tries to simulate target systems under a realistic environment, where nontrivial aggregation of complex network services comes into play.

Small-scale testbeds based on actual nodes consist of a few dozen network equipments and computers. Their use incurs higher costs, since one has to prepare physical nodes needed by experiments, connect and configure them as necessary. Such laboratory-level testbeds are more realistic than simulations, but from the point of view of scale they are not suited for whole ubiquitous network tests.

There are already a number of implementations of emulators and testbeds for ubiquitous networks. TOSSIM [2] is a TinyOS [3] simulator implemented in Java which aims to simulate TinyOS applications in a virtual environment. ATEMU [4] also is able to emulate TinyOS applications, and has a flexible architecture to support other platforms. However none of these tools provides the user with a method to describe surrounding environments, or an interface with real nodes, which would allow tested systems to interact with real environments. This is of great use, especially in the last phases of development, and our system addresses these needs through the software tool nicknamed RUNE (Real-time Ubiquitous Network Emulation environment). In order to enable home network emulation, we take into account current *de facto* standards [5] and recommendations [6] (see [7] for details). For a discussion of smart homes and assisted living see [8].

A survey of existing WLAN-related real-world and simulation testbeds is available in [9]. Previous approaches to WLAN emulation are oversimplified in general. Some emulators, for instance Seawind [10] or Empower [11], introduce network layer effects directly provided by the user, such as bandwidth limitation, delay, packet loss; this means the link between these effects and reality is user's task, and possibly not accurately defined. W-NINE [12] and the wireless-network emulation extension of SDNE [13] try to reproduce this link. However the accuracy of the conditions they recreate is relatively low because of the simplicity of the models used. MobiNet [14] is another wireless network emulator; it focuses on ad hoc routing but the detail level of wireless communication emulation itself is reduced. Our WLAN emulation system, QOMET (Quality Of transforMing Environments Testbed), addresses the requirements in [10] through a scenario-driven two-stage approach inspired by [12] and [13]. The novelty of QOMET consists in the quality degradation view embedded in the design, the emphasis on emulation realism through the use of detailed models, as well as the use of a large-scale testbed for experiments.

We performed tests on the large-scale network experiment environment, StarBED, and its successor StarBED2 [15], at the Hokuriku Research Center of the National Institute of Information and Communications Technology (NICT) in Ishikawa, Japan. This experiment environment is a cluster-based testbed currently employing about 700 PCs. By combining the functionality of ubiquitous network emulation of RUNE with that of WLAN emulation of QOMET, and executing the experiments on StarBED, we have created one of the first large-scale environments that allows the emulation and performance assessment of large ubiquitous networked systems.

2 Methodology Overview

Our methodology makes it possible for researchers to assess system performance in the context of ubiquitous networks. This can be done at several levels: performance of network systems (bandwidth, delay, packet loss), performance of network applications (e.g., file transfer throughput), performance of ubiquitous system from a user perspective (e.g., the range of temperature variation, as in the experiment we'll present in section 3). Next we'll discuss the framework that makes it possible to perform such experiments in controllable conditions.

Given the size of the devices used in ubiquitous computing and their number, one of the most used types of networks used to interconnect them is the wireless network. Our wireless network emulator, QOMET, supports so far emulation of IEEE 802.11 WLAN in a two-stage scenario-driven approach [16]. QOMET gives us direct control over the network conditions under which we assess application performance. Next we'll describe the tool we use for experiment integration for ubiquitous system emulation, RUNE, and some possible applications of our system.

Ubiquitous networks have different properties than computer networks in many aspects, such as: high node variability and network media variety, huge number of nodes, etc. The required functionality for ubiquitous network testbeds is as follows: (i) emulate the surrounding environments, and provide the interface between the emulated nodes and these environments; (ii) provide an interface between emulated and physical systems, so as to make possible virtual and real mixed setups; (iii) support the numerous nodes of ubiquitous networks; (iv) emulate the various architectures of nodes and networks that form typical heterogeneous networks; (v) provide a multi-level emulation layer to users.

In order to meet these requirements on StarBED2, the experiment-support software RUNE is being developed. RUNE provides an API set which controls experiment environments. The fundamental goal of RUNE is to implement an environment in which a number of *spaces* that emulate each experiment target can work on either single or multiple nodes. RUNE provides a reasonably abstracted interface for easily implementing emulation targets as spaces without much concern about the interaction between emulation nodes. RUNE has the following roles: (i) experiment environment setup/cleanup and progress management; (ii) procedure invocation; (iii) interaction between spaces; (iv) time synchronization; (v) mutual exclusion.

Figure 1 shows the structure of an experiment implemented using RUNE. The RUNE master manages the configuration of each experiment, and controls the progress of the experiment. The execution of spaces is initiated by the RUNE master via RUNE managers. RUNE managers are deployed on every emulation node and mediate communication between them through objects called *conduits*. Spaces on emulation nodes are shared objects, loaded dynamically by RUNE managers.

The emulation process performed by RUNE takes place as follows. First of all RUNE master is compiled with the experiment definition file which includes the information regarding spaces and conduits. Then the RUNE master sends the instruction "attach process" to the RUNE manager executed on each node. A space then returns its entry point information to the RUNE manager, which includes pointers to the available functions. When the RUNE manager notifies the RUNE master of the completion of the "attach" process, the latter indicates the initialization process of all spaces to the RUNE managers on each node. After the initialization of all spaces is finished, the RUNE master starts the iterated invocation of the "step" symbol. Spaces execute the emulation step-by-step on request from the RUNE manager, and inform the RUNE master of the execution status. At the end of the experiment the RUNE master starts the finalization process by notifying all nodes. Spaces release then the work area allocated in the initialization process.

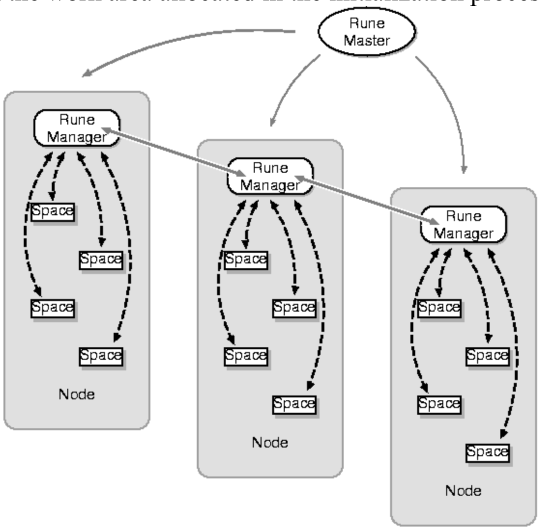


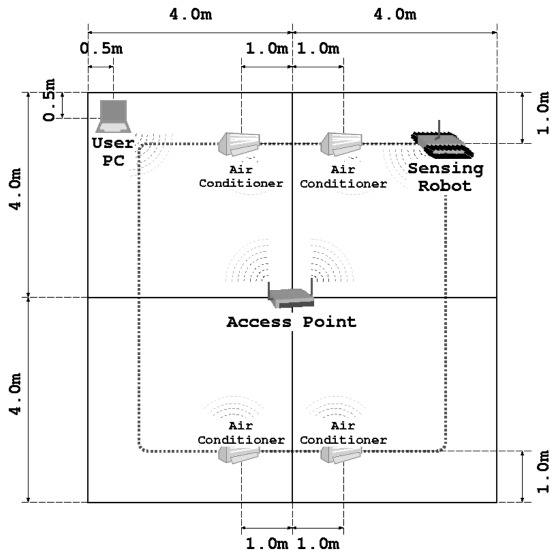
Fig. 1. Structure of experiments implemented using RUNE

A key feature of RUNE are the spaces it can emulate. For example by emulating heat propagation it is possible to perform experiments using temperature regulation in a house environment, such as those described in the next section. QOMET is the system that allows reproducing the wireless connection between devices that are emulated on StarBED2. At the moment QOMET only supports emulation of 802.11 WLAN communication, including features such as rate adaptation, but other wireless technologies more specific to sensor networks will be added in the future (e.g., Bluetooth and Zigbee). QOMET also emulates the motion of the wireless devices and computes the WLAN communication conditions associated to this motion.

The range of application for these two tools is very large, both in indoor and outdoor environments. At the moment we pay particular attention to emulation of robots that cooperate in order to achieve various tasks using wireless communication. Such robots could be used to facilitate human life, reduce cost of activities, or to avoid danger. Autonomous robots can perform various activities to support the human rescue teams. In office buildings or homes, robots can automatically clean areas, monitor and regulate environment conditions (if equipped with temperature, humidity, luminosity sensors), etc. For motion-planning purposes robots use mobile ad hoc networks and share the information of trajectories or maps from visual sensors. The implementation of such robot systems in reality incurs high costs since they need to be equipped with sensors, motors, WLAN cards. As these costs become prohibitive when testing systems with hundreds of robots, one can employ emulation environments such as ours in the initial design and implementation phases.

3 Experimental Results

In this section we show some illustrative results that we obtained on StarBED2, using RUNE as experiment-support software, and QOMET to emulate the WLAN communication environment. The experiments will show how one can evaluate network application performance (file transfer) and user-perceived performance of networked systems (temperature variation range) for a home ubiquitous network. The emulated topology is presented in Figure 2. In this environment each room's air conditioner (AC) is wire-connected to a home controller operating based on information sent from a robot that monitors environment conditions. The robot is equipped with a heat sensor, but other factors may be controlled, such as light level.



Parameter	*Time [min]*	*Value*
α	0 – 5	4.02
σ	0 – 5	7.36 dB
Usual noise	0–0.5; 1.5–2.5; 3.5–5	−100 dBm
Interference 1	0.5 – 1.5	−50 dBm
Interference 2	2.5 – 3.5	−60 dBm
Packet size: *Robot to AC*	0 – 5	64 bytes
Packet size: *User to Server*	0 – 5	1500 bytes
Packet size: *Server to User*	0 – 5	64 bytes

Fig. 2. Emulated topology and parameters used in the experiment

The temperature information is sent from the robot to the home controller using WLAN, and then communicated to the AC. An access point located in the house provides WLAN capability. The WLAN is used by a human user, and for simplicity reasons we assumed this user is performing a file transfer for the duration of the experiment; this is emulated using *netperf* [17]. In a home environment, microwave ovens or cordless phones interfere with WLAN communication. We assumed that during the experiment such interference, will occur twice after the experiment is started, and last each time for one minute. QOMET computes the characteristics of WLAN communication using the properties shown in Figure 2. Actual network quality degradation is enforced using *dummynet* [18] in real time on the testbed. The robot moves counter-clockwise on the trajectory depicted in Figure 2, from the indicated initial position with a speed of 0.3 m/s. The robot pauses in each room for 10 s to determine the temperature, then sends a packet to the AC controller and resumes its motion. The AC heats up the ambient air depending whether the temperature sent by the robot is below the intended temperature (23 degrees Celsius) or above. AC temperature is 80 degrees Celsius; in the absence of information, the air conditioner keeps its current operation status. Temperature variation in the room is calculated using heat propagation theory every 25 ms; room wall temperature is considered constant and equal to 15 degrees Celsius.

The spaces corresponding to equipments (air conditioners, robot, user PC, and WLAN access point) are executed on separate nodes. A space that emulates heat transfer in the room is also deployed on a dedicated node. QOMET does not appear as one space, but multiple spaces distributed on every node on which WLAN equipments are emulated. The spaces are called "dnconf" (*dummynet* configuration), and configure the *dummynet* pipe dynamically during the experiment, by applying the WLAN properties pre-calculated by QOMET repetitively on each node every 500 ms. Every node in StarBED has at least two interfaces connected to different network segments, the control network and the experiment network. The control network is used for communication between the RUNE master and managers. The experiment network is used for communication between spaces through the emulated WLAN.

Figure 3 shows the available bandwidth of the WLAN connection between the heat sensor and the access point. As it can be seen, network degradation occurs when the noise level raises, and it is more severe for the higher-noise period (from 30 to 90 s).

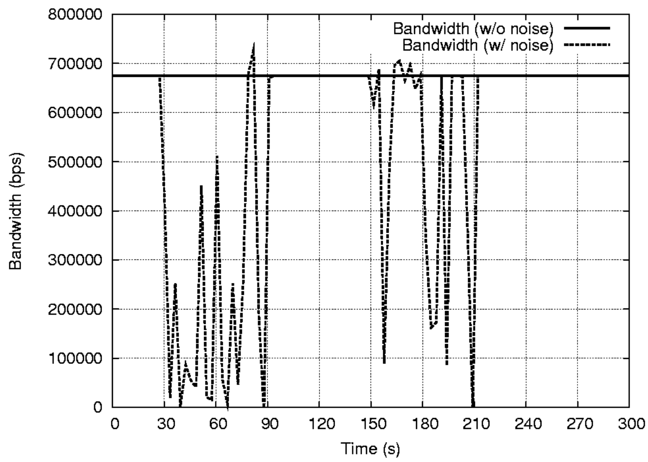


Fig. 3. Bandwidth of the WLAN connection between robot and the access point

Figure 4 shows the variation of the attained temperature at the user location, both in the case when no interference noise is present, and in the case with noise. Without noise this variation is bounded to a 3-4 degrees difference compared to the intended temperature of 23 degrees Celsius. However, when noise presence is also emulated, the attained temperatures exceeds 28 degrees Celsius, therefore an error of 5 degrees. This temperature variation is an objective measure of the user satisfaction; hence our system can be used to assess in a quantifiable manner the effects of WLAN interference on user satisfaction in such a scenario. Concerning the file transfer test, losses have as expected a significant effect on TCP/IP performance. The throughput observed without interferences was of about 6 Mbps, and it dropped to around 3 Mbps when interference noise was considered.

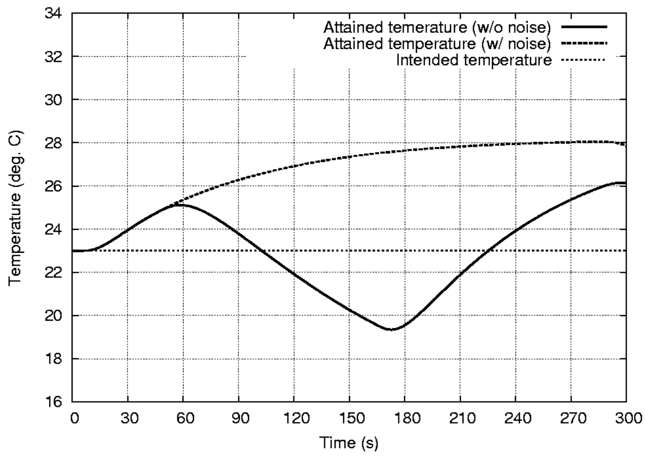


Fig. 4. Temperature variation at the user location, with and without interference noise

4 Conclusions

The use of ubiquitous networks in home environments is one of the most promising solutions for improving the quality of life. However, ubiquitous and sensor networks are large and complex, and it is difficult to predict their behavior. This is nevertheless mandatory when deploying them in close proximity to humans. Assessing performance in realistic environments before deployment is also a key element in reducing development costs. In order to assess the behavior of such technologies and evaluate implementations for these networks, we proceeded to design and implement a suitable testbed.

StarBED is a large-scale network testbed on which users can perform experiments with time and space sharing at reduced cost. By adding some components on top of StarBED, its present successor, StarBED2, can now be used for assessing ubiquitous network behavior too. The experiment-support software RUNE is intended to assist emulation execution of ubiquitous networks with a large number of nodes on multiple PCs of StarBED2. Emulation can be done on StarBED2 by implementing nodes, networks, and surrounding environments as *spaces*. They communicate with each other through *conduits* during the emulation process. This emulation is done with real-time constraints so that our system can work cooperatively with real components.

To illustrate the use of the experimental platform we provided some experimental results that show some of the capabilities of StarBED2 and RUNE that make it possible to study home environments and assisted living technologies. The example emulated a home environment in which a robot controls temperature conditions, and in which users are also actively using the WLAN, all in presence of interference from home appliances. WLAN communication is emulated using the software tool QOMET. We showed the different effects that interferences in WLAN environments can have on application performance, depending on each application characteristics.

Development of StarBED2 is still in progress. A high priority is given to adding support for processor and middleware emulation. In order to emulate more complex sensor networks, other environment spaces will be added as well, such as acoustic and visual environments, for example. Additional desired features for RUNE are, for example, more strict synchronization between spaces and mutual exclusion.

References

1. The VINT Project: The ns Manual. http://www.isi.edu/nsnam/ns/nsdocumentation.html
2. Levis, P., Lee, N., Welsh, M., Culler, D.: TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. In: Proc. of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003) (2003)
3. TinyOS embedded networked sensor operating system. http://www.tinyos.net
4. Polley, J., Blazakis, D., McGee, J., Rusk, D., Baras, J.S.: 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) (2004)
5. ECHONET. http://www.echonet.gr.jp/english/index.htm
6. Tan, Y. (ed.): Ubiquitous technology series: Home networks and Smart home appliances. Ohmsha (2004)
7. ITU-T Recommendation J.190: Architecture of MediaHomeNet that supports cable-based services (2002)
8. Wang, Q., Shin, W., Liu, X., Zeng, Z., Oh, C., Alshebli, B.K., Caccamo, M., Gunter, C.A., Gunter, E., Hou, J., Karahalios, K., Sha, L.: I-Living: An Open System Architecture for Assisted Living. In: Proc. of SMC 2006, Taipei, Taiwan (October 2006)
9. Kropff, M., Krop, T., Hollick, M., Mogre, P.S., Steinmetz, R.: A Survey of Real-World and Emulation Testbeds for Mobile Ad hoc Network. In: Proc. of TridentCom2006, Barcelona, Spain (March 2006)
10. Kojo, M., Gurtov, A., Manner, J., Sarolahti, P., Alanko, T., Raatikainen, K.: Seawind: a Wireless Network Emulator. In: Proc. of 11th Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), Aachen, Germany (September 2001)
11. Zheng, P., Ni, L.M.: EMPOWER: A Network Emulator for Wireline and Wireless Networks. In: Proc. of IEEE Infocom 2003, San Francisco, U.S.A. (April 2003)
12. Perennou, T., Conchon, E., Dairaine, L., Diazet, M.: Two-Stage Wireless Network Emulation. In: Proc. of WCC2004, Toulouse, France (August 2004)
13. Bateman, M., Allison, C., Ruddle, A.: A Scenario Driven Emulator for Wireless, Fixed and Ad Hoc networks. In: Proc. of PGNet2003, Liverpool, U.K. (June 2003) 273-278
14. Mahadevan, P., Rodriguez, A., Becker, D., Vahdat, A.: MobiNet: A Scalable Emulation Infrastructure for Ad hoc and Wireless Networks. In: Proc. of the International Workshop on Wireless Traffic Measurements and Modeling (WiTeMe 2005) (2005)
15. Miyachi, T., Chinen, K., Shinoda, Y.: Automatic Configuration and Execution of Internet Experiments on an Actual Node-based Testbed. In: Proc. of Tridentcom2005, Trento, Italy (February 2005) 274-282
16. Beuran, R., Chinen, K., Latt, K.T., Miyachi, T., Nakata, J., Nguyen, L.T., Shinoda, Y., Tan, Y.: Application Performance Assessment on Wireless Ad Hoc Networks. In: Proc. of AINTEC 2006, LNCS 4311, Bangkok, Thailand, November 28-30, 2006, pp. 128-138.
17. Netperf: A Network Performance Benchmark. http://www.netperf.org/
18. Rizzo, L.: Dummynet: a simple approach to the evaluation of network protocols. ACM Computer Communication Review 27 (1997) 31–41