IPv6 Home Network Domain Name Auto-configuration for Intelligent Appliances

Tin-Yu Wu, Chia-Chang Hsu, Han-Chieh Chao

Abstract — *When network techniques advance dramatically, Network applications merge broadly into the daily lives of people. More electrical appliances now contain tiny embedded systems that support network interfaces within the home due to the great advances in IC design and semiconductor device manufacturing. Remotely controlling an information appliance (IA) in a home network has become a major request for nowadays consumers. IPv6 plays an important role because of the enormous number of device network interfaces needed in the home network around the world. In our work, a protocol that automatically integrates the user interface and command transmission is proposed. The proposed method also allows IA to acquire its domain name automatically without manual configuration. Through this initial automatic message communication protocol, users can "plug and play" the IA device with its unique ID.* [1]

Index Terms — IPv6, DNS, IA, OSGi

I. INTRODUCTION

There are relevant issues pertaining to the development of IA (Information Appliances). The burgeoning home network market was forced to use NAT (Network address translation) to set a private IP address because a large number of IP addresses cannot be easily obtained. Because the exterior host cannot establish a connection into a host in the NAT, the user generally cannot control IA through peer to peer at home unless a special NAT configuration is established. Another difficulty is getting IA a domain name through NAT because it is a private IP configuration. Confronting more IA support networks in the future, such as refrigerators, electric radiators, video recorders, the home gateway with a NAT function is not enough. The next generation IP protocol - IP version 6[1], provides more characters that can be used for embedded IA systems. The IPv6 protocol provides for enormous IP addresses. This system will give each IA one unique IP address that can be addressed automatically without a DHCP server. It also makes it convenient to register a domain name for an embedded system with a global IP address routed into a public network. It is not feasible at present time for every IA user to register a domain name through Ipv4 due to the limited space availability.

The most attentive home network architecture protocol is Open Services Gateway Initiative (OSGi)[2]. OSGi provides for a complex and large residential gateway mechanism in a home network. It supports various applications called bundles that can be installed in the OSGi service platform. User can start or stop the bundles according to their needs. OSGi uses Java Intelligent Network Infrastructure (Jini) and universal plug and play (UpnP) technology to assist the service platform in discover and service a device [3]. The Jini network technology enables devices to form impromptu communities that can be assembled without any planning, installation, or human intervention. The UPnP technology provides protocols that specify how a device joins a network and is controlled using XML messages through an HTTP server for a peer-to-peer network. These technologies can assist the user in decreasing the number of configuration actions. Configuration actions are significant in a home network because the OSGi organization is devoted to adding new techniques into OSGi.

Three other related windows server 2003 applications are applied for DNS and home network applications. They are explained thoroughly as follows.

(1) Dynamic DNS

Dynamic DNS have been used for a long time [4]. In IPv4, many organizations provide dynamic DNS services. Translate IPv4 dynamic DNS service into IPv6 is quite easy, but with a problem that the client application must be applied. The client application must configure by user, therefore IA owner cannot get domain name automatically through plus and play because IA has not any configuration inside beforehand.

(2) IPv6 Stateless DNS discovery

IPv6 Stateless DNS discovery is a new draft proposed in IETF [5]. It mainly supports a service that assist client host to configure DNS when Router Advertisements function cannot achieve it. The procedure can only assist host to con-

[1] This work is a partial result of project no NSC 91-2219-E-259-008- and NSC 91-2626-E-233-001-conducted by National Dong Hwa University under the sponsorship of the National Science Council, Taiwan, ROC.

Tin-Yu Wu, Chia-Chang Hsu, and Han-Chieh Chao are with the Department of Electrical Engineering, National Dong Hwa University, Hualien, Taiwan, Email: {tyw, m9123030, hcc}@mail.ndhu.edu.tw

Contributed Paper
Manuscript received February 24, 2004

figure DNS but it cannot get a domain name for host automatically.

(3) Peer-to-Peer Networking Name Resolution

The Peer Name Resolution Protocol (PNRP) [6] supports a service by Name Space Provider. It provides an API that permits the peer-to-peer resolution of names to endpoints. This protocol also is acting as a name resolution related protocol.

Another mechanism supporting same function as one of our proposing mechanism is Domain Name Auto-Registration for Plugged-in IPv6 Nodes [7]. This draft also supports domain name auto-registration function, but it cannot let device obtain a regular domain name that users expect for. It will reduce the popularity for this mechanism. The comparison is listed in table 1.

TABLE I
Related DNS services

Service / Function	Dynamic DNS	IPv6 Stateless DNS discovery	PNRP	Domain Name Auto-Registration for Plugged-in IPv6 Nodes	Domain name auto-configuration (We propose)
Register Domain name	Yes, but user must configure client host.	No support.	No support.	Yes, but user cannot anticipate type of domain name.	Yes, support registering domain name that can be anticipated.
Resolve domain name	No support.	Yes, must configure DNS.	Yes, but host must install API.	No support.	No support.

In this paper, we propose three mechanisms that add additional functions to home network architecture. The first function assists IA in acquiring a regular domain name without manual configuration. Because a domain name has wide utility in many applications such as SIP etc, this mechanism will carry more services into a home network. The second mechanism is similar to the previous one. It mainly provides session initiation protocol (SIP)-Uniform Resource Identifiers (URI) auto-configuration and SIP-URI register. This procedure will also become much convenient in VoIP devices. The third function is a mechanism that initiates communication messages between devices manages the residential gateway and configures the user management system interface simultaneously. It provides another, thin home network architecture.

II. HOME NETWORK ARCHITECTURE

Although the present IPv4 protocol has been in use for many years, it can no longer satisfy user requirements in many applications. For this reason, the newer IPv6 protocol was instituted. Besides the larger IP address, IPv6 can use Stateless Auto-configuration or DHCP6 to obtain IP when new devices begin network interface [8]. In the stateless auto-configuration, the device uses only a prefix that route advertises and the network MAC interface to compose an IP through EUI-64. Another Multihome characteristic allows a network interface to support more than two IP. Devices. One IP is used to configure, the other IP is used to communicate. Other characteristics provide more support for QoS, Security and enhance the home network architecture.

A wireless network is a far better choice in a home setting. Wireless networks provide both PC and IA access to network resources without being hampered by space limitations. Furthermore, AP can also be used as a home gateway in conjunction with a PC in a wireless environment without paying any extra cost. Another nice choice for a home network is the powerline device. Powerline network communications occur through a power cord, which is fitting for IA in a home network.

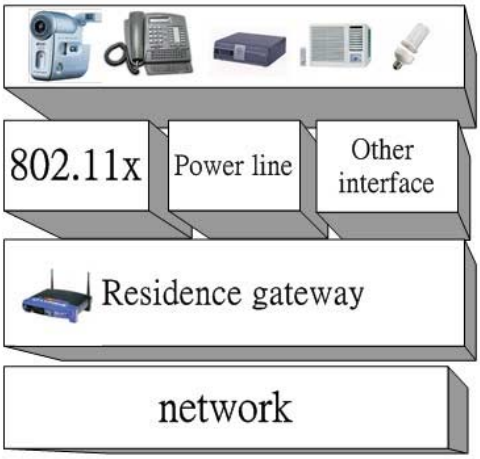

Fig. 1. Wireless home network architecture

Figure 1 illustrates home network architecture with AP and powerline devices. An AP or PC is used as the home gateway in this architecture. Any device that has a wireless interface or powerline device can connect to the Internet through the home gateway, i.e., wireless IA devices, printer, PC or Internet Telephone [9].

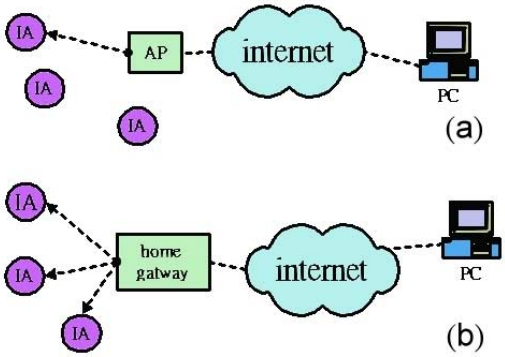

Fig. 2. Remote control methods

There are few control restrictions for IA devices in a home network because of the network interface. Controlling messages can be made of serial numbers, characters or any data. This produces nimble control and data exchange in a home network. Controlling IA with global IP is accomplished in two ways as illustrated in Figure 2. In Figure 2(a), the user can control IP directly through a program like BBS or a thin web server. In Figure 2(b), the user connects to the home gateway first and uses a management interface to controlling all IA devices. The later method provides more practicability with advanced designs to deal with commands and messages [10].

III. AUTOMATIC DOMAIN NAME CONFIGURATION AND SMALL MANAGEMENT INTERFACE

3-1. Domain name auto-configuration

We propose a procedure that enables a PC or embedded system to acquire a domain name automatically at extremely small cost. The manufacturer only needs to embed a program that runs during the system boot. The original system design needs to be modified to add different routines to the embedded IA system. IA uses a local IP to exchange messages with the home gateway and another global IP to communicate with a foreign network. Figure 3 illustrates how the embedded system acquires a domain name. This procedure can also be used in a PC or other network device. The following steps show how a Linux embedded system acquires a regular domain name.

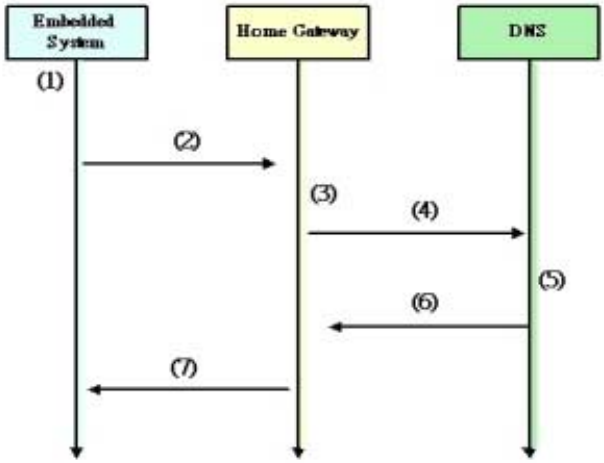

Fig. 3. The domain name auto-configuration procedure

(1) In the embedded system, a program installed in the file system is executed by the initiation procedure after the system loads the kernel. The IP address generated automatically through this program is then collected.

(2) The next step is to send the IP and appliances label defined in embedded system to the home gateway.

(3) After the home gateway receives the IP and appliance labels, a program in the home gateway combines the user's ID and appliances label to generate a combined string to register a domain. The program also records the IP and IA information in the home gateway mini database. This information can be used later to manage the home network.

(4) The home gateway program then sends a combined string to register at the DNS. The combined string is a domain name type string. For example, if the appliance label is "refrigerator", and the user ID is "user", the combined string the home gateway sends to the DNS is "refrigerator-user". If the DNS management domain is ".ia.tw", the registered domain name is "refridgerator-user.ia.tw". A regular domain name "type-user_ID.ia.tw" is produced.

(5) The program in the DNS that receives the domain name registration message from the home gateway writes the domain name and corresponding IP address into the DNS configuration file and reloads the DNS configuration. The domain name can be used after this procedure is completed.

(6) The DNS then returns a value that indicates successful registration to the home gateway. The home gateway then disconnects the connection to the DNS.

(7) The home gateway then returns a successful value to the embedded system. The embedded system then disconnects its connection to home gateway.

There are several advantages in the above procedures. For the embedded system, only new programs are added to the original system. Messages to the home gateway are received when the system boots up. If the home gateway does not accept these messages, the embedded system or home gateway is not affected. The program in the embedded system can be added to every IA that has a remote control function. The user can decide whether to use this function. AP manufacturers can insert programs like ours into the AP to make it a home gateway and support domain name registration. This is similar to the preceding state and will not affect the AP's original function. Another advantage is that the AP can cooperate with a PC to compose a home gateway without the presence of the program.

More detailed configurations will be discussed next. The appliance name affects its label. Every information appliance has a relevant name. Different embedded electric appliance systems must have different appliance labels identical to the appliance word set in the system. A refrigerator is "refrigerator", video recorder is "video-recorder", etc.. The string composed using two or more words is linked by "-"because of the domain name configuration requirement. If there are two electrical appliances that have same appliance label, the home gateway solves this problem by generating a different domain name for each appliance of the same type. The appliance label can be extended to refridgerator1, refridgerator2, etc. PC users can execute a program to send registered messages to the DNS or use the PC as an information appliance

following appliance registration procedure using the appliance label "PC".

3-2. Experiment for automatically obtaining a domain name

To put into practice the domain name registration procedure, we set up a simple home network to verify the DNS auto-configuration functions.

Figure 4 shows a flow chart of the experimental system. The IA is an embedded system using the Linux 2.4.18 version kernel. NB is a notebook with the Linux operating system.

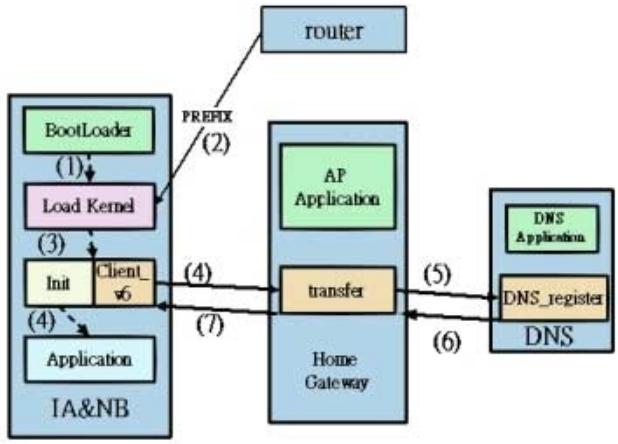

Fig. 4. Domain name auto-configuration flowchart

Both boot procedures are the same and all use the same wireless network interface supporting 802.11b to connect the home gateway. The most important feature of this setup is the Ipv6 support in the kernel that supports stateless configuration. In the system initiation procedure, a compiled statistical program "client_v6" is added to send the IPv6 address and appliance label to the home gateway.

Because it is difficult to add a program in AP, we used a personal computer and AP to compose the home gateway. The PC used the Linux operating system. Daemon "transfer" is run to assist the embedded system to register a domain name and allow the user to set his user ID.

We set up a DNS server that supports IPv6 in the PC. A daemon "dns_register" is executed to receive registered messages from the embedded system and reload the DNS configuration when the program deals with a register message. After reloading the DNS configuration, a new domain name will be used. To broadcast the prefix and allow the home network to communicate with outer networks, a router is set above the home gateway. The following are other configurations that must be applied. To allow the embedded system to find the home gateway, individual IPs are distributed to each embedded system and home gateway. The two IPs are "3ffe:3600:1::1111" and "3ffe:3600:1::-2222". These IPs are used only for communicating between embedded systems and the home gateway. A router will not forward packets with IPs in the "3ffe:3600:1::/64" domain. This way our program operates in the home gateway and embedded system without having to change any source codes. This is a little like the private network IP address of IPv4. The user ID and IP of the DNS can be configured manually to determine desired domain names. In our experiment, the three programs were written using C language and operated as shown in Figure 3. A regular domain name is registered after the embedded system boots within approximately 5 seconds. DNS's configuration reload time is the only time expended.

3-3. Extend domain name – ENUM and SIP

Handheld devices have number keys that are different from a PC keyboard. To popularize VoIP and allow handheld device Internet connection with domain names, IETF instituted RFC2916 to allow a string of numbers to be used as a domain name [11].

Figure 5 shows a zone file. In an ENUM DNS zone file configuration can be set if a device has a domain name. A user can register a number to IA easily because all IA domain names are regularly generated by our method. A device domain name can be foreseen without installing the device because the device label "VoIP" is known. Users can therefore connect to IA and control the device through handheld devices that have a browser.

```
$ORIGIN 4.3.2.1.6.7.9.8.6.4.e164.arpa.
 IN NAPTR 10 10 "u" "sip+E2U"    "!^.*$!sip:sven@sips.se"
 IN NAPTR 10 10 "u" "mailto+E2U" "!^.*$!mailto:sven@ispa.se"
 IN NAPTR 10 10 "u" "http+E2U"   "!^.*$!http://svensson.ispa.se"
 IN NAPTR 10 10 "u" "tel+E2U"    "!^.*$!tel:+46-8-9761234"
```

Fig. 5. ENUM zone file

The VoIP device in home network can also use this method to economize configuring time. Through SIP and ENUM integration, VoIP device can initiate a session without a proxy. In this way, using domain name or IP is the only method to initiate session. We also propose another method for device initiating session through proxy. The procedure that we propose is called SIP URI auto-configuration and will mainly provides SIP URI registration [12]. Figure 6 illustrates the SIP URI auto-configuration procedure. The following are the steps needed for the operation:

Step 1: In the embedded system, it operates as domain name auto-configuration only to send device label and IP address to home gateway.

Step 2: when the home gateway receives a message from the embedded system, it determines the device label. If the device label is "VoIP" or some other device that uses SIP, the home gateway executes another register procedure. The home gateway delivers a user ID and IP address to the Registrar defined previously by the user.

Step 3: The registrar will store the data into the Location Service and the VoIP device will obtain a SIP URI.

Steps 4-6: After the response is received by VoIP device, the SIP URI auto-configuration is executed thoroughly.

No authentication process is used in our procedure. All authentications follow the SIP standard. According to our procedure, the user only defines the UserID and domain in the home gateway, then obtains a SIP URI formation as "UserID@domain".

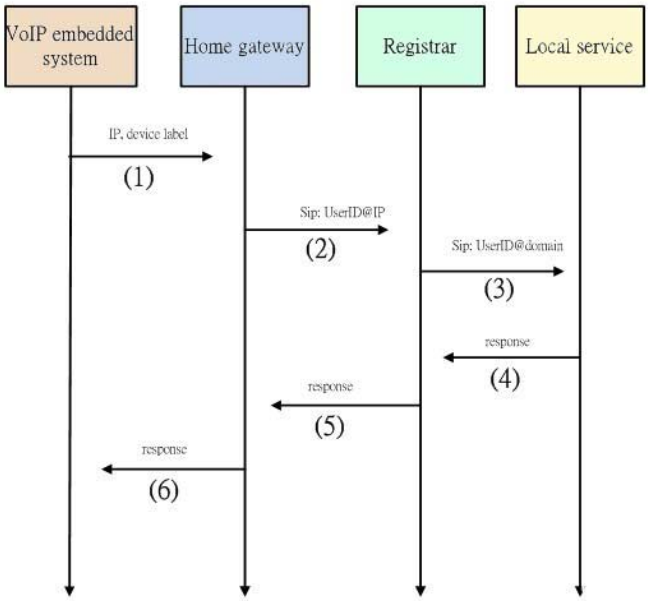

Fig. 6. SIP URI auto-configuration procedure

3-4. A high comparability management home network interface

Previously, we focused on how an embedded system acquires a domain name automatically. The home gateway combines the appliance label and registration number. Because most home gateways carry electric appliance control functions, another way to enhance control interfaces with different manufacturer IAs is discussed. To achieve high IA comparability management interface, we can carry a "function string" to allow the home gateway management interface know the IA's function when a domain name is acquired. Because the home gateway knows the functions of all connected appliances, the user can control the functions of every IA through the management interface. This management interface is achieved using PHP.

Figure 7 illustrates how to operate the high comparability management system.

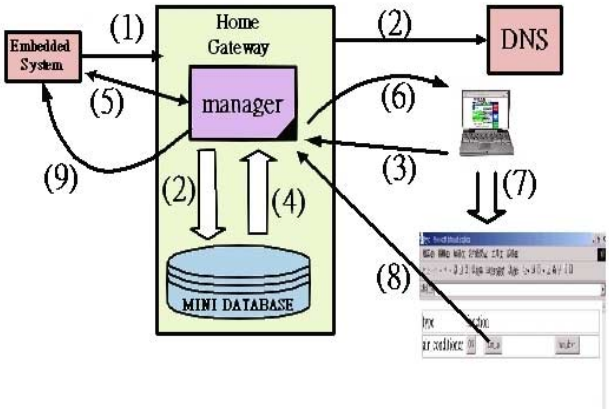

Fig. 7. the flow chart of the management system

The following are the steps needed for the operation:

Step 1: In the embedded system, after the system loads the kernel, a program sends the appliance label, IP address and function strings to the home gateway.

Step2: After the home gateway receives all messages from the embedded system, a program in the home gateway stores all messages into the mini database and registers a regular domain name.

Step 3-7: When a user connects to the home gateway management interface, the management interface will list all IA's types and each function and device state string. The user can select the display function string to start the related function.

Step 8-9: After the button is clicked in the browser, the home gateway returns a command to the embedded system to start the IA function.

The function string is a string like "Power-_ON", " Probe_temperature", "Turn_up", "Strong_wind" etc. Function strings are defined by IA manufacturers and have concrete definitions. This means that IA system control commands are not be restricted by the home gateway if a user uses the home gateway to control electric appliances. For example, an air conditioner has "probe_temperature" functions, "Power_ON", "Power_OFF", "Turn_u-p", "Turn_down". In the user's view, the function string is the button identity. Clicking the button will start the function. The function string is also the command that the management interface uses in the embedded system to start a given function.

Another important message is the device state string. It is returned from the embedded system to tell the management interface what states appliances are currently in. An example is"//Power_ON=on/Power_OFF=off/Probe_temperature=24/Turn_up=off/Strong_wind=on//". Management can list device state strings directly on the page to tell the user the state of a given device. Script language can also be use before listing. This architecture enables the home gateway to easily control various IA devices. However, this

architecture requires more embedded system design. Additional design is required for the embedded system to receive function strings and transfer them to electronic appliance control through RS-232 or USB.

3-5. Experiment for home gateway management interface.

To achieve our objective, we constructed a small home network with an embedded system and home gateway.

Figure 8 illustrates our home network test system. We used two embedded systems to send IA information and the home gateway will deal with message exchanges. The home gateway used the Linux OS. And the IA system used PHP 4.2.2 and Appach2.0.40. The home gateway Mini database used a txt file. We record information in a fixed format in txt file and used PHP to write the management interface. Figure 9 illustrates our PHP program procedure.

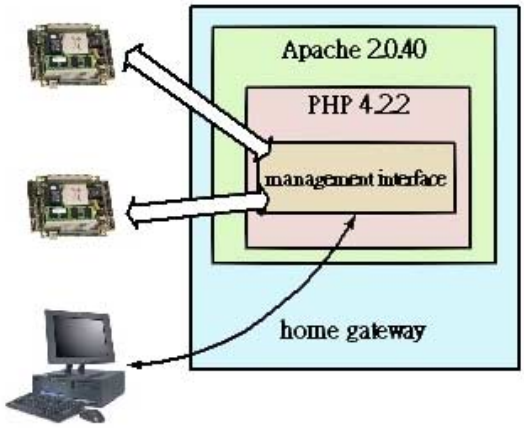

Fig. 8. High comparability management interface test system.

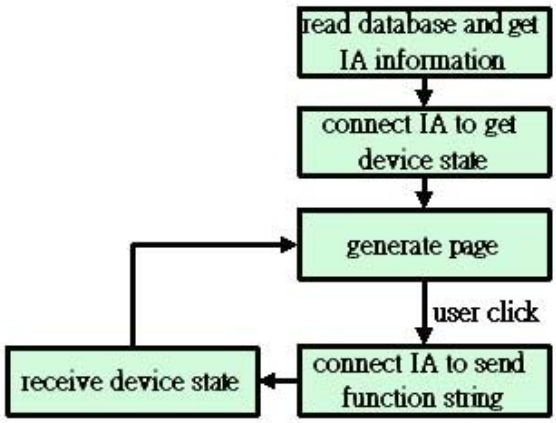

Fig. 9. Management interface program procedure

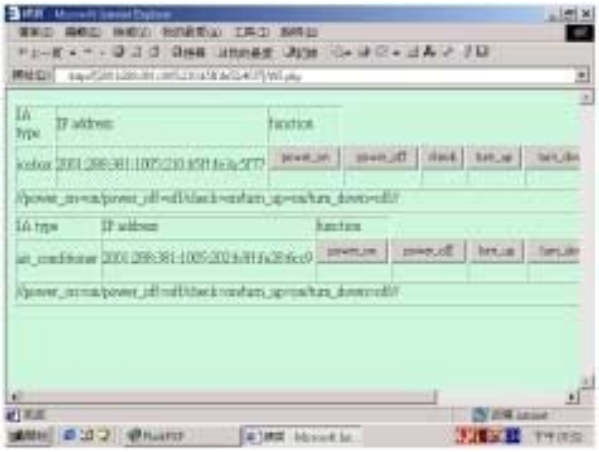

Fig. 10. management interface

All procedures are written on a PHP script page. When the user browses this PHP page, PHP script will read the txt file to acquire the recorded IA information and connect to the embedded system to acquire the device state string. The web server then generates a page to the user's browser. The page is shown in Figure 10.

When the user click any button, the web sever will send the corresponding function string to the relative IP IA address and receive the device state return string. The web server then generates a page and waits for the user to click the next button. Our PHP script has a function that decomposes the device state string and shows the functions of each string button. The user can easily know the state of any appliance device and control that appliance.

3-6 Home Network data stream architecture.

The home network must accommodate large amount of data streams crossing through the home gateway. This data includes voice, video, etc. We conducted an experiment to obtain the network performance analysis. Through analysis graphs (Figure 11), we will prove that the IPv6 wireless architecture is the best home network architecture. In our experiment, home network packet delay time using IPv6 and NAT was compared. The NAT server must waste some time to transfer packets, which affects the system performance according to the total traffic load. When the traffic is heavy, the packet delay time is longer. When many services are running in the home network at the same time the IPv6 network architecture will be the best choice.

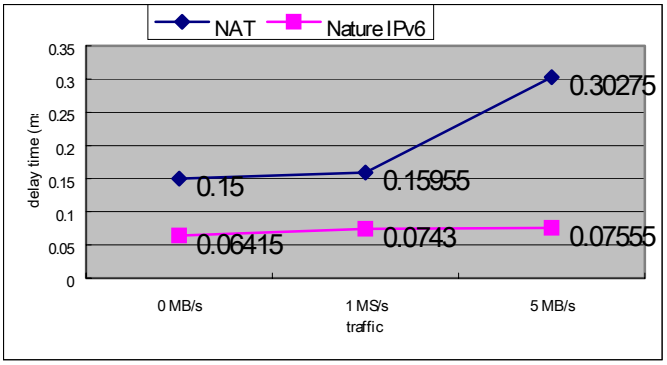

Fig. 11. Packet delay time

IV. Conclusion

To increase user acceptability, we created two methods to auto-configure IA in this study. The first method deals with domain name automatic configuration. The proposed method has the advantage of easily importing new device information. IA manufacturers need only to insert a compiled program into the IA's embedded system. A second method uses the IA device function string to achieve management interface and IA device control. This system requires the embedded system to be designed for receiving and executing commands. IA users can use domain name auto-configuration to connect to the embedded system and control in the initial stage. When IA manufacturers produce IA support for receiving and executing function strings, the user can use the home network management system to control all IA devices through the home gateway management interface. In the future, IA will extend over the world, providing better living for all.

References

[1] S. Deering, and R. Hinden, "Internet Protocol Version 6 (IPv6) Specification", RFC 2460, December 1998.

[2] D. Marples, P. Kriens, "The Open Services Gateway Initiative: an introductory overview", Communications Magazine, IEEE , Volume: 39 Issue: 12 , Dec. 2001 Page(s): 110 –114

[3] P. Dobrev, D. Famolari, C. Kurzke, B. A. Miller, "Device and service discovery in home networks with OSGi", Communications Magazine, IEEE , Volume: 40 Issue: 8 , Aug. 2002 Page(s): 86 –92

[4] P. Vixie, S. Thomson, Y. Rekhter, J. Bound, Dynamic Updates in the Domain Name System (DNS UPDATE), RFC 2136, April 1997

[5] D. Thaler, J. Hagino, IPv6 Stateless DNS discovery, draft-ietf , 08-Mar-02.

[6] Christian Huitema, and John L. Miller, Peer-to-peer name resolution protocol (PNRP) and multilevel cache for use therewith, Patent Number: EP1248441.

[7] H. Kitamura, Domain Name Auto-Registration for Plugged-in IPv6 Nodes, draft-ietf, 23 July 2003.

[8] S. Thomson, T. Narten,IPv6 Stateless Address Auto-configuration, Standards Track Request for Comments (RFC) 2462, IETF, December 1998

[9] R. Want, R., and G. Borriello, "Survey on information appliances", IEEE Computer Graphics and Applications,Volume:203,May-june 2000,page(s):24-31

[10] R. J. C. Nunes, and J. C. M. Delgado,"An internet application for home automation", Electrotechnical Conference, 2000.MELECON 2000.10th Mediterranean, Volume: 1, 2000,Page(s): 298-301

[11] P. Faltstrom "E.164 number and DNS", RFC 2916, September 2000

[12] J. Rosenberg, H. Schulzrinne, and G. Camarillo, "SIP: Session Initiation Protocol", June 2002

Tin-Yu Wu is currently serving as the chief of Operation Division in the University Computer & IT Center at National Dong Hwa University (NDHU) , Hualien, Taiwan, R.O.C. He received his MS degrees in Electrical Engineering from NDHUin 2000. His research interests focus on the next generation Internet protocol, mobile computing and wireless networks. He is now a PhD student in Department of Electrical Engineering, NDHU.

Chia-Chang Hsu was born in Taiwan, ROC, in 1979.. He has his BS degree from the Department of Electronic Engineering, Feng Chai University in July 2002. He is pursuing his MS degree with interests in embedded system and mobile communication at Department of Electrical Engineering, National Dong Hwa University.

Han-Chieh Chao is a Full Professor and Chair of the Department of Electrical Engineering. He is also serving as the director of the University Computer & IT Center at National Dong Hwa University, Hualien, Taiwan, R.O.C. His research interests include High Speed Networks, Wireless Networks and IPv6 based Networks and Applications. He received his MS and Ph.D. degrees in Electrical Engineering from Purdue University in 1989 and 1993 respectively. Dr. Chao is also serving as an IPv6 Steering Committee member and Deputy Director of R&D division of the NICI Taiwan, Co-chair of the Technical Area for IPv6 Forum Taiwan.