Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

HOME NETWORK INFRASTRUCTURE BASED ON CORBA EVENT CHANNEL

Eko Henfri Binugroho, Young Bong Seo, and Jae Weon Choi

School of Mechanical Engineering and Research Institute of Mechanical Technology
Pusan National University, Busan 609-735, Korea
dev_art@yahoo.com

Abstract: This paper proposes a home network infrastructure based on CORBA as a middleware with client-server configuration. Communication inside the network is designed as an event-oriented communication, by using event channel that defined in OMG event service. By modifying CORBA event channel, several functions are developed for enabling the connection, communication, remote control, remote monitoring, and device management of home appliances. CORBA event channel enables decoupling interface between objects, so asynchronous communication between multiple clients can be developed easily. The proposed design has been tested using CORBA based computer test-bed that equipped with PLC Ethernet and WI-Fi communication. Several functions and services that provided in the design have been tested in order to examine test-bed's performance. Simulation of connection from small number of clients until large number of clients has been made, to measure overall network performance. A linear relation between number of clients and the latency of command delivered by server can be seen from the experiment results within test-bed environment.

1. INTRODUCTION

Home networking development has an objective to improve the quality of human life to be more convenient by providing some services such as remote control and remote monitoring of home appliances, home automation, home security management, home multimedia network and so on (Keshi, 2005). Home networking technologies and capabilities have received an increased attention from consumers, software developers, hardware manufacturers, and service providers (G.T. Edens, 2001). Beside that, the robustness of Internet Protocol (TCP/IP) has contributed to its success in the internet environment, and the role of this kind of communication is already well established. It seems that TCP/IP also will become the de facto standard for connecting diverse media throughout the home network. Furthermore the chip's development also became more and more aggressive and it gave a support to TCP/IP protocol. This fact makes and advantages in developing home network appliances.

Many wide-ranging applications in the future home network environment have been proposed, such as; Jini (Sun Microsystems, 2005), LonWork (http://www.echelon.com), LnCP (K.S. Lee, 2001; S.C. Kim, 2002), UPnP (UPnP Forum, 2006), ECHONET (ECHONET Consortium, 2006), HAVi (HAVi Inc., 2000), X10, etc. However there are so many heterogeneous technical aspects in hardware and middleware architecture that used on the home network infrastructure. Home appliances will increase in complexity, and obviously they will need higher connection speed to transfer their information too. Finally the preparation of home network infrastructure that feasible for further application and technology demands is needed yet providing flexibility and also interoperability.

In this paper we propose a home network infrastructure based on CORBA Event Service as a middleware to enable the connection and integration of multiple home devices by giving communication, control, monitoring, and management services. Home network installation includes pulling wire, installing software, and configuring the system (Bill Rose, 2001). Normal house physically doesn't have the existing network infrastructure installed yet. So if we want to install the network, cabling process will became an annoying problem. CORBA based Core Middleware Architecture using IEEE1394 was reported (J.Y. Oh, 2003), but is still needs an additional communication line. To overcome this problem, in this paper we propose a home network infrastructure using Power Line Communication (PLC) Ethernet device as an alternative solution for fast home network connection. By using PLC Ethernet we can deploy the device directly to the existing power line cable infrastructure, especially for connecting devices that separated by wall. Another choice is using wireless connection such as Wi-Fi or combines several network media together in the network.

The organization of this paper is structured as follows: Section 2 describes the overview of CORBA event channel. Section 3 shows the design of proposed home network. Section 4 shows the home network performance evaluation. Section 5 draws some conclusions and directions for future research.

2. CORBA EVENT CHANNEL OVERVIEW

As stated by www.omg.org, CORBA is a vendor-independent architecture and infrastructure. Any vendor on almost any hardware platform, operating system, programming language, and network, can interoperate using

978-1-1234-7890-2/08/$20.00 © 2008 IFAC

10.3182/20080706-5-KR-1001.4028

CORBA-based program. CORBA also still can handle large number of clients, at high hit rates, with high reliability. It is not only suitable for large applications but also specialized versions of CORBA can run a real-time system even in a small embedded system. Because CORBA itself is a distributed object computing, home appliance can be described as an object inside the home network programming structure. Within this object oriented environment, programmer can develop the application easier.

Event channel is the central abstraction of OMG Common Object Services (COS) Event Service, which plays the role of a mediator between consumers and suppliers. Event channel manages object references to suppliers and consumers (D.C. Schmidt, 1997). Suppliers generate events to the channel, and consumer will receive that events. Event Channel it self is a mediator that encapsulates the queuing and propagation of events from supplier to consumer. Both supplier and consumer may choose between a push or pull model in delivering or acquiring events. The push model calls the consumer asynchronously as soon as events are ready for it. If the pull model is used, the channel stores events for consumer until consumer decides to call the channel to collect them.

Despite of many features given by CORBA event service, its overhead process still considered as major disadvantages. Event channel uses excessive network utilization while delivering unnecessary events especially under large number of client condition. Modifications are needed in order to reduce this process and finally can increase communication throughput**.** A context-based dynamic channel management for efficient event service already studied (Y.K. You, 2007).

3. HOME NETWORK DESIGN

In this proposed home network, ACE/TAO event service is chosen for developing home network test-bed. Connections and services are developed based on the event channel with client-server configuration.

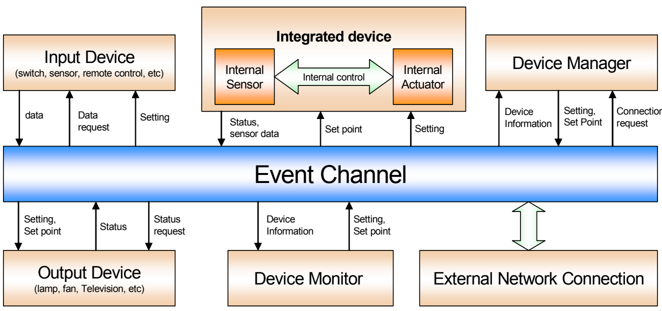

Fig. 1. Home network infrastructure based on CORBA Event Channel

Figure 1 shows the connection of home network infrastructure inside CORBA event channel point of view. In the event channel, we configure user input setting and input device as a suppliers that generate event to the channel. Output device is configured as a consumer that receives events from the channel, while integrated device and device monitor configured both as consumer and supplier in the event channel. And device manager is configured as event channel administrator to manage the devices connection, both consumer and supplier.

Several modifications are made in ACE/TAO event channel service to make it suitable with home network application. In the native event channel, channel just delivers data to all consumers, because there is no routing decision for delivering data, and it will increase the network utilization. One of modification is making a routing decision in the channel, yet maintaining the decoupled fashion. This route decides the event transfer by grouping several suppliers and consumers that have same criteria, such as device type or device location. Structured event is chosen to replace type any event that usually used in event channel communication.

3.1 Server Configuration

Server is built using a computer based on CORBA middleware that equipped with the communication interface for outside and inside network (Fig. 2).

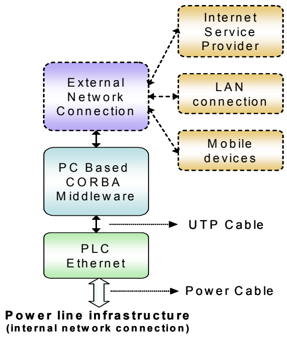

Fig. 2. Server hardware configuration

Server is designed to provide several functions to:

- manage devices and distribute data among clients (as a notifier in the event channel).
- control and monitor every devices attached to the client part.
- bridge the control and monitoring process with the external connection such as LAN, internet or mobile device to make the monitoring and controlling process of home network devices more extensive and flexible.

3.2 Client Configuration

Similar with server, our client test-bed also using a computer based on CORBA middleware that equipped with PLC Ethernet and an interface board (Fig. 3).

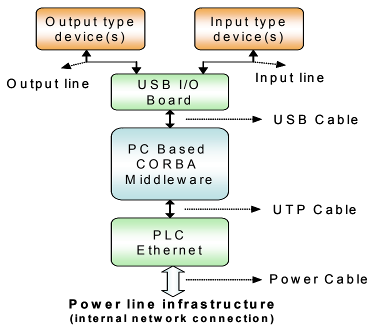

Fig. 3. Client hardware configuration

For client that configured as device monitor, attachment of the interface board is not needed, because it will be connected to the device remotely using event channel. Clients can be configured into several configuration:

- output device : only connectted to controlled device (configured as consumer in the event channel).
- input device : only connectted to sensor or another input device (configured as a supplier in the event channel).
- combined input-output device : connected to the controlled and sensor device (configured as both consumer and supplier).
- designed as a monitoring device (configured as both consumer and supplier for remote control and monitoring process).

3.3 Server Side Program

In order to examine test-bed's performance, several services are made in the server part. Block diagram of Server side program is shown in the figure 4.

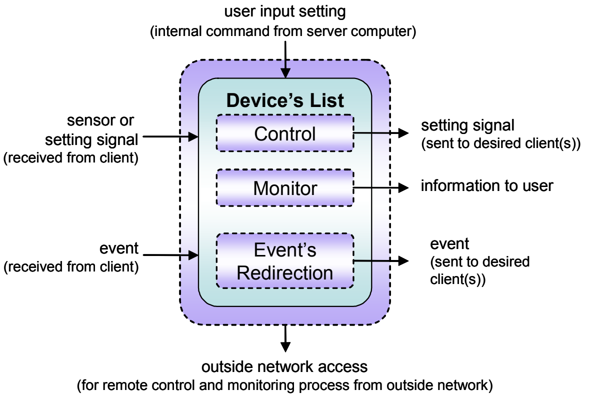

Fig. 4. Block diagram of server side program.

- Control block is a user interface to give a setting point into client using user input device such as keyboard, mouse or touch screen (attached at server computer).
- Monitor block is used to display information about all client connected to the server.
- To maintain client's connection all clients connection's information saved in a device list.
- Server also redirects command and data from one client to another in the server's device list
- If there is a device monitor available in the network, server will redirect all events that invoked by server or clients (sensor data, input setting, and command) to device monitor.

3.4 Client Side Program

Client side program is divided into two types, the fist type is a client that connected to the input or controlled device, and the second one is designed for monitoring device. Figure 5 shows the block diagram of client side program that connected to the device. In order to reduce the communication's traffic, internal control process is designed to take place in each client. Data that is transferred on the communication line are only data for setting, control and monitoring process that can be sent either by server or other clients.

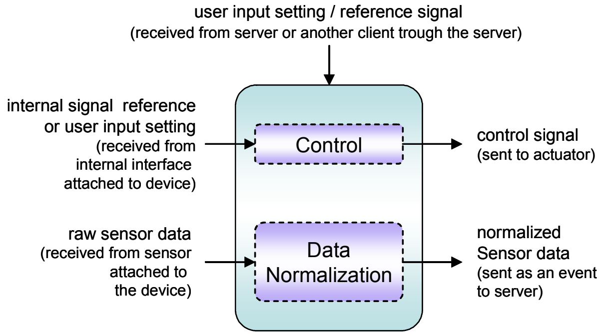

Fig. 5. Client side program connected to device.

Any changes in sensor data of each client can be sent to server as an event for monitoring process using push method. In another way, server can get sampling of data in a periodic time for each client using pull method. Data normalization block converts raw sensor data into SI and before it is used as a controller feedback or information for monitoring process. If client is proposed as an input only device (independent sensor, switch, remote receiver, etc), it is not connected to device to be controlled directly. In this case all data are sent to another desired device through the server first. If client is proposed as an output only device (lamp, on/off device, etc), reference signal can be received either from server directly or from another device (input only device, monitoring device) through the server.

For combined input-output device (such as refrigerator, air conditioner, etc), reference signal sent to the controller is came its own interface attached in the device or from event. In this case all information (any change in the status value) should be sent to server also in order to keep information at server is up-to-date. Another type of client is proposed as a monitoring device which is designed to monitor and control another device through to the home network remotely. These functions are similar with functions provided in the server part. The difference between control and monitoring process implemented in server is the communication path. All data transferred to or from devices are passed to the server first, and then server redirects these data among device monitor and other dedicated clients. Block diagram of client side program that use to be connected with device is shown in the figure 6.

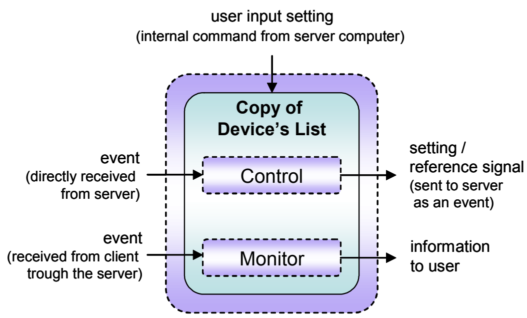

Fig. 6. Client side program designed as a device monitor

Device monitor is designed in order to extend the flexibility of monitoring and controlling process in the network, because it can be deployed anywhere in the home network structure.

3.5 Device Description on Event Channel

Using Interface Description Language (IDL), we describe devices in the home network test-bed as a structured event that will be delivered through event channel (Fig. 7).

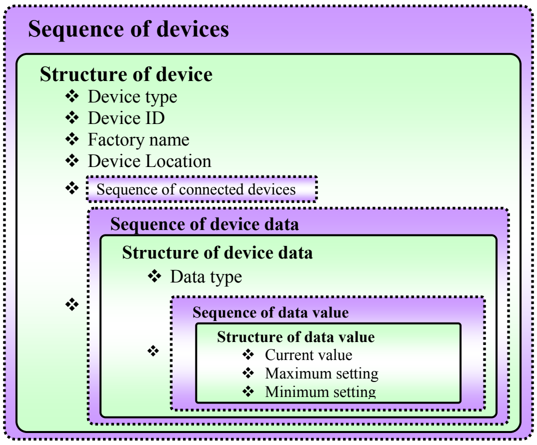

Fig. 7. Structure of device data

- Device information structure is written as a nested type sequence of structures. The length of sequence is variable in order to extend its flexibility in accommodating heterogeneous type of devices.
- Detail information that stored in each device is divided into static, semi static and dynamic properties.
  - ✓The static properties are a read only attribute that cannot be changed.
  - ✓Semi static properties store device's configurations that can be written in the run-time but still to be stored when the device is turned off.
  - ✓Dynamic properties can be modified in the run-time process.
- Device's data is stored in a sequence of device data that can wrap several data types. Each structure of device data contains the information about each data type and each sequence of data details.
- Once device's detail information is sent, in order to reduce the size of data transferred in the communication line, only sequence of data value is needed to be sent into event channel unless another host has requested again that detail information.
- Device ID is a unique property to identify each device. In the practical implementation, it can store a serial number provided by device manufacture. Implementation program uses this ID to distinguish one device from another.
- Device location is attribute that can be set by user to tell installation location of the device or can be use for grouping event delivery.
- List of connected device stores IDs of other devices connected to that device. This list is used by the channel as routing information in delivering events.

4. PERFORMANCE TEST

4.1 Test-bed Configuration

To test the functionality of home network design, in the experiment we make a simple test-bed using 3 computers that configured as one server and two clients. PLC Ethernet is utilized for mediating communication process between client and server. PLC Ethernet are attached between each host and power line plug, and for better communication performance all PLC Ethernets should be plugged at the same cable line. Wi-Fi can be used to replace PLC Ethernet if communication performance is not acceptable and additional cable is not desired.

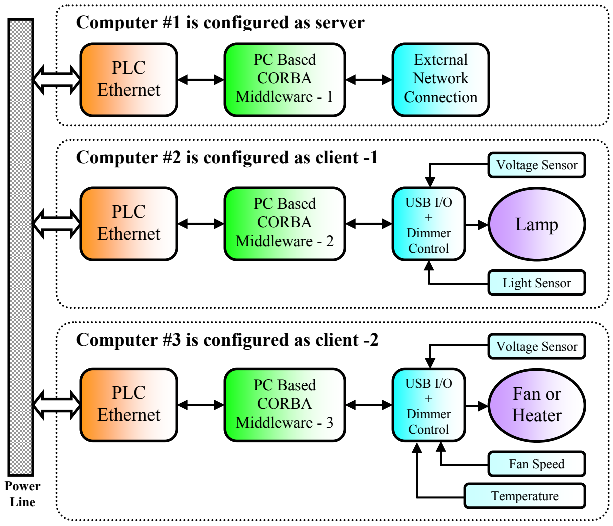

Fig. 8. Test-bed configuration

Server's specification

CPU	: Pentium 4 3 GHz
RAM	: DDR 2, 1GB (Dual Channel enabled)
Mainboard	: BIOSTAR 915P –A7 COMBO
Ethernet Adaptor 1	: Onboard RTL 8110C (Realtek 8139 Family)
Ethernet Adaptor 2	: Realtek RTL 8139C
PLC Ethernet	: HomePlug1.0 Turbo 85MBps Ethernet Adaptor
WI-FI	: D-LINK DI-624+ wireless router

Client's specification

CPU	: Pentium 4 2.6 GHz
RAM	: DDR 1, 1GB
Mainboard	: ASUS P4B533
Ethernet Adaptor	: Realtek RTL 8139C
PLC Ethernet	: HomePlug1.0 Turbo 85MBps Ethernet Adaptor
USB I/O Board	: K8055 USB Interface Board

Client 1(connected to lamp and light sensor)

Lamp	: Bulb 60 watt
Light sensor	: LDR 9K-200K

Client 2(connected to heater and temperature sensor)

Heater	: 600 watt
Temperature Sensor	: LM 35 (National semiconductor)

Function schemes that have been used in the experiment are divided into two parts. The fist parts only consist of functions that connecting server with one client (Fig. 9), and the second parts consist of functions that used for communicating two or more clients trough server (Fig. 10).

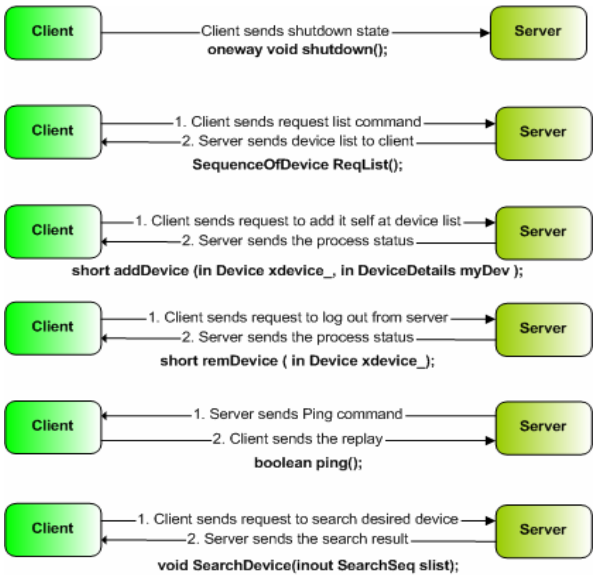

Fig. 9. Functions that communicate server with one client

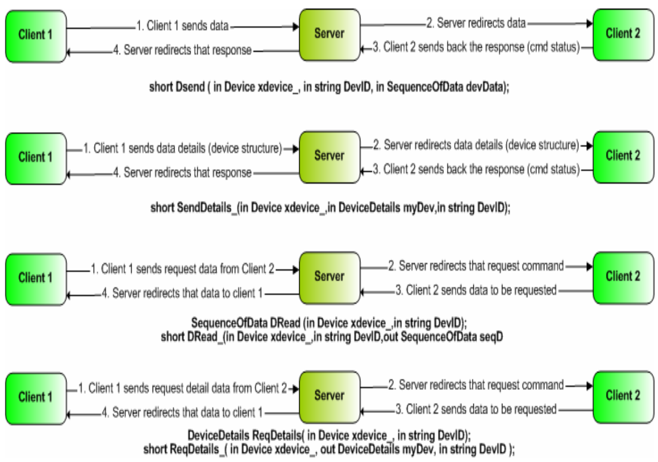

Fig. 10. Functions that used for communicating two or more clients trough server.

4.2 Experiment Result

This experiment is divided into two types of test, the first one is to observe the functionality of the design using test-bed configuration that is depicted in figure 8. The second test measures the network performance by measuring the latency in executing command over the network. To measure the performance server and client part are located in different computer, but all clients are simulated in the same computer. The performance tests it self are divided into three types of test. The first performance test measures the single access latency from server to client. Communication speed response between client and server is measured using ping command with periodic pooling method. Each command only takes one round communication from server to one client and then returns to server again. Server pools the response of ping command from all connected clients. During test, all events that triggered from clients are disabled in order to put all server resources to handle ping command. With the proposed scheme, we obtain the latency measurement result from one client until one hundred clients that is depicted in Figure 11.

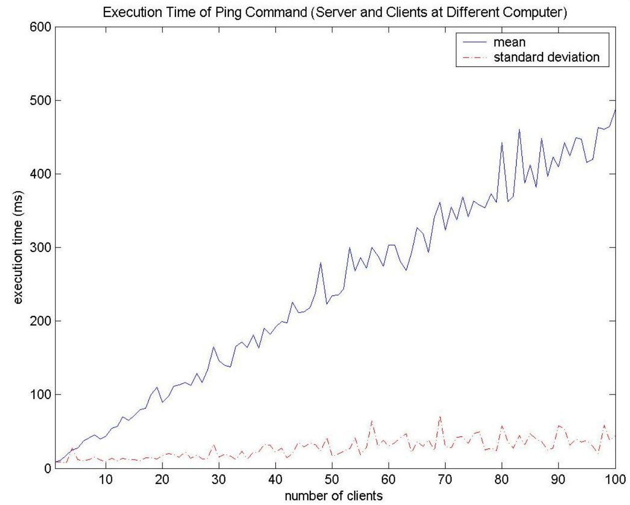

Fig. 11. Latency of ping command

The second performance test uses request detail command to measure latency from one client to other clients through server. Execution time is measured since client sends the request detail command until it receives back the response from server. Server receives request from a client and then it will run a batch process of one round command in order to request data from other clients inside the server's device list. All information is saved in a sequence of device structure before it be sent back to client that has been requested the data. The result of this test is shown in the figure 12.

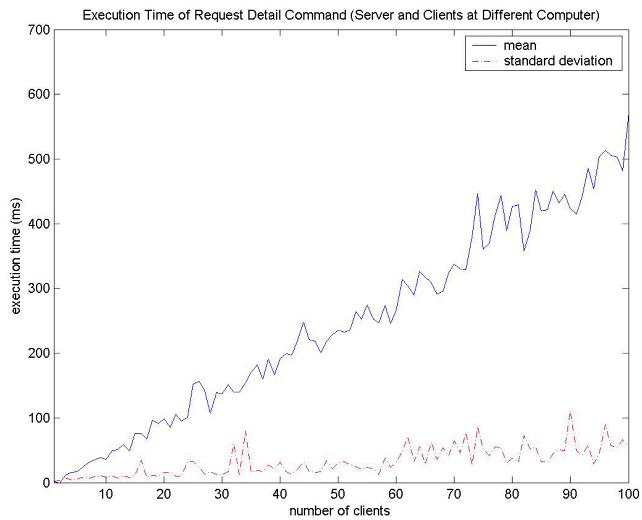

Fig. 12. Latency time of request detail command.

The third performance test measures the latency for simultaneous request from several clients at the same time. In here, clients were grouped into pairs, and each pair consists of two clients. We use windows SYSTEMTIME to synchronize each pair in sending the request to server. Figure 13 shows the latency time for simultaneous access using one pair and two pair of clients. The latency time is measured within one hundred measurements for each pair. In one pair tests, the mean value of latency is 5.91041 ms with standard deviation value is 0.813989 ms. While in two pairs measurements, the mean value of latency is 8.730986 ms with 0.51416 ms standard deviation.

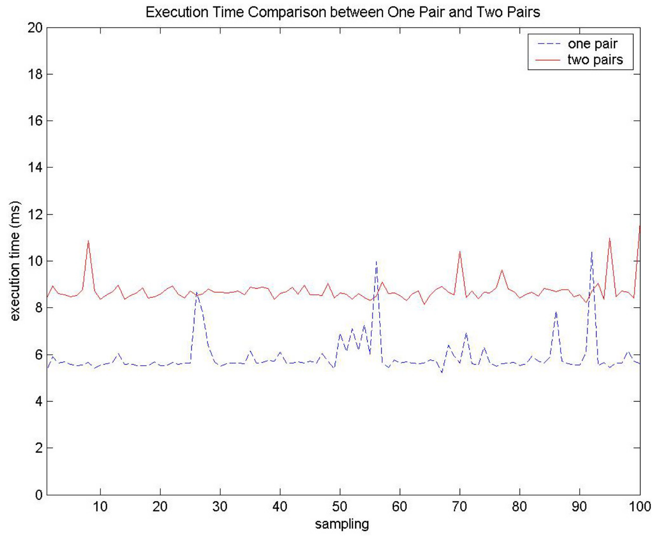

Fig. 13. Latency comparison between 1 pair and 2 pairs of client

To measure the latency time of simultaneous access to the server according to the increasing number of clients, several pairs are loaded in the client computer incrementally from one pair of clients until 50 pairs of clients. Each pair is measured in one hundred times, and the measurement result is shown in the figure 14.

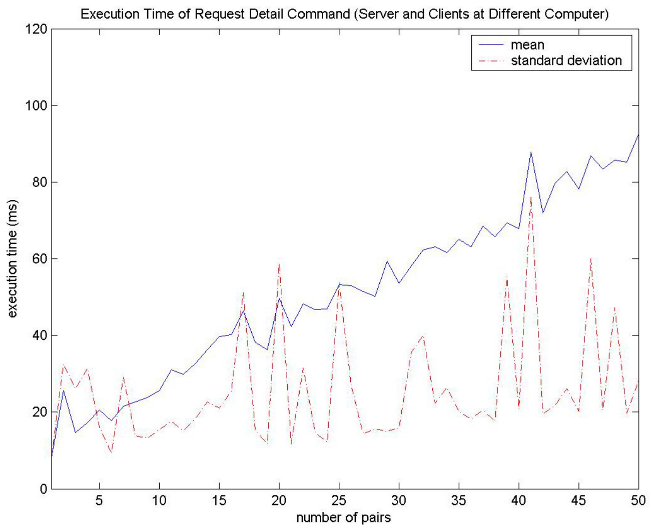

Fig. 14. Latency for simultaneous accesses regarding to number of client

5. CONCLUSIONS

In this paper, we design home network based on CORBA middleware. From the experiment results we withdraw several conclusions:

1. CORBA Middleware will be a feasible solution in home network technology and event channel also becomes a reliable method in delivering information among clients inside the home network.
2. Simultaneous access to the event channel gives less total latency then single access that use pooling method.
3. One round process (server - client - server) has similar execution time with two rounds transfer process (client A - server - client B - server - client A).
4. Reducing the client's system specification still available for the real implementation, because in the experiment a simulation of simultaneous access from 100 clients still works properly inside single computer.
5. All command latency increase almost linearly according to the number of connected device.
6. For the future work, priority controller can be used to maintain the latency an important client when the number of client is increased.

REFERENCES

Keshi, D. Shiraishi, Y. Niwamoto, H. Okada, and M. Yamamoto (2005), Is Home Network Application Acceptable or Not?, *Circuits and Systems,* ISCAS, **vol. 5**, pp. 5337-5340.

G.T. Edens (2001), Home Networking and the CableHome Project at CableLabs, *Communications Magazine,* IEEE, **vol. 39**, pp. 112-121.

Sun Microsystems, Jini Specifications ver. 2.1, 2005.

http://www.echelon.com.

K.S. Lee, H.J. Choi, C.H. Kim, S.M. Baek (2001), A New Control Protocol for Home Appliances - LnCP, *proceedings of ISIE 2001*, May 2001, pp. 291-296.

S.C. Kim, J.A.Park, K.W. Lee, and S.W. Lim (2002), Home networking digital TV based on LnCP, *Transactions on Consumer Electronics,* IEEE*,* **vol. 48***,* pp. 990-996.

UPnP Forum, UPnP (Universal Plug and Play) Device Architecture, ver. 1.0, 2006,

ECHONET Consortium, ECHONET Specification ver. 3.20, 2006.

HAVi Inc., Specification of the Home Audio/Video Interoperability (HAVi) Architecture ver. 1.0, 2000.

Bill Rose (2001), Home Network : A Standards Perspective, *Communication Magazine*, IEEE*,* **vol. 39**, no.12, pp. 78-85.

J.Y. Oh, J.H. Park, G.H. Jung, and S.J. Kang (2003), CORBA based Core Middleware Architecture Supporting Seamless Interoperability between Standard Home Network Middlewares, *Transactions on Consumer Electronics*, IEEE*,* **vol. 49**, no. 3, pp. 581-586.

D.C. Schmidt, and Steve Vinoski (1997), Object Interconnections - The OMG Events Service**, Column 9**.

Object Management Group. http://www.omg.org.

Y.K. You, S.W. Han, H.Y. Youn (2007), Context-based Dynamic Channel Management for Efficient Event Service in Pervasive Computing, *International Conference on Computational Science and its Applications*, ICCSA, pp. 24-32.