Server Support Approach to Zero Configuration In-Home Networking

Kiyohito Yoshihara[1], Takeshi Kouyama[2], Masayuki Nishikawa[2], and Hiroki Horiuchi[1]

[1] KDDI R&D Laboratories Inc., 2-1-15 Ohara Kamifukuoka-shi Saitama 356-8502, JAPAN
{yosshy hr-horiuchi}@kddilabs.jp
[2] KDDI Corporation, 3-10-10 Iidabashi Chiyoda-ku Tokyo 102-8460, JAPAN
{ta-kouyama masa-n}@kddi.com

Abstract. This paper proposes a new server support approach to zero configuration in-home networking. We show three technical issues for zero configuration. Lack of a protocol or technique addressing all issues simultaneously motivated us to design a new approach based on (1) a two-stage autoconfiguration, (2) a UPnP and HTTP-based autoconfiguration, and (3) extended UPnP services. An elaborated flow for the global Internet connection from scratch will be presented. The proposed approach can obtain software and settings from remote servers, and updates/configures for devices. We implemented a system based on the proposed approach, and evaluated its total autoconfiguration time, and the number of technical calls to a help desk during a field trial for five months. We delivered a user-side configuration tool and an all-in-one modem to approximately 230,000 new aDSL subscribers as part of the trial system. Over 40 settings are properly configured for diverse devices in 14 minutes and 10 seconds, while the ratio of the number of calls to the number of new subscribers per month decreased from 14.9% to 8.2%.

1 Introduction

As seen in the number of Internet access subscribers via x Digital Subscriber Line (xDSL) across the globe exceeding 6.3 million by the end of 2003, we can have an always-on broadband Internet connection at home and office as well as at traditionally limited universities or research institutes.

A typical home network for xDSL Internet access is composed of Customer Premises Equipment (CPE) devices including an xDSL modem, residential gateway, and PCs. Before we use Internet applications such as e-mail and Voice over IP (VoIP), it is necessary to configure application and user-specific settings associated with the applications as well as IP network settings, for diverse devices. An e-mail account and Session Initiation Protocol (SIP) server address are examples of such settings. Media-specific settings including an Extended Service Set Identifier (ESSID) and an encryption key must be configured if we use applications

A. Sahai and F. Wu (Eds.): DSOM 2004, LNCS 3278, pp. 232–244, 2004.

via IEEE802.11b[1] Wireless Local Area Network (WLAN) communications. Additionally, software updates are prerequisite for the configuration, for such new software as firmware on an xDSL modem and device driver of a WLAN card may be released for bug fixes or upgrades even after the shipping.

In contrast, the configuration and software update together require a highly skilled and experienced user with technical knowledge of the Internet, as it was initially created for academic purposes. This poses a barrier to Internet novices and raises technical issues. In order to break down this barrier, some protocols and techniques for zero configuration networking [2,3,4,5,6,7,8,9,10,11,12,13,14] have been developed; however, almost all existing protocols and techniques could not fully address this issue: some are only for single and specific devices or applications, and others are restricted to IP network settings, omitting applications and user-specific settings.

This paper proposes a new server support approach to zero configuration in-home networking to solve this issue. The proposed approach allows us to update software on diverse devices and to configure all settings: application and user-specific settings together with IP network settings, required to make available Internet applications. The proposed approach consists of two stages: the former is a Local stage in which a home network is isolated and has only local connectivity, and the latter is a Global stage in which the home network has global Internet connectivity after Local stage. In the Local stage, the proposed approach can discover all devices based on Universal Plug and Play (UPnP) [14] and configure local settings for the devices. In the Global stage, the proposed approach obtains software and settings from servers and customer information systems managed by an Internet Service Provider (ISP) then it updates the software and configures the settings instead of the user. New application and user-specific UPnP actions with the associated state variables are defined and together used in both stages in a secure manner, to cover the shortcomings of the UPnP specifications, which only provided general-purpose items at the development phase.

The emphasis of this paper lies not only in prototyping a system, but also in deploying this system to demonstrate its proven practicality. We implemented a system based on the proposed approach and yet conducted a field trial in which we offer an all-in-one asymmetric Digital Subscriber Line (aDSL) modems with IP router, VoIP, and WLAN access point capabilities to new subscribers, together with a CD-ROM that stores user-side autoconfiguration tools. The tool will automatically configure all required settings for the PC and the aDSL modem with minimum user intervention once a user inserts the CD-ROM into a PC in the home network and clicks the start button. Even an Internet novice can easily have browser access and use e-mail through WLAN communication as well as VoIP with the proposed approach, while an ISP can reduce operation costs through decreasing number of technical calls to the help desk. For proven practicality, we evaluated the proposed approach based on the total processing time of the system, including some software updates, and the number of technical calls to a help desk that were empirically collected during the field trial.

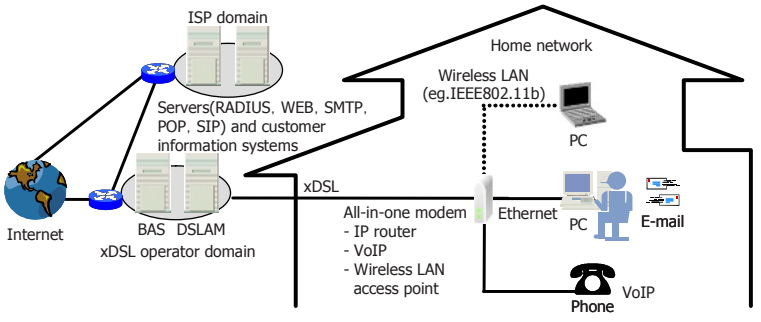


Fig. 1. Typical Home Network with xDSL Connections

This paper is organized as follows: In Sect.2, we present an overview of a typical home network with xDSL connections and address technical issues for zero configuration. We review recent related work in Sect.3. In Sect.4, we propose a new server support approach to zero configuration in-home networking. In Sect.5, we implement a system and evaluate it through a field trial.

2 Typical Home Network with xDSL Connections and Technical Issues for Zero Configuration In-Home Networking

2.1 Typical Home Network with xDSL Connections

Figure 1 shows a typical home network for Internet access via xDSL, with xDSL operator and ISP domains. The CPE devices including an all-in-one xDSL modem, PCs, and phone are connected in a tree with the modem as its root. Each home network is connected to an ISP domain, in which Remote Authentication Dial-In User Service (RADIUS), World Wide Web (WEB), Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), and SIP servers are operated to serve such Internet applications as e-mail and VoIP via Digital Subscriber Line Access Multiplexer (DSLAM) and Broadband Access Server (BAS) installed at an xDSL operator domain. In addition to the servers, customer information systems maintain user account information, and the information should be configured for CEP devices as the application and user-specific settings.

The user must take care of software updates at present. The user voluntarily performs manual updates when they learn of new software releases.

2.2 Technical Issues for Zero Configuration In-Home Networking

The following technical issues should be addressed to achieve zero configuration in-home networking by studying typical home networks in Sect.2.1.

1. All diverse CEP devices shown in Fig.1 should be autoconfigured with minimum user intervention, to release users from the complicated and labor-intensive configuration task (Issue 1).
2. Application and user-specific settings maintained by the servers and customer information systems in a remote ISP domain should be obtained and configured for local CPE devices, to make Internet applications such as e-mail and VoIP available in a simple and easy way (Issue 2).
3. Software updates of CPE devices including firmware on an xDSL modem and device driver of a WLAN card should be performed fully within the autoconfiguration whenever applicable, to run applications as reliably as possible without abnormal terminations of devices due to software bugs (Issue 3).

3 Related Work

Research and development work on the zero configuration networking have been conducted. We summarize recent related work below and show that none of them alone addresses all issues in Sect.2.2 simultaneously.

Dynamic Host Configuration Protocol (DHCP)[2] is a well-known practical protocol. It partially meets Issue 1 in Sect.2.2, in that a DHCP server centrally configures an IP address for diverse IP devices. The server can configure other IP and Domain Name System (DNS) settings; however, neither Issue 2 nor 3 could be addressed only with DHCP as they are restricted to link-local settings while software updates are out of scope.

The Internet Engineering Task Force (IETF) Zero Configuration Networking (zeroconf) working group was standardizing a distributed protocol[3] for dynamic configuration of link-local IPv4 addresses. The IETF Mobile Ad-hoc Networks (manet) working group has also standardized a distributed autoconfiguration protocol[4] for the manet. Although the protocol has inspired subsequent research efforts[5,6,7,8], they are still the same as DHCP for the three issues.

DOCSIS[9] and PacketCable[10] provide similar protocols based on DHCP and Trivial File Transfer Protocol (TFTP) for cable modems. They configure downstream frequency, Class of Service (CoS), etc. for modems. Software updates are also available for DOCSIS protocol. The protocols may meet Issue 3; however, they cannot address Issue 1 and 2 alone, as the intended device of the protocols is a single cable modem, while the configuration is limited to cable and IP-specific settings. This is also true for other effort[11] with Cisco CPE devices.

In terms of media-specific settings, Co-Link configuration technique[12] for wireless links such as IEEE802.11b and Bluetooth has been developed. This technique may meet Issue 1 partially as it introduces configuration point hardware, from which diverse devices can obtain the media-specific settings including an ESSID and encryption key. The technique integrated with the protocols[2,3,4] may achieve autoconfiguration of WLAN communication in a home network; however, even this integration addresses neither Issue 2 nor Issue 3, due to its locality and the lack of the communication software installation and update.

UPnP[14] is designed to support zero configuration networking as devices can join a network dynamically, obtain IP addresses typically with DHCP, and

exchange its presence and capabilities with other devices with Simple Service Discovery Protocol (SSDP)[15]. A service interface defined as an action with state variables is described in XML and is conveyed by Simple Object Access Protocol (SOAP)[16], for controllers or control points to control and transfer data among networked devices. General Event Notification Architecture (GENA)[17] supports eventing, through which control points listen to state changes in devices after subscriptions. Although UPnP may address Issue 1, its service interfaces and typical scope restricted to proximity networking require more work to address Issue 2 and 3. Jini[13] is the same as UPnP for the three issues.

4 Server Support Approach to Zero Configuration In-Home Networking

The findings in Sect.3 motivated us to propose a new server support approach to zero configuration in-home networking to meet all issues in Sect.2.2, which will be presented in the following sections.

4.1 Design Principles and Assumptions

Design Principles. The proposed approach is designed based on the three principles as shown in Fig.2.

1. Two-stage autoconfiguration: Local and Global stages
   a) **Local stage**: In this first stage, the proposed approach configures all required settings: media, IP network, application, and user-specific settings for diverse CPE devices in a carefully-designed flow in order to address Issue 1. The successful completion of this stage enables an intended home network to have global Internet connectivity.
   b) **Global stage**: In this succeeding stage, which is the heart of the proposed approach to address Issue 2 and 3, the approach obtains software and settings from remote servers and customer information systems in an ISP domain after user authentication then it updates software and configures settings for devices.
2. UPnP and HTTP-based autoconfiguration
   The proposed approach leverages UPnP and Hyper Text Transfer Protocol (HTTP), the de-facto standards for the device configuration, to autoconfigure multi-vendor devices comprising a home network for xDSL connection, meeting Issue 1. In particular, we introduce a user-side autoconfiguration tool running on a single device, typically a PC. The device performs as a UPnP control point and autoconfigure itself and all other devices in the intended home network.
3. Extended UPnP services
   Autoconfiguring all required settings only with UPnP standard service interfaces[18] is insufficient as they are given in a generic form and are not

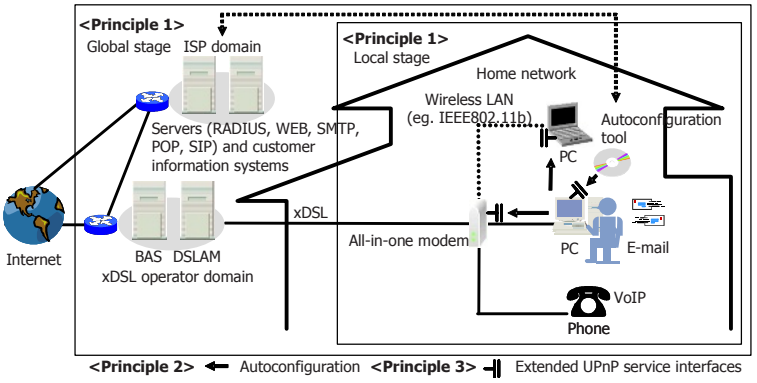


Fig. 2. Principles of Proposed Approach

ready for application and user-specific settings for e-mail and VoIP. We cannot configure all IP network settings with them. We extend UPnP services and define service interfaces so that the proposed approach may autoconfigure all the necessary settings together with the standard ones in order to meet Issue 1 and 2. See Sect.4.3 for details.

Assumptions. We assume the following before and during use of the proposed approach.

1. Application, user-specific settings and new software for updates are registered with servers and customer information systems in an ISP domain.
2. An all-in-one modem and the user-side autoconfiguration tool in removable media are delivered to a user. A password for the modem configuration is preset. The tool recognizes it, but it is treated opaquely.
3. Hardware installation of all intended CPE devices including power and Ethernet cable connections is performed properly.
4. Users initially turn on device power.
5. There are devices that can execute user-side autoconfiguration tools and perform as a UPnP control point in a home network.
6. A DHCP server works in the home network and it configures link-local IP settings containing an IP address, IP subnet mask, default gateway IP address, and DNS server IP address for devices. Recent all-in-one modems normally support a DHCP server and enables it after startup.
7. A WLAN access point conforms to IEEE802.11b/g and broadcasts a beacon including an ESSID periodically if the modem supports WLAN access point capability. The access point permits only authorized access from a device with proper encryption keys.

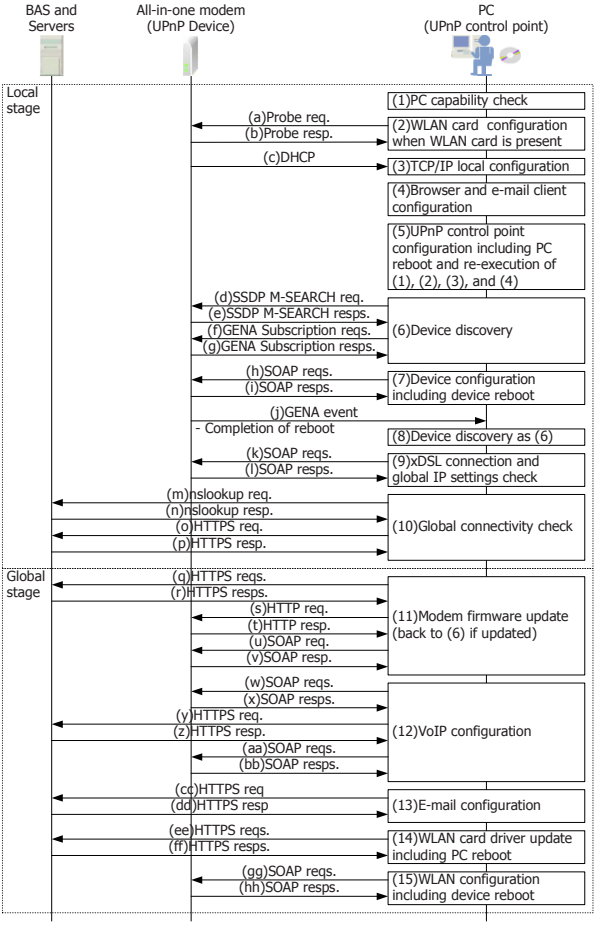


Fig. 3. Autoconfiguration Flow of Proposed Approach

8. A user manually configures an xDSL subscriber account and password, used as a key to associate applications and user-specific settings for modems. An alliance between device vendors and ISP permits modems to be shipped with preset accounts and passwords, thus the user may skip manual configuration.

4.2 Autoconfiguration Flow

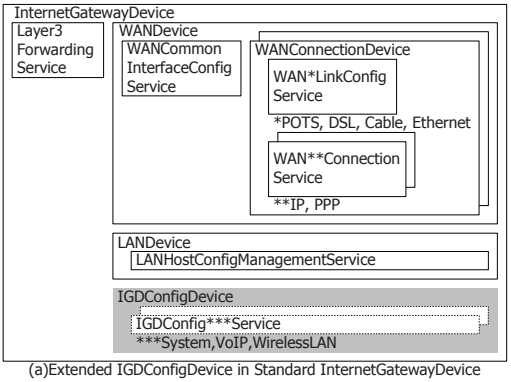
Figure 3 shows autoconfiguration flow of the proposed approach. For simplicity, we suppose a typical home network shown in Fig.1 except that a single PC, an all-in-one modem, and phone constitute the network for an aDSL connection. The flow description assuming the alliance in Sect.4.1 is provided below.

Local Stage. A PC capability check (Fig.3(1)) should be performed first. The autoconfiguration tool checks the OS type and version, login users and their authority, PC hardware specs, HDD free space size, other programs running, active network interface card and/or WLAN card, TCP/IP protocol stacks, and browsers with an e-mail client and the version. The tool exits if any of these are inappropriate or insufficient.

The tool configures the card when an active WLAN card is attached to the PC (Fig.3(2)). It probes an ESSID from the modem. An encryption key is derived from the ESSID with a predefined algorithm. The tool configures these for the card to establish a peer-to-peer link. After that or when the PC is wired, the tool configures link-local IP settings obtained from the DHCP server supported by the modem (Fig.3(3)). In addition, the tool checks and configures dial-up, proxy, and SSL settings for the browser (Fig.3(4)). With the UPnP control point flagged on, the tool reboots the PC to re-execute Fig.3(1) thru (4) to check if the PC has booted and is operating properly (Fig.3(5)).

The tool tries to discover a device or modem (Fig.3(6)) by sending an SSDP M-SEARCH request as a UPnP control point. Then the tool sends GENA subscriptions to the modem to know the completion of the modem reboot required for subsequent new settings. The tool configures aDSL-specific settings: the operation mode (Point to Point Protocol over ATM (PPPoA) or PPP over Ethernet (PPPoE)), the PPPoE bridge option, the connection mode (always-on or on-demand), the encapsulation type (Logical Link Control Encapsulation (LLC) or Virtual Connection (VC)), the pair of Virtual Connection Identifier (VCI) and Virtual Path Identifier (VPI), the encryption method, the PPP keep-alive option, and the PPP retry timer, for the modem for global connectivity (Fig.3(7)). The tool leverages SOAP during configuration. The tool discovers the modem (Fig.3(8)) again then checks if the modem has an expected aDSL connection and obtains global IP settings from a BAS (Fig.3(9)), after modem reboot enabling the above settings is completed. The tool communicates with remote DNS and WEB servers to ensure the global connectivity at the end of the stage (Fig.3(10)).

Global Stage. The tool attempts to update firmware on the modem (Fig.3(11)). The tool asks the remote servers the newest version of the firmware and determines availability. The tool downloads it from the servers and updates it for the modem if the latest firmware is available. Then the tool reboots the modem and goes back to Fig.3(6) after receiving a GENA event describing the completion of modem reboot for the reconfiguration on the most recent firmware. The tool proceeds to the VoIP configuration when the firmware is the latest (Fig.3(12)). The tool checks whether the modem has VoIP capabilities. If it does, the tool downloads VoIP-specific settings from remote servers, and configures them for the modem. The settings contain SIP server address, user name and password for the SIP server, SIP URL, area code, and phone number. The tool goes on to the e-mail configuration (Fig.3(13)). The tool downloads the application-specific settings, and configures them for the e-mail client on the PC as with the VoIP configuration. The settings contain an SMTP server name,



(a)Extended IGDConfigDevice in Standard InternetGatewayDevice

Service interfaces	Operation	Action name
IGDConfig-VoIPService	Set SIP server address	IGDConfigVoIPService X_SetSIPServerAddress
	Set SIP URL	IGDConfigVoIPService X_SetSIPUrl
	Set area code	IGDConfigVoIPService X_SetAreaCode
	Set phone number	IGDConfigVoIPService X_SetVoIPPhoneNumber

(b)Extended Service Interfaces in IGDConfigVoIPService

State variables	object	Variable name
IGDConfig-VoIPService	SIP server address	IGDConfigVoIPService X_SIPServerAddress
	SIP URL	IGDConfigVoIPService X_SIPUrl
	Area code	IGDConfigVoIPService X_AreaCode
	Phone number	IGDConfigVoIPService X_VoIPPhoneNumber

(c)Extended State Variables in IGDConfigVoIPService

Fig. 4. Extended UPnP Services, Interfaces, and State Variables for All-In-One Modem

POP server name, e-mail account, password, user name, e-mail address, and user identifier.

Finally, the tool attempts to update a WLAN card driver (Fig.3(14)), and configures the WLAN-specific settings for the card (Fig.3(15)). The tool obtains the corresponding driver from remote servers, installs the driver including uninstalling the older version, and reboots the PC if the driver update is applicable. The tool configures for the card as appropriate after reboot, assuming a new WLAN card attachment after Fig.3(2).

Although the above flow may somewhat have redundant parts and be still optimized, the highest priority is given to the dependability for more practicality. For example, the second device discovery (Fig.3(8)) is for preventing loss of the event telling the completion of the modem reboot.

4.3 Extended UPnP Services

We extend the standard UPnP services, to achieve the autoconfiguration of all required settings as described in Sect.4.2. As shown in Fig.4 (a), we define an IGDConfigDevice (grayed out area in Fig.4) as a container of the extended three UPnP services: IGDConfigSystem Service, IGDConfigVoIPService, and IGDConfigWirelessLAN Service in the standard InternetGatewayDevice for a typical all-in-one modem. Each of them includes 32, 26, and 19 service interfaces for the configuration of the entire modem, VoIP-specific, and WLAN-specific settings. Note that we can now locate standard service interfaces for the configuration of WLAN-specific settings, while undefined at our development phase.

Figure 4 (b) and (c) shows some service interfaces and state variables of IGDConfigVoIPService. For example, the tool leverages X_SetSIPServerAddress service interface with the state variable X_SIPServerAddress using the desired value as the input argument in order to configure a SIP server address.

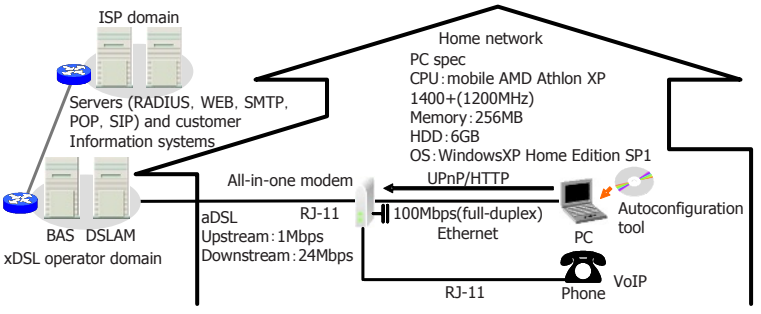


Fig. 5. Evaluation Conditions

5 Implementation and Evaluations

5.1 Implementation

We implement a system based on the proposed approach and describe a brief overview of the system below.

1. We offer the user-side autoconfiguration tool implementing the flow in Sect.4.2 in the CD-ROM as part of the proposed approach.
2. The runtime environment of the tool is WindowsXP. The tool leverages UPnP control point software on WindowsXP installed as default.
3. We embed extended UPnP services in Sect.4.3 and standard ones in the InternetGatewayDevice to a commercially available all-in-one modem. We can configure this modem via UPnP interfaces together with HTTP ones that the modem originally supports.
4. We install a server for the software update in an ISP domain.
5. The tool collects all configuration logs and uploads them to a server after successful completion of the flow in Sect.4.2 (Hereafter referred to as logging). The logs will be used to track future problems and make diagnoses.
6. The tool indicates each process. A user can gain insight into problems even when the user is unable to correct them with the tool. A help desk operator will give advice when the user indicates the problem being experienced.

5.2 Evaluations

The total processing time of the system in Sect.5.1 including all software updates will be evaluated first. After showing our promising results, we deployed the system in Sect.5.1 and deliver the tool and all-in-one modem to new aDSL subscribers in order to empirically verify its real practicality. Then the number of technical calls to a help desk that were collected for five months will be shown.

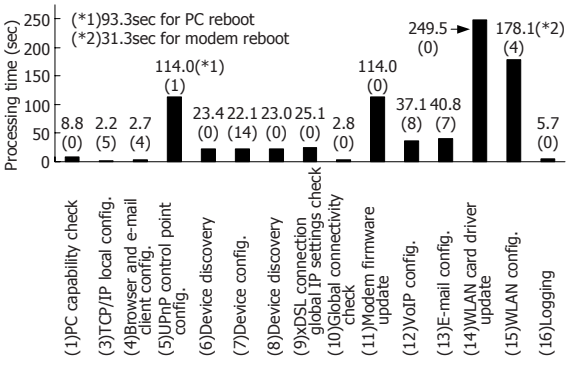


Fig. 6. Processing Time of Proposed Approach

Performance Evaluation. We evaluate the total processing time along the flow in Sect.4.2. Figure 5 shows the evaluation conditions. Note that all steps but (2) in Fig.3 will be performed. Figure 6 shows the results, where the processing time and parenthetic number of settings configured by the tool are shown over the bar for each step.

The 43 settings are properly configured for diverse devices and applications in 14 minutes and 10 seconds. This will be reduced to 8 minutes and 6 seconds if none of the software updates is required. This implies that the proposed approach is suitable for practical use, when we recall that such configuration tasks are error-prone and will generally take more time even for professionals.

Empirical Evaluation. We deployed the system in Sect5.1 and delivered the tool and all-in-one modems to approximately 230, 000 new aDSL subscribers in a field trial for five months. Note that we did not have software updates as the software was the latest version throughout the trial.

The ratio of the number of technical calls to the number of new subscribers per month decreased from 14.9% (November 2003) to 8.2% (March 2004). The decrease in the absolute number of calls and in their total time to the help desk was estimated at 48, 600 and 24, 300 (hours). These were factored from the increase in the number of new aDSL subscribers for the five months and the number of the calls observed just before the trial, assuming no deployment.

The decrease shows that the proposed approach provides both user and provider benefits in that Internet novices can also easily connect and use typical applications, while ISP can reduce operation costs.

6 Conclusions

This paper proposed a new server support approach to zero configuration in-home networking. We showed three technical issues and indicated that none of

related work alone addressed all issues simultaneously. To address these issues, we designed a new approach based on: (1) a two-stage autoconfiguration, (2) a UPnP and HTTP-based autoconfiguration, and (3) extended UPnP services.

To verify practicality, we implemented a system based on the proposed approach and evaluated the total processing time of autoconfiguration including software updates, and the number of technical calls to a help desk that were collected during a filed trial for five months. We delivered the user-side configuration tool and all-in-one modems to new aDSL subscribers as part of the system in the trial. Over 40 settings were properly configured for diverse devices and applications including software updates in 14 minutes and 10 seconds. The ratio of the number of calls to the number of new subscribers per month decreased from 14.9% to 8.2%. These results suggest that the proposed approach is suitable for practical use when we recall that such configuration tasks are error-prone and will generally take more time even for professionals. It provides both user and provider benefits in that Internet novices can also easily connect, while ISP can reduce operation costs via this decrease.

The proposed approach may apply to IPv6 and the cable-based network. Further studies including interworking with other in-home technologies such as HAVi, OGSi or Bluetooth, as well as Web service technologies emerging with UPnP 2.0 are now underway.

Acknowledgment. We are indebted to Mr. Tohru Asami, President & CEO of KDDI R&D Laboratories Inc., for his continuous encouragement for this research.

References

1. IEEE: IEEE Std 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. 1999 ed. (1999)
2. Droms, R.: Dynamic Host Configuration Protocol. IETF, RFC 2131. (1997)
3. Cheshire, S., Aboba, B., Guttman, E.: Dynamic Configuration of Link-Local IPv4 Addresses. IETF draft-ietf-zeroconf-ipv4-linklocal-14.txt. (2004)
4. Perkins, C., Malinen, J., Wakikawa, R., Belding-Royer, E., Sun, Y.: IP Address Autoconfiguration for Ad Hoc Networks. IETF draft-ietf-manet-autoconf-01.txt. (2001)
5. Misra, A., Das, S., McAuley, A.: Autoconfiguration, Registration, and Mobility Management for Pervasive Computing. IEEE Personal Commun. 8 (2001) 24–31
6. Weniger, K., Zitterbart, M.: IPv6 Autoconfiguration in Large Scale Mobile Ad-Hoc Networks. In: Proc. of European Wireless 2002. (2002) 142–148
7. Nesargi, Prakash, R.: MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network. In: Proc. of IEEE INFOCOM 2002. (2002) 1059–1068
8. Zhou, H., Ni, L.M., Mutka, M.W.: Prophet Address Allocation for Large Scale MANETs. In: Proc. of IEEE INFOCOM 2003. (2003) 1304–1311
9. CableLabs: DOCSIS 2.0 Specifications: Operations Support System Interface Specification. (2004)
10. PacketCable^{TM}: CMS Subscriber Provisioning Specification. (2002)

11. Shen, F., Clemm, A.: Profile-Based Subscriber Service Provisioning. In: Proc. of IEEE/IFIP NOMS2002. (2002)
12. Tourrilhes, J., Krishnan, V.: Co-link configuration : Using wireless diversity for more than just connectivity. Technical Report HPL-2002-258, HP Labs. (2002)
13. SUN Microsystems: Jini™ Architecture Specification Version 2.0. (2003)
14. UPnP Forum: Universal Plug and Play™ Device Architecture. (2000)
15. Goland, Y., Cai, T., Leach, P., Gu, Y., Albright, S.: Simple Service Discovery Protocol/1.0 Operating without an Arbiter. IETF draft-cai-ssdp-v1-03.txt. (1999)
16. World Wide Web Consortium: SOAP Version 1.2. (2003)
17. Cohen, J., Aggarwal, S., Goland, Y.: General Event Notification Architecture Base: Client to Arbiter. IETF draft-cohen-gena-p-base-01.txt. (2000)
18. UPnP Forum: Internet Gateway Device (IGD) Standardized Device Control Protocol V1.0. (2001)