

Large-scale Simulation Method of Mobile Robots

Takashi Okada Junya Nakata Razvan Beuran Yasuo Tan Yoichi Shinoda

Japan Advanced Institute of Science and Technology
National Institute of Information and Communications and Technology

Abstract

Recently mobile robots act in various situations such as disaster areas, office buildings, factories and homes. When these new mobile robots are released, they should be confirmed to work correctly and safety from evaluation. In this paper, we propose large-scale simulation environment of mobile robots on StarBED which is a large-scale networked testbed. By using the simulation environment, we could confirm that four hundreds of mobile robots act in real-time.

1. Introduction

1.1. Overview

The evaluation method is one of the big issues of the research of mobile robots. When mobile robots with new architecture are released, researchers have to evaluate them whether the mobile robots work correctly and safety.

One of the evaluation methods is to prepare real mobile robots on real environment. The result from this method is real. But they need to prepare enough number of these real mobile robots and act them on real environment. And if they try to do large-scale experiment, it takes much cost.

Second are simulation. This method may not take much cost. But the reality of the experimental result depends on the accuracy of the modeling method which is adopted in the simulation.

Our approach mixes above of them. It means that we use partially real hardware and the other parts are simulated. The processor, memory, disk and network are real hardware. The other special hardware of mobile robot such as motors, sensors devices and experimental environment are virtually simulated.

We use StarBED [1] as the experimental environment which is large-scale networked testbed placed at Ishikawa Japan. It enables to construct various virtual network topology by configuring the intelligent switches.

Large number of simulated mobile robots are executed on StarBED with our simulation method. Finally we could

execute four hundreds number of mobile robots in real-time.

In this paper, we propose the simulation method to model virtual mobile robots on our experimental environment. Next, we summarize the considerations for enabling to execute large number of mobile robots in real-time experiment. Finally we conclude with future works.

1.2. Related work

As we mentioned in previous section, there are some simulation method of mobile robots. Webots [2] is a simulation software which co-developed by the Swiss Federal Institute of Technology in Lausanne, Switzerland. Eye-Sim [3] is a multiple mobile robot simulator that allows experimenting with the same unchanged EyeBot programs that run on the real robots. These software simulator models the components of mobile robots, network connections, motion, time scheduling and surrounding environment. It does not take much cost because of these prepared components. But the accuracy of the result of experiments depends on these modeling.

RobotCup [4] is one of the major competition of mobile robots. For example in the soccer competition, the robot teams are entered into a competition to play soccer against opponent teams fielded by other research groups. Hsu [5] made experiments on his research team testbed. These robots move frictionlessly on an air bearing on a three meters times four meters table. ARL (Stanford Aerospace Robotics Laboratory) [6] makes various experiments of real robots. This real experimental environment takes much cost and time to manage real robots and experiments. If researchers want to test large number of robots, it is hard to make ready all robot hardware including processing power and connection to wireless network.

2. Simulation approach

2.1. System overview

As we mentioned in previous section, software simulator and real experimental environment have the merits and de-

merits. Our simulation approach is emulation which mixes simulation models and real hardware. The processor, memory, disk and network are real hardware. The other special hardware of mobile robot such as motors, sensors devices and experimental environment are virtually simulated.

When we simulate large-scale mobile robots, StarBED [1] which is large-scale networked testbed is used. It has about eight hundreds of PCs and they are connected through intelligent switches. If mobile robots have wireless network interfaces, it can create any virtual topology by configuring VLAN. Mobile robots are assumed as these nodes on StarBED, the application which works on these mobile robots are executed.

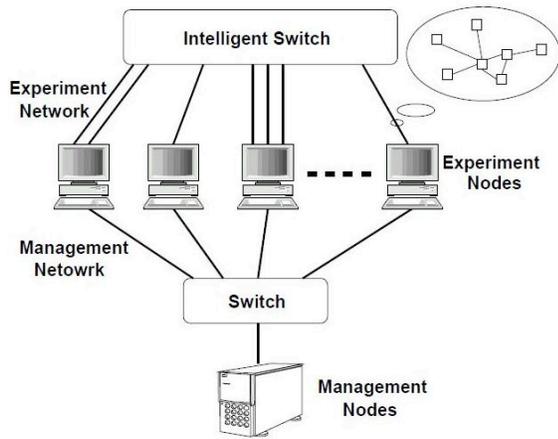


Figure 1. StarBED: large-scale networked testbed

But since these nodes on StarBED don't have any driving motors or sensor devices, the simulation specific hardware are modeled. The simulation environment is also modeled on our simulation environment such as experimental field, objects in the field, location information of each robots and other physical information.

2.2. The modeling of virtual world

To simulate the virtual mobile robots, we model various components of mobile robots and surrounding environment. In our experiment, mobile robots are assumed that they are equipped with GPS, WLAN and visual sensor. They can know where they are from GPS. By using WLAN device, mobile robots can communicate each other. And they can sense closed objects with visual sensor. Our system provides the API to access these virtual devices. Figure 2 describes it.

To synchronize between these virtual devices and virtual world, our system has management structure of physical

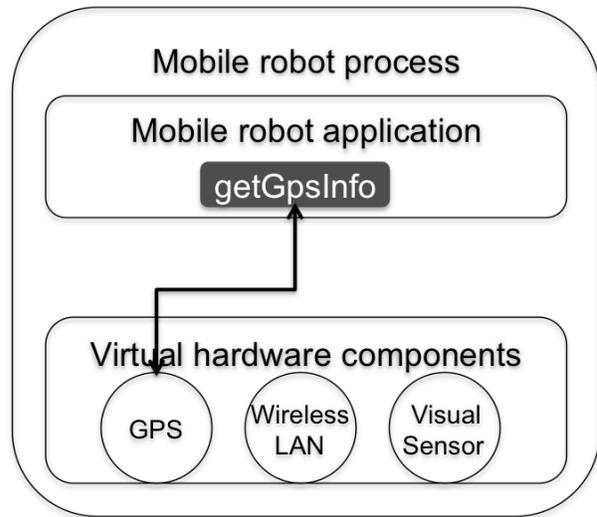


Figure 2. Example for GPS information access

events such as location changes, wireless radio signals, the changes of the images from visual sensors and so on. These virtual devices keep on communicating with the management node named Map Manager [7]. Figure 3 shows the overall architecture of our mobile robots simulation. They update their inner state and information from the communication with Map Manager.

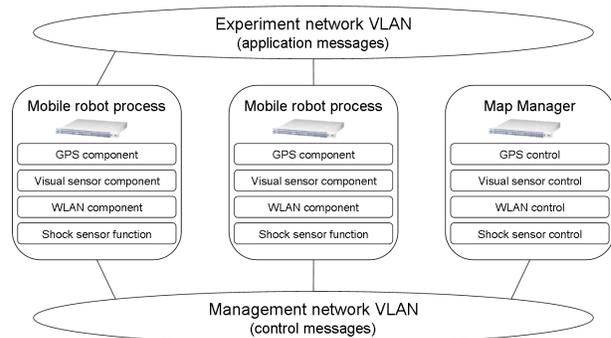


Figure 3. Overall architecture of simulation system

2.3. Network configuration

Figure 3 describes two VLAN on our system. One is the management network which mobile robots process and Map Manager communicate. They send the messages when their hardware has any events. If they receive the messages, they update their virtual world information. Map Manager

handles all of these information, distributes the events of virtual world.

Second are the experiment network which mobile robots communicate with each other robots. As this network is assumed as wireless network in our experiment, we simulate wireless network on cable network of StarBED. This is mentioned in next section.

2.4. The supporting softwares

To execute these large-scale mobile robots simulation, we use two supporting software.

One is QOMET [8]. It is a library to calculate the wireless communication specification. In our experiment, each robots process communicate with other robots through wireless network modeling component by using QOMET. By applying the parameter from QOMET to network emulator such as dummynet [9] or Netem [10] or NIST Net [11], we simulate wireless network on cable network.

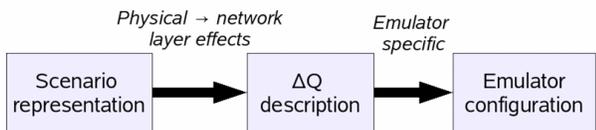


Figure 4. QOMET: WLAN emulation

Second are experiment-support software RUNE [12]. It is designed to support simulation of large ubiquitous networks, having features such as the surrounding experimental environment, networked nodes, real-time experiment, multi-level simulation, etc. Each mobile robots are implemented as the execution process of RUNE.

2.5. The application of mobile robots

In our experiment, the application of mobile robots is motion planning application [7] which avoid any collisions with other robots or any obstacles. Mobile robots try to go to their destination to accomplish their tasks from their initial source. They keeps on avoiding the possible collisions with other robots and static objects. They are equipped with motion planning application to calculate the trajectory to go to their destination without any collisions. The information from GPS, wireless network devices and visual sensor is used to execute this planning application.

3. Automation of experiment

3.1. The issues of large-scale experiment

When we execute large-scale experiments, we encountered lots of issues. Followings are the issues of large-scale

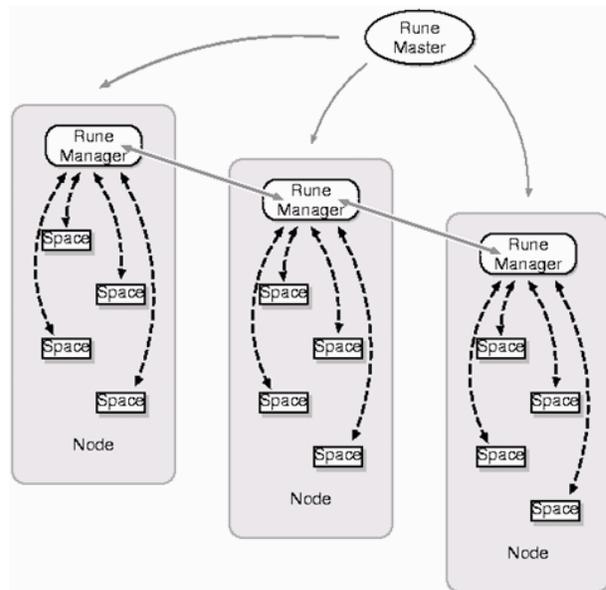


Figure 5. Rune: structure of experiments using Rune

experiment.

1. Network configuration

The experimental nodes should be enabled network interfaces and given IP address. Initially we need to construct experimental environment by configuring all of nodes.

2. Distribution of modified application

Usually we modify the application from the experimental result and distribute it to all of nodes.

3. Centralized management of nodes

We have to control nodes when we start or stop the experiment, gather logs of experiment and change any settings of nodes.

4. Reconstruction of experimental environment

We may execute same settings of experiment again on other node set.

3.2. Experimental infrastructure

To simplify to create the infrastructure of experiment, we prepare PXE boot environment on StarBED. The boot process of the node is describes as follows:

1. DHCP request from pxe boot code of client
2. DHCP reply
3. TFTP request from pxe boot code of client

4. TFTP server sends pxeboot
5. DHCP request from pxeboot of client
6. DHCP reply
7. pxeboot mount NFS
8. NFS server sends kernel to pxeboot
9. Kernel of client boots on mounted NFS directory

All of experimental nodes boot with same kernel on mounted directory of NFS server. If we want to modify the application, we should update only the application of NFS server.

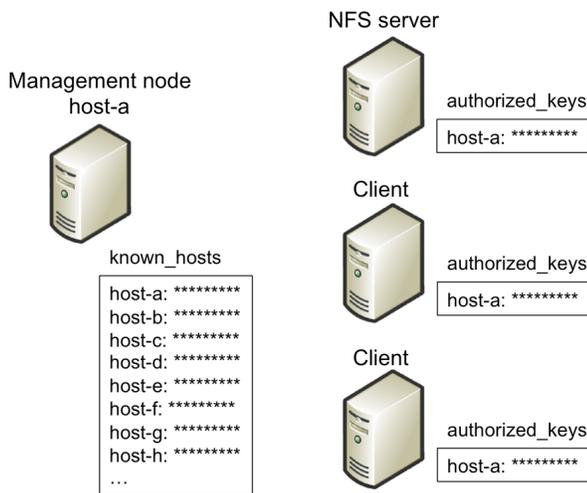


Figure 6. SSH key configuration

The shared directory on NFS server has SSH `authorized_keys` of the management node. Figure 6 shows it. The management node has all of client keys in `known_hosts`, this enables the management node to do remote control of all of the nodes.

This infrastructure can be made by only changing DHCP configuration file, `known_hosts` file and `authorized_keys` file.

3.3. Automatic execution

To solve above issues, we create automation programs of the experiment. The programs consist of shell script set. They create configuration files from one configuration file and control experimental node set. All of control of the experiment can be done by these scripts. Followings are the configuration files and scripts.

- Configuration files

- `mm_starbed.conf`
The node structure on StarBED
- `mm_env.conf`
The identifier, IP address, subnet mask, network interface settings.

- Experimental environment control

- `mm_env-gen`
Generate `"mm_env.conf"` from `"mm_starbed.conf"`.
- `mm_setenv`
Enable network interfaces and set IP address.
- `mm_resetenv`
Reset above network configurations.
- `mm_dhcpd-gen`
Generate `"dhcpd.conf"` file.
- `mm_knownhosts-gen`
Generate `"known_hosts"` file.
- `mm_exec`
Execute the experiment.
- `mm_killall`
Stop the experiment.

We create only one configuration file `"mm_starbed.conf"`. After that, scripts generate all of configuration files, the management node can control all of the experimental environment by using script set.

Listing 1 shows the configuration file `"mm_starbed.conf"`. This is the configuration of three hundreds mobile robots experiment. It means that the management node works on 172.16.4.10 node, the network interface `"em0"` and `"em1"` are enabled on clients from 172.16.4.11 to 172.16.4.110. Every clients execute three processes on their nodes.

Listing 1. mm_starbed.conf

```
M 172.16.4.10 em0 em1
C multinode 3
N range 172.16.4.11 172.16.4.110
```

If users want to use different node set, they change only this configuration file.

4. Experiment

4.1. Experimental settings

As mentioned in previous sections, our mobile robots are equipped with motors to move around. Table 1 describes the parameters of our mobile robots.

definition	parameter
shape of robot	circle
radius of robot	1.0(m)
velocity of robot	0.5(m/sec)
visual sensor range	10.0(m)
WLAN type	802.11b
degree which robot can move	omnidirectional
time step of robot	250(msec)

Table 1. Parameters of mobile robots

The GPS module is assumed as an ideal GPS which can tell the current location. The visual sensor can inform the surrounding objects to the mobile robot. The motion planning application of mobile robot exchanges their trajectory with other application of robot, keeps on changing or planning their trajectory. Every robot is given the initial starting location and the destination, tries to go to their destination without any collisions with other robots or obstacles.

We confirmed the validity and the scalability of the application from the experiments.

4.2. Experimental scenarios

We prepared four scenarios to evaluate the motion planning application on our experimental environment. The difference of the scenarios is the number of mobile robots. We tested one hundred, two hundreds, three hundreds and four hundreds of mobile robots. In each scenario, we use one hundred real nodes as mobile robot and one node as the management node. The scenarios without one hundred of mobile robots execute multiple mobile robot process on one node. For example three hundreds scenario, three mobile robots processes are executing on one node.

4.3. Experimental results

The experimental results can be checked from the logs or the visualizer. The logs are generated by the mobile robot processes. The management node can gather them. Figure 7 describes the visualizer of the experimental results. It can visualize not only from the logs but the real time execution. Colored circles are the mobile robots which are visualized with two dimensions viewpoint. The arrows describe the current trajectories of mobile robots. The one grid is ten meters square.

By increasing the number of mobile robots, we can detect the problems of motion planning application, systematic bugs, bugs of automation programs and the errors of real hardware. Finally we could execute four hundreds number of mobile robots process on our experimental en-

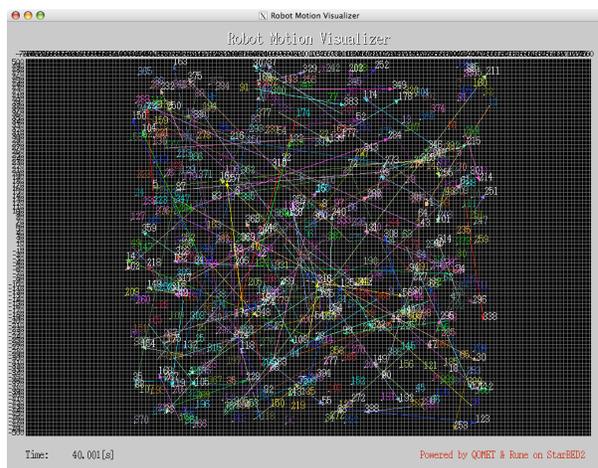


Figure 7. Four hundreds of mobile robots experiment

vironment. In this experiment, four robots process are executed on one node.

5. Conclusion & Future work

In this paper, we propose the simulation method of mobile robots called emulation which mixes the simulation part and real hardware. It has a virtual world to synchronize between simulated modules and real. And we proposed the automation of the large-scale experiment. It enables users to construct the experimental environment easily. We evaluated the motion planning application as the application of mobile robots. Finally we could execute four hundreds mobile robots in real-time.

In these experiments, we have encountered and solved many unpredictable problems. For instance, the deadline miss of the motion planning application, NFS read errors and so on. Sometime the motion planning application failed to plan in large-scale experiment. We could modify the application from the error logs of the deadline miss. When we tried to increase the number of mobile robots over five hundreds, NFS server made "read errors" many times in starting time of the experiment. It is true because much processes access to NFS server at once. In future we will prepare multiple NFS servers for them.

Moreover we should make proper method to detect the errors of the experimental environment. It means that the experiment consists of many application and modules, so it is hard to know analyzing where the errors occur. The experimental environment should have the integrated error detection structure.

References

- [1] StarBED org <http://www.starbed.org/>
- [2] Webots Fast Prototyping and Simulation of Mobile Robots <http://www.cyberbotics.com/>
- [3] EyeSim EyeBot Simulator <http://robotics.ee.uwa.edu.au/eyebot/doc/sim/sim.html>
- [4] RoboCup The Robot World Cup Soccer Games and Conferences <http://www.robocup.org/>
- [5] D.Hsu, R.Kindel, J.C.Latombe, S.M.Rock Randomized Kinodynamic Motion Planning with Moving Obstacles
- [6] Stanford Aerospace Robotics Laboratory <http://arl.stanford.edu/>
- [7] T.Okada, R.Beuran, J.Nakata, Y.Tan, Y.Shinoda Collaborative Motion Planning of Autonomous Robots *3rd International Conference on Collaborative Computing (CollaborateCom2007)*, White Plains, New York, U.S.A, November 12-15, 2007.
- [8] 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
- [9] Luigi Rizzo dummynet http://info.iet.unipi.it/luigi/ip_dummynet/
- [10] Netem <http://www.linuxfoundation.org/en/Net:Netem>
- [11] NIST Net <http://snad.ncsl.nist.gov/itg/nistnet/>
- [12] 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 TridentCom2007, Orlando, Florida, U.S.A., May 21-23, 2007.
- [13] Takashi OKADA, Junya NAKATA, Razvan BEURAN, Yasuo TAN and Yoichi SHINODA Motion Planning of Autonomous Robots UCS2007, pp.57-58, Akihabara, Tokyo, Japan, 2007