

NAT Traversal for Pure P2P e-Learning System

Kazuo Kuramochi
Graduate School of Engineering
Tottori University
4-101, Koyama-Minami
Tottori, JAPAN
Email: s032021@ike.tottori-u.ac.jp

Takao Kawamura and Kazunori Sugahara
Faculty of Engineering
Tottori University
4-101, Koyama-Minami
Tottori, JAPAN
Email: {kawamura,sugahara}@ike.tottori-u.ac.jp

Abstract—In this paper, we present Network Address Translator (NAT) traversal technique suitable for pure P2P applications, such as a distributed asynchronous Web-Based Training (WBT) system that we have previously proposed. P2P systems have the advantages of scalability and robustness, however, P2P systems face difficulties in constructing its network over the Internet, since end-to-end communication through NATs is sometimes restricted. In this study, as a solution of this issue, we ensure uniqueness of identification of nodes through the Internet and enable to establish a connection over the NAT by means of adding static routing entry to the NAT table with the help of Universal Plug and Play (UPnP). Our NAT traversal technique is implemented in the proposed WBT system and is examined by experiments.

I. INTRODUCTION

Nowadays, e-Learning systems, especially asynchronous Web-Based Training systems (hereafter we abbreviate as WBT) are very popular. A WBT allows a learner to complete the WBT on his own time and schedule, without live interaction with the instructor. Although a large number of studies have been made on asynchronous WBT [1], [2], [3], all of them are based on the client/server model. Client/server systems have the advantage of ease of construction and maintenance, however, client/server systems generally lack scalability and robustness. In recent years, P2P research has grown exponentially. Although the current P2P systems are well-known for its file sharing ability, and the consequent legal problems, P2P systems are gradually proving themselves to be a very promising area of research. Because they have potential for offering a decentralized, self-sustained, scalable, fault tolerant and symmetric network of computers providing an effective balancing of storage and bandwidth resources. On the other hand, P2P systems face Network Address Translator (NAT) [8] traversal problem to be described later.

We have proposed and implemented a distributed e-Learning system based on P2P architecture [4], [5] using Maglog that is a Prolog-based framework for building mobile multi-agent systems that we have also developed [6], [7]. The proposed e-Learning system has two distinguishing features. Firstly, it is based on P2P architecture to improve the scalability and robustness of the system. In the proposed e-Learning system, every user's computer plays the role of a client and a server. Namely, while a user uses the system, his/her computer (hereafter we refer to such a computer as a node) is a part of the system. It receives some number of contents from another

node when it joins the system and has responsibility to send appropriate contents to the requesting nodes. In addition to the above advantages of using P2P architecture, we can construct the proposed e-Learning system at low cost because the system need no server computers.

Secondly, each exercise in the system is not only data but also an agent so that it has functions, such as scoring user's answers, telling the correct answers, and showing some related information without human instruction.

In this paper, we present NAT traversal technique suitable for pure P2P applications, such as the proposed distributed e-Learning system. We aim to construct the proposed system consists of a number of nodes across the Internet, however, communications over NATs are sometimes restricted. Therefore, we cannot construct the proposed system which consists of nodes, both behind and outside the NAT. We solve this problem in decentralized manner by ensuring uniqueness of identification of nodes through the Internet and enabling communications over the NAT by means of adding static routing table entry to the NAT.

This paper is organized in 6 sections. The proposed e-Learning system is described in Section II. We describe the design overview of the NAT traversal for the proposed system in Section III and the implementation of the NAT traversal in Section IV, respectively. In Section V, we present experimental results. Finally, some concluding remarks are drawn in Section VI.

II. PROPOSED E-LEARNING SYSTEM

A. Overview

All exercises in the proposed system are classified into categories, such as "English/Grammar", "Math/Statistic", and "History/Rome", etc. A user can obtain exercises one after another through specifying categories of the required exercises.

While a user uses the proposed e-Learning system, his/her computer is a part of the system. Namely, it receives some number of categories and exercises from another node when it joins the system and has responsibility to send appropriate exercises to requesting nodes. The important point to note is that the categories a node has are independent of the categories in which the node's user are interested, as shown in Fig. 1. Fig. 1 illustrates that user A's request is forwarded first to the

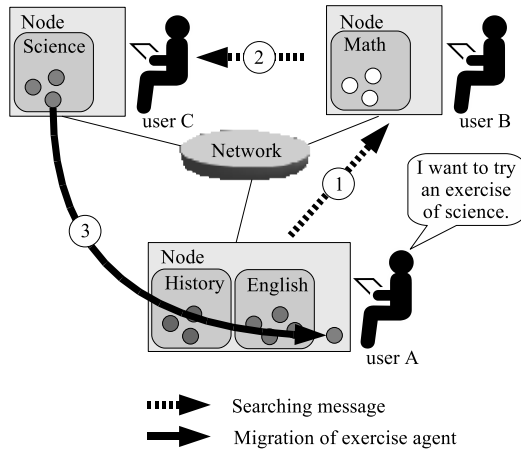


Fig. 1. Proposed e-Learning system.

neighbor node, and next the request is forwarded to the node which has the requested category.

B. P2P network

When the proposed system bootstraps, one initial node has all categories in the system. When another node joins the system, it is received certain number of categories from the initial node. The categories are distributed among all nodes in the system according as nodes join the system or leave the system.

We would like to emphasize that in existing P2P-based file sharing systems, such as Napster [9], Gnutella [10], and Freenet [11], each shared file is owned by a particular node. Accordingly, files are originally distributed among all nodes. On the other hand, the categories in the proposed system are originally concentrated. Consequently, when a new node joins the system, not only location information of a category but the category itself must be handed to the new node. Considering that, the P2P network of the proposed system can be constructed as a CAN [12].

Our P2P network is constructed with 2-dimensional coordinate space $[0,1] \times [0,1]$ to store exercise categories, as shown in Fig. 2. The figure shows the situation that node C has just joined the system as the third node. Before node C joins, node A and node B shared the whole coordinate space half and half. At that moment, node A managed “Math/Geometry”, “Math/Statistics”, and “History/Rome” categories and node B managed “English/Grammar”, “English/Reader” and “History/Japan” categories, respectively. When node C joins the system, we assume node C knows IP addresses of some nodes already in the system and node C send the join request to some node in the list. Then node C is mapped on a certain coordinate space according to a random number and takes on corresponding categories from another node. For example, in the case of Figure 2, node C takes on the “History/Japan” category from node B and exercises of the category move to node C. After joining, node C has a list of IP addresses of

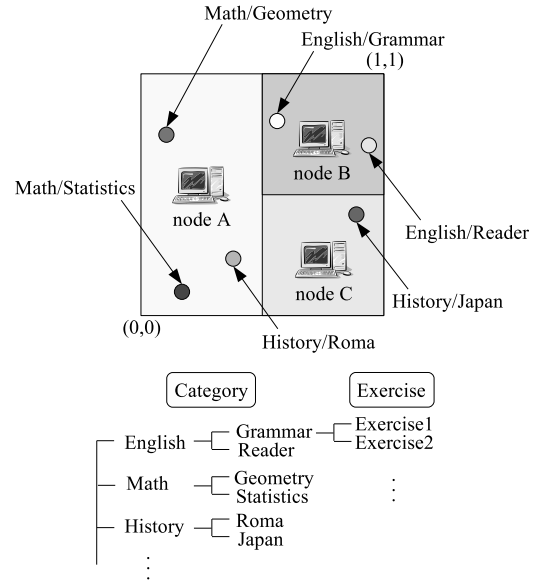


Fig. 2. P2P network of the proposed e-Learning system.

neighbor nodes in the coordinate space, such as node A and node B. Therefore, neighbor nodes can communicate with each other.

Note that our P2P network can be categorized as “pure P2P”. While some P2P applications such as Napster and Skype [13], [14] need a centralized server or super nodes, all nodes in the proposed system act as equals.

C. Components

For the proposed system that can be considered as a distributed WBT system, it is not enough that only exercises are distributed among all nodes. Functions to provide the above services also must be distributed among all nodes. We adopt mobile agent technology to achieve this goal.

There are following agents and user interface programs on each node.

- **Exercise Agent:** Each exercise agent has a question and functions to score user’s answers, tell the correct answers, and show some related information about the exercise.
- **Category Agent:** Each category agent stands for a unit of a particular subject. It manages exercise agents in itself and sends them to the requesting node.
- **Node Agent:** Each node has one node agent. It manages the zone information of a CAN and forwards messages to the category agents in the node.
- **Student Interface:** One student interface is on each node of which a user logs in as a student. It is a user interface program for studying.

III. OUR NAT TRAVERSAL APPROACH

A. Problems and solutions

As mentioned previously, the proposed system is intended for the use over the Internet by nodes in various networks,

such as “home networks” and “corporate networks”. However, we cannot construct the P2P network of the proposed system over NATs, since agent migration between nodes is sometimes restricted by NAT. This problem is known as NAT traversal problem.

A NAT is widely used in the Internet to share global IP addresses among nodes on a private network. A NAT makes it possible for a node with a private IP address behind the NAT to connect to a node with a global IP address by translating IP address and port number of packets. A NAT is suitable for client/server application since a client node behind the NAT can establish outbound connections to a server node which have a globally unique IP address. By contrast, the NAT is not suitable for P2P application, which needs end-to-end communication, because nodes behind the NAT typically cannot receive inbound connections.

In the proposed system, agents migrate among nodes. When an agent on node N_1 migrates to node N_2 , N_1 establishes TCP connection to N_2 and N_1 sends programs and data of the agent over the connection. However, the connection is sometimes refused by the previously stated operation of NAT.

Until now, several NAT traversal techniques, which enable connections over NAT, are proposed. Some nodes connected with a globally reachable server can communicate with each other by relaying through the server. This is most reliable technique, however, it consumes the server’s processing power and bandwidth. UDP hole punching techniques are proposed for UDP communication over NAT. The basic concept of UDP hole punching is that each node behind a NAT connects a server to make the NAT translation state and register endpoint information of the node to the server, and then the node connects with each other using the endpoint entry on the server. STUN [15] presents the detailed protocol for UDP hole punching. For TCP communications, some TCP hole punching techniques are also proposed in [16], [17]. However, the above NAT traversal techniques are not suitable for pure P2P applications, such as the distributed e-Learning system, since these techniques require a globally reachable node for NAT traversal.

To solve previously described issue caused by a NAT in decentralized manner, we apply following methods to the proposed e-Learning system.

- Ensuring uniqueness of identification of nodes, both assigned globally routable IP address and assigned private IP address.
- Enabling establishment of two-way connection between nodes behind a NAT and nodes outside the NAT by means of adding a static routing table entry to the NAT.

B. Identification of nodes in networks with NATs

Several applications in the Internet use IP address as identifier of nodes. However, identifying a node only by IP address cannot ensure uniqueness of nodes when we make up the proposed system consists of nodes, both with a global IP address and with a private IP address. To ensure uniqueness

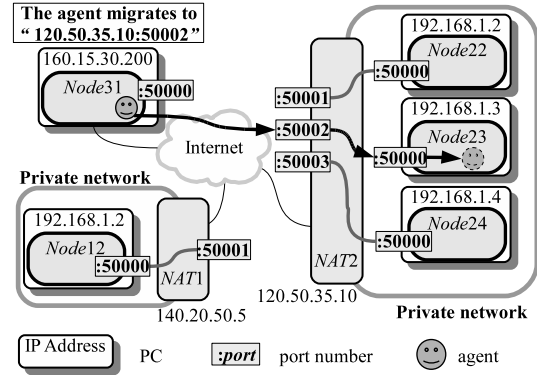


Fig. 3. Agent migration between a node with a global IP address and a node with a private node behind NAT.

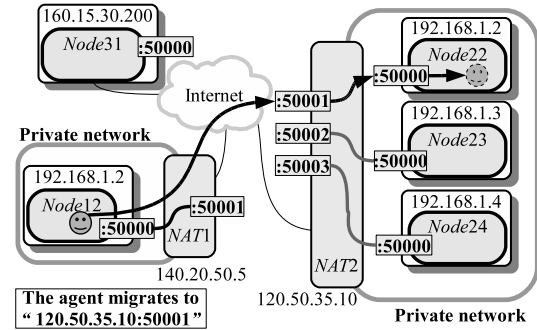


Fig. 4. Agent migration between nodes behind different NATs.

of these nodes, we use a pair of a global IP address and a port number.

If a node is assigned a global IP address, the node use a pair of the global IP address and the port number of the process for the proposed system as its identifier, otherwise the node use a pair of an external IP address and the port number of the NAT corresponds to the process on the node behind the NAT as its identifier.

C. Agent migration over a NAT

We enable communications over a NAT by adding a static routing table entry to the NAT for allowing inbound connection to a particular host and port behind the NAT, before joining the P2P network of the proposed system. To add the NAT table entry without human instruction, we use Universal Plug and Play (UPnP) [18] if it is enabled on the NAT.

D. Example of agent migration using our NAT traversal

We give examples of three communication scenarios: (1) communications between a node with a global IP address and a node with a private IP address behind NAT, (2) communications between nodes behind different NATs, and (3) communications between a common NAT.

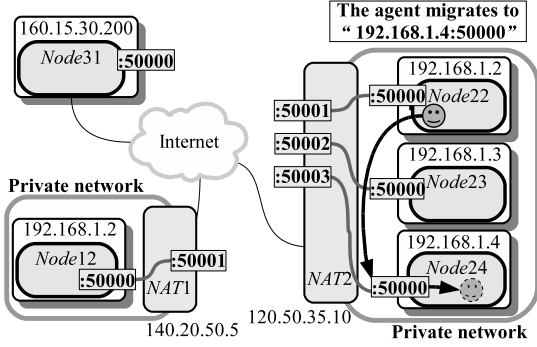


Fig. 5. Agent migration between nodes behind a common NAT.

First, consider the agent migration from a node assigned global routable IP address to a node behind NAT $NAT2$, as shown in Fig. 3. When an agent on node $Node31$ migrates to node $Node23$, its identifier is $120.50.35.10:50002$, the destination address and port number of packets for the migration are translated into $192.168.1.3:50000$ by $NAT2$. Then the connection for the migration is established.

Suppose the migration from a node behind NAT $NAT1$ to a node behind $NAT2$, as shown in Fig. 4. When an agent on node $Node12$ migrates to node $Node22$, its identifier is $120.50.35.10:50003$, the source address and port number on packets are translated into a global IP address of $NAT1$, $140.20.50.5$, and a corresponding port, and the destination address and the port number are also translated into $192.168.1.4:50000$ by $NAT2$. Then the connection is established, too.

Fig. 5 illustrates the migration from a node behind $NAT2$ to a different node behind $NAT2$. In [16], the authors report that support for “hairpin translation”, translating a pair of (external IP address of the NAT, port) on packets from inside NAT into (private IP address, port) of corresponding node inside the same NAT, is much less common among existing NAT routers. Therefore, we use private IP address and port number of the node as destination of the connection when the source node and the destination node are behind a common NAT. When an agent on $Node22$ migrates to node $Node24$, $Node22$ establishes a connection using destination address and port $192.168.1.4:50000$.

Note that our NAT traversal approach does not cover communications between nodes over multiple levels of NAT.

IV. IMPLEMENTATION OF THE PROPOSED SYSTEM WITH NAT TRAVERSAL

A. Overview of the proposed system

Fig. 6 shows the implementation overview of the proposed e-Learning system. Each node runs an agent server written in Java and all agents on a node are threads in the agent server process. The student interface is implemented as a plug-in for Firefox web browser.

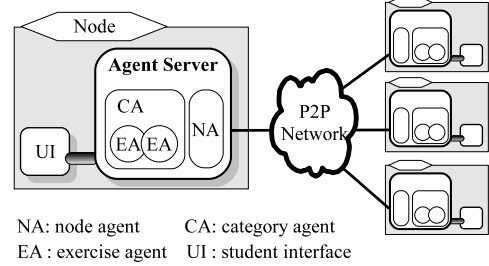


Fig. 6. Implementation overview of the proposed e-Learning system.

B. Communication over a NAT

To construct the proposed system consists of nodes, both behind and outside a NAT, we implement following procedures.

When a user starts using the proposed e-Learning system, an agent server is started firstly. The start-up procedure of an agent server is as follows.

- 1) If the node is assigned a private IP address, then perform following steps:
 - a) Add an entry shown in Table I to a NAT in the private network.
 - b) Set an identifier of the agent server to a pair of an external IP address of the NAT and a port number configured in the above step.
- 2) If the node is assigned a global IP address, then set an identifier of the agent server to a pair of the global IP address and a port number of the agent server.

TABLE I
A ENTRY OF THE NAT TABLE FOR INBOUND TCP CONNECTIONS TO AN AGENT SERVER ¹.

a node outside NAT	NAT	a node inside NAT
* : *	$NAT_IP : n$	$P_IP : m$

In a step 1b of above procedures, if the NAT supports UPnP, then the agent server do the step 1b using UPnP, otherwise the step 1b is accomplished manually. With above procedures, nodes, both behind and outside the NAT, are uniquely identified in the proposed system, and then these nodes are ready for communicating each other over the NAT.

If an entry of the NAT table is added at the start-up time of the agent server, the static routing table entry should be deleted from the NAT at the time of halting the agent server. If the NAT supports UPnP, then the agent server delete the entry through the UPnP service.

¹ $ip : port$ stands for a pair of an IP address ip and a port number $port$, * stands for an arbitrary value, NAT_IP stands for an external IP address of a NAT, P_IP stands for a private ip address of the host of the agent server, n stands for a port number not equals to the value of existing NAT table entry, and m stands for a port number of agent server.

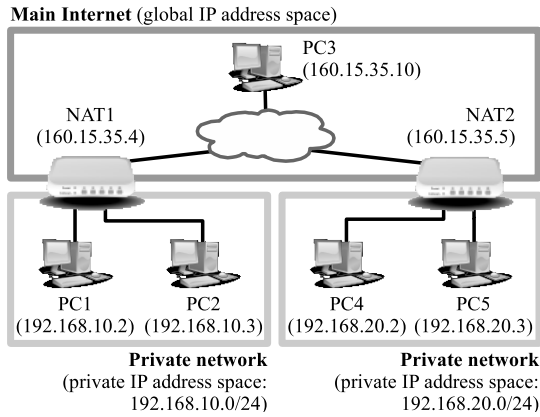


Fig. 7. Network layout of the experiments.

V. EXPERIMENTS

This section presents the experimental results. The experimental environment consists of 5 PCs and 2 NAT routers. All PCs have Pentium4 2.66GHz processor and 512MB of RAM and are running on GNU/Linux (kernel version is 2.4.27) operating system. The version of Java runtime environment is 1.5.0. The NATs support UPnP and the UPnP services on the NATs are enabled. The physical network layout is shown in Fig. 7. We use 9 categories ($C1-C9$) and 10 exercises on each category in the experiments.

In the experiments, we examine the agent migration of the proposed system over a NAT. First, we let the proposed e-Learning system start up on PC1, and then make PC2-PC5 join the system. PC2-PC5 knows a pair of IP address and port of PC1 as an entry of known nodes, i.e., PC2-PC5 send the join request to PC1. Categories on each node every time a node joins are shown in Fig. 8. The results indicate categories are transferred from PC3 to PC4, from PC1 to PC2 and from PC2 to PC5. In other words, agents migrate over the NAT with our NAT traversal technique even if from a nodes outside a NAT to a node behind the NAT, from a node to another node behind a common NAT and from a node behind a NAT to a node behind another NAT.

Second, we try exercises using the proposed system on PC1, PC3 and PC5, after above experiment. Categories of retrieved exercises are $C5$, $C2$, and $C6$, mapped onto the zone of PC2, PC3 and PC5, respectively. As a result of this examination, we can try exercises at all times. The examination also indicates that each exercise agent migrates at all times of trying exercises even if between nodes over the NAT.

VI. CONCLUSION

Since existing asynchronous WBT systems are based on the client/server model, they have problems of scalability and robustness. The proposed e-Learning system solves these problems in decentralized manner through both P2P technology and mobile agent technology.

In this paper, we have presented the NAT traversal technique suitable for pure P2P based applications, such as the proposed

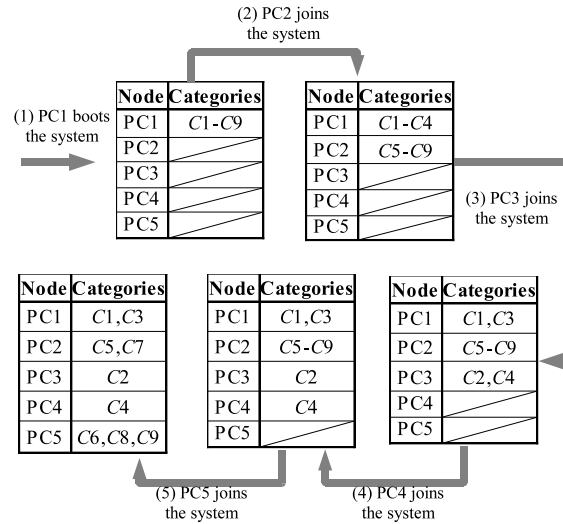


Fig. 8. Experimental result: categories on each node.

e-Learning system, using following methods: (i) ensuring uniqueness of identification of nodes, both assigned global IP address and assigned private IP address using a pair of a global IP address and port number, and (ii) enabling establishment of two-way connections between nodes behind and outside the NAT by means of adding a static routing table entry to NAT with the help of UPnP if it is enabled on the NAT. The proposed e-Learning system including NAT traversal technique was examined by experiments.

We provided students practice exercises for “The Information Technology Engineers Examination” conducted by Information-Technology Promotion Agency, Japan (IPA) through the proposed e-Learning system in classroom. As a result of applying our NAT traversal technique to the proposed e-Learning system, we can construct the e-Learning system over the Internet at low cost. More practical experiments are needed in the Internet to confirm the effectiveness of the proposed e-Learning system with the NAT traversal technique.

REFERENCES

- [1] T. Nishita and et el, “Development of a web based training system and courseware for advanced computer graphics courses enhanced by interactive java applets,” in *Proceedings of International Conference on Geometry and Graphics*, vol. 2, 8 2002, pp. 123–128.
- [2] H. Homma and Y. Aoki, “Creation of wbt server on digital signal processing,” in *Proceedings of 4th International Conference on Information Technology Based Higher Education and Training*, 7 2003, marrakech, Morocco.
- [3] D. Helic, H. Krottmaier, H. Maurer, and N. Scerbakov, “Enabling project-based learning in wbt systems,” *International Journal on E-Learning*, vol. 4, no. 4, pp. 445–461, 2005.
- [4] T. Kawamura and K. Sugahara, “A mobile agent-based p2p e-learning system,” *IPSJ Journal*, vol. 46, no. 1, pp. 222–225, 1 2005.
- [5] S. Motomura, R. Nakatani, T. Kawamura, and K. Sugahara, “Distributed e-learning system using p2p technology,” in *Proceedings of the 2nd International Conference on Web Information Systems and Technologies*, 4 2006, pp. 250–255, setubal, Portugal.

- [6] S. Motomura, T. Kawamura, and K. Sugahara, "Maglog: A mobile agent framework for distributed models," in *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Systems*, 11 2005, pp. 414–420, phoenix, Arizona, USA.
- [7] —, "Logic-based mobile agent framework with a concept of "field"," *IPSI Journal*, vol. 47, no. 4, pp. 1230–1238, 4 2006.
- [8] P. Srisuresh and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)," IETF RFC 3022, 1 2001.
- [9] Napster, "<http://www.napster.com/>," 1999.
- [10] Gnutella, "<http://welcome.to/gnutella/>," 2000.
- [11] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," <http://freenetproject.org/papers/freenet.pdf>, 2000.
- [12] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "A scalable content-addressable network," in *Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications*. ACM Press, 2001, pp. 161–172.
- [13] Skype, "<http://www.skype.com/>," 2007.
- [14] S. A. Baset and H. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol," in *Proceedings of 25th IEEE International Conference on Computer Communications*, 4 2006, barcelona, Spain.
- [15] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)," IETF RFC 3489, 3 2003.
- [16] B. Ford, P. Srisuresh, and D. Kegel, "Peer-to-peer communication across network address translators," in *Proceedings of the 2005 USENIX Annual Technical Conference*, 4 2005, pp. 179–192, anaheim, California, USA.
- [17] S. Guha, Y. Takeda, and P. Francis, "NUTSS: A SIP based approach to UDP and TCP connectivity," in *Proceedings of Special Interest Group on Data Communications (SIGCOMM) Workshops*, Portland, Oregon, USA, 8 2004, pp. 43–48.
- [18] UPnP Forum, "Upnp™ standards," <http://www.upnp.org/standardizeddcps/>, 2007.