2012 International Conference on Computer Technology and Science (ICCTS 2012)
IPCSIT vol. 47 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V47.37

LUIIS: Local UPnP IGD Interworking Scheme for NAT Traversal

Chien-Chao Tseng and Chia-Liang Lin [+]

Department of Computer Science, National Chiao Tung University

Abstract. Peer-to-peer (P2P) communication has emerged as the mainstream of network applications. However, Network Address Translation (NAT) is a barrier to P2P applications and induces NAT traversal problems. Furthermore, some NAT devices disable the Universal Plug and Play (UPnP) service for security reason and some P2P applications support only UPnP for NAT traversal. This paper proposes a Local UPnP Internet Gateway Device (IGD) Interworking Scheme (LUIIS) for P2P applications to traverse NATs. When a P2P application needs to traverse a NAT, it can use UPnP to obtain an effective transport address from LUIIS. Furthermore, LUIIS does not require modification to the P2P applications that support UPnP. The test results verified that LUIIS can help hosts behind NATs establish communication paths.

Keywords: Network Address Translation, NAT, NAT Traversal, Universal Plug and Play, UPnP.

1. Introduction

Peer-to-peer (P2P) communication has emerged as the mainstream of network applications and has gained immense popularity in recent years. P2P communication can not only avoid the expense but also shorten the delay of handling traffic at a server. Voice over internet protocol (VoIP) is one of the most common P2P applications. However, this style of communication often has problems dealing with Network Address Translation (NAT) [2].

NAT is a solution to alleviate the exhaustion of IPv4 address. By modifying network address information stored in packet header when packets pass through a traffic routing device, NAT remaps a given address realm into another, while also providing transparent routing for the hosts behind a NAT. The nature of NAT causes NAT traversal problem [6], which is a barrier to P2P applications. Not until an internal host (IH) behind a NAT device sends a packet to an external host (EH) outside the NAT first can the EH send packets to the IH directly. In other words, NAT device typically blocks session requests originating from the external side, which prevents the establishment of P2P sessions. The situation becomes worse when both hosts are behind different NAT devices. As a remedy, many NAT traversal techniques [4, 5, 6, 7] have been proposed to establish and maintain TCP/IP network sessions across NAT devices. NAT traversal is indispensable for P2P applications running in NAT environment.

Many existing NAT traversal schemes rely on a server with publicly routable IP addresses. Some schemes only use the server when establishing a session (such as STUN [3]). Some relay all data through the server (such as TURN [5]), but this approach increases both bandwidth costs and latency. These relaying schemes are also detrimental to real-time voice and video communication. Unlike previous schemes, Universal Plug and Play (UPnP) extends plug-and-play, which is a technology for dynamically connecting devices to a computer through network, to provide further assistance with NAT traversal. With UPnP, a device can dynamically join a network, retrieve the external IP address of the NAT device, enumerate existing port mappings, and add or remove port mappings. However, some NAT devices disable the UPnP service for security reason and some P2P applications support only UPnP for NAT traversal. As a result, these P2P applications cannot work properly with the presence of NAT.

[+] Corresponding author. Tel.: +886-3-5712121 ext.54792; fax: +886-3-5721490.
E-mail address: cllin@cs.nctu.edu.tw.

This article aims to propose a Local UPnP Internet Gateway Device (IGD) Interworking Scheme (LUIIS) to traverse NAT while providing UPnP service for P2P applications. LUIIS detects the presence of IGD on the network, collects the NAT information about the mapping behaviour and filtering behaviour, decides if the mapped-address is valid and acquire a TURN resource when the mapped-address is invalid. When a P2P application needs to traverse a NAT, it can use UPnP to obtain an effective transport address from LUIIS. Then the application can receive packets sent from the remote peer with such transport address.

The remainder of this article is organized as follows. We first describe mapping and filtering rules of a NAT device in detail, and then introduce three common NAT traversal schemes. In the following sections, we describe the design of LUIIS and present test results. Finally, we summarize our findings and provide suggestions for further research in the final section.

2. Literature Review

2.1. Network Address Translation

NAT allows IHs within a private network to connect to EHs in a public network [2]. The usage and toleration of NAT ameliorates IPv4 address depletion by allowing globally registered IP addresses to be either reused or shared among several hosts. Network Address Port Translation (NAPT) is a commonly-adopted NAT implementation, which allows many hosts to share a single IP address through multiplexing streams differentiated by a TCP/UDP port number.

In the rest of this paper, NAT refers to NAPT implementation, and a mapped-address is an external global IP address along with a port number allocated by a NAT for a connection attempt from an IN. Different NATs may differ in NAT mapping and/or packet filtering behaviours. These behaviours are discussed in details in the following.

NAT mapping behaviour, which refers to how mapped-addresses are allocated to connections. A mapping chooses an external address and a port for each session. Sessions originating from different IHs are certainly allocated different mapped-addresses. For connections that originate from the same IH but are destined for different hosts and/or different ports, mapping policies can be independent mapping, address dependent mapping, or address-and-port dependent mapping [3]. Independent mapping NAT uses the same address and port for all outbound packets originating from the same IH regardless of destination address and port (Fig. 1ai). Address dependent mapping differentiates connections that are destined for different EHs. For packets to the same destination address, NAT uses the same mapped-address (Fig. 1aii). Finally, address-and-port dependent mapping NAT generates one unique mapped-address for each connection (Fig. 1aiii).Beside these polices, some NATs generate mapped-addresses in a random fashion while some others do so in a sequential order.

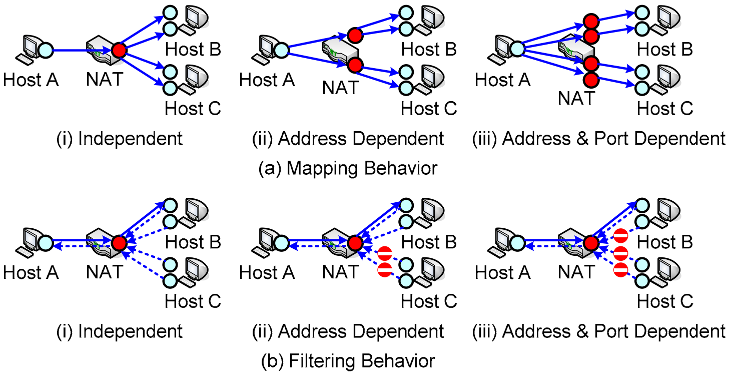


Fig. 1: NAT mapping and filtering rules: a) mapping; b) filtering.

Packet filtering behaviour, which determines whether a packet sent form the external side is allowed to pass through the NAT. The judgment is based on the source address and port number. The authors in [3] classified packet filtering behaviour into independent filtering, address dependent filtering, and address-and-port dependent filtering. In independent filtering, any EH regardless of its address and port number can send packets to IHs with valid mapped-addresses (Fig. 1bi). Address dependent filtering only accepts packets sent from an EH for which a mapped-address (Fig. 1bii) has been created previously. In address-and-port dependent filtering, packets originating from a host with address B and port number b are allowed only if a mapped-address has already been created for that address and port (Fig. 1biii).

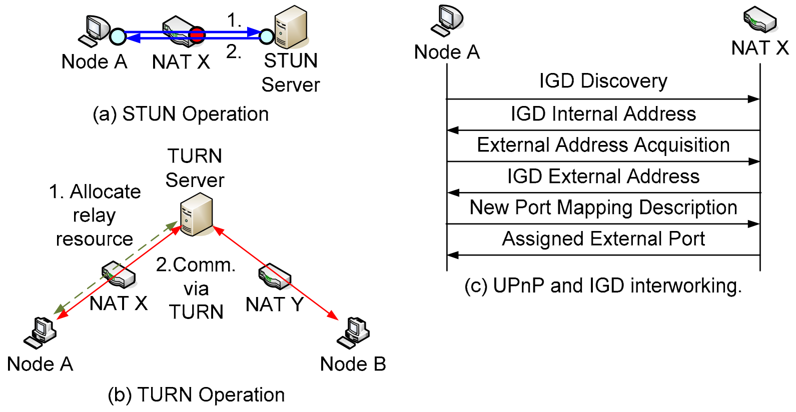
2.2. Related work

The key point of NAT traversal is hole punching [4]. If an IH wants to receive packets from EHs, the IH must create a mapped-address for an inbound session. For every incoming packet, a NAT table must also list a binding and state. Otherwise, it cannot find the mapped IH to which the packet belongs. Though there are different techniques for solving the NAT traversal problem, no single method provides a solution that works well with all NAT applications and network topologies. This subsection reviews the existing techniques, including STUN [4], TURN [5] and UPnP [6].

STUN is a suite of schemes, one of which includes a network protocol used in NAT traversal for applications such as real-time voice, video, messaging, and other types of interactive IP communication. As Fig. 2a illustrates, The STUN protocol requires assistance from a 3rd-party STUN server located on the external side of the NAT. This server allows applications behind a NAT to discover the presence of a NAT. It also helps obtain the mapped public IP address and port number that NAT allocates for the application's User Datagram Protocol (UDP) connections to the remote hosts.

TURN utilizes a relay node in the public domain to bridge together two connections independently created by two hosts behind different NATs. As Fig. 2b illustrates, Node A connects to a TURN server to request relay resources and inform Node B of the relay resource. Once two hosts wish to communicate with each other, they can relay their data through the TURN server.

UPnP is a set of networking protocols that allows networked devices, such as personal computers and home gateways, to discover each other's presence and establish ervices for communications. UPnP is intended primarily for residential networks. Its concept is an extension of plug-and-play, which is a technology for dynamically connecting devices to a computer through network. As Fig. 2c shows, UPnP can provide further assistance with NAT traversal. With UPnP, a device can dynamically join a network, obtain an IP address, convey its capabilities, and learn about the presence and capabilities of other devices automatically. Devices can subsequently communicate with each other directly, thereby further enabling P2P networking. UPnP uses standard TCP/IP and Internet protocols to seamlessly fit into existing networks. One solution for NAT traversal, called the IGD protocol, is implemented via UPnP. Some routers and firewalls expose themselves as IGDs, allowing local UPnP control point to perform a variety of actions, including retrieving the external IP address of the device, enumerate existing port mappings, and add or remove port mappings. By adding a port mapping, a UPnP controller behind the IGD can enable traversal of the IGD from an external address to an internal client.



(a) STUN Operation

(b) TURN Operation

(c) UPnP and IGD interworking.

Fig. 2: NAT traversal Schemes: a) STUN; b) TURN; c) UPnP and IGD interworking.

3. Local UPnP IGD Interworking Scheme (LUIIS)

The underlying idea of LUIIS is providing NAT traversal service to P2P applications on the same host through UPnP. These P2P applications are not aware of other NAT traversal schemes but UPnP. The LUIIS implementation can be viewed as an application running on the host. As Fig. 3 illustrates, when a user initializes LUIIS, LUIIS will firstly detect the presence of IGD on the network. If NAT X is an IGD, then

LUIIS is needless. After IGD detection, LUIIS will perform NAT behaviour test with a STUN server to find the mapping and filtering behaviours of NAT X. Since a dependent-mapping NAT generates different mapped-address for each session, hosts behind such NAT cannot acquire effective mapped-addresses for communication. Even if NAT X is an independent-mapping NAT, dependent-filtering is not acceptable in LUIIS because such dependent-filtering only accepts packets sent from an EH for which a mapped-address has been created previously. As a result, LUIIS can only use the mapped-address when both behaviours of NAT X are independent. For the rest of the cases, LUIIS will acquire a relay address generated by TURN server for NAT traversal.

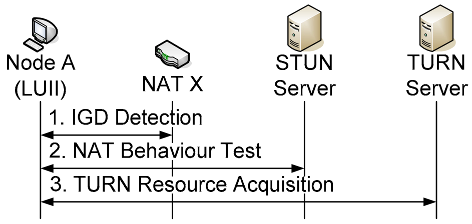


Fig. 3: LUIIS initialization

When a P2P application on Node A wishes to establish a session with Node B, it has to obtain an effective external transport address first. As Fig. 4 shows, this P2P application gets an external address form LUIIS through UPnP. This architecture maintains the backward compatibility to P2P applications. P2P applications on Node A do not need any modification to traverse NAT X. After obtaining a transport address, Node A sends out a session initiation request (INVITE in SIP) to Node B. This request contains a transport address for receiving subsequent data. Then Node B responds Node A with a message (200 OK in SIP) contains its external transport address. In Fig. 4, we assume Node B has already got a transport address.

After receiving the transport address of Node B, the P2P application on Node A will send subsequent data to this transport address directly. On the other hand, the data sent form Node B destined for Node A will be delivered to LUIIS first. This is because Node A obtains its external transport address from LUIIS. Since P2P application and LUIIS use different ports on the same host, after receiving data, LUIIS will divert such data to P2P application. This can be done by adding a new iptable rule in Linux.

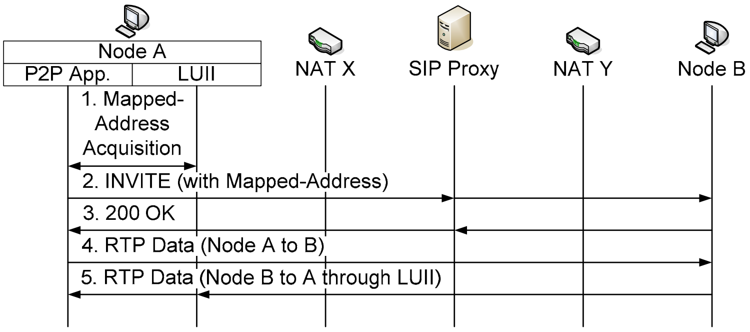

Node A
P2P App.
LUII
NAT X
SIP Proxy
NAT Y
Node B
1. Mapped-Address Acquisition
2. INVITE (with Mapped-Address)
3. 200 OK
4. RTP Data (Node A to B)
5. RTP Data (Node B to A through LUII)

Fig. 4: NAT Traversal using LUIIS (SIP-based VoIP application).

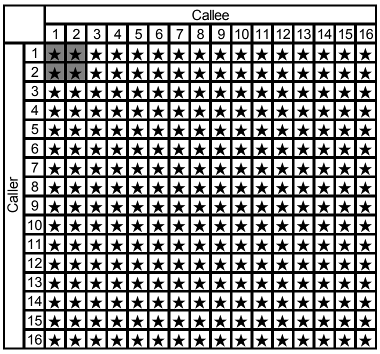
4. Experiment Design and Results

To study the feasibility of the proposed approach, we design an experimental environment for testing. Conducting the repeated experiments, collecting test results, and verifying the performance of each NAT combinations can be done by hand, but this may waste a lot of manpower and time. Therefore, the experiments in this study use a fully-mesh topology to verify the feasibility of LUIIS. As Table 1 illustrates, this topology uses 16 visible domestic NATs available on the market. Table 1 also shows the mapping and filtering behaviours of each NAT. During the experiments, each NAT disables its UPnP service to test if LUIIS can function properly. Node A and Node B are behind the 16 NATs and install LUIIS. Both hosts can switch to the dedicated NATs and thus generates 16*16=256 NAT combinations.

The star sign in Fig. 5 indicates that both hosts can establish a communication path. The shaded area stands for the case both host can establish a direct communication without relaying packets through TURN server while the blank area stands for the case that at least one of the hosts sending packets through TURN server. The test results verified that LUIIS can establish communication path even if its NAT is not an IGD.

Table 1: Required NAT devices in the experiments.

No.	Brand	Model	Mapping/ Filtering	No.	Brand	Model	Mapping/ Filtering
1	D-Link	DI-604	I/I	9	Edimax	BR-6204WG	I/D
2	SMC	SMCWBR14-G2	I/I	10	Lemel	LM-WLG6400	I/D
3	Linksys	WRT150N	I/D	11	Linux	Iptables	I/D
4	Corega	CG-BARMX2	I/D	12	3Com	3CRWER100-75	I/D
5	Planex	BLW-54MR	I/D	13	AboCom	FSM410	I/D
6	SMC	SMCWGBR-14N	I/D	14	Asus	RX3041	D/D
7	Belkin	F5D8231TW4	I/D	15	Netgear	PR614	D/D
8	Draytek	Vigor 2104P	I/D	16	Zyxel	P334	D/D
I: Independednt; D: Dependent							



Caller \ Callee	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
2	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
3	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
4	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
5	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
6	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
7	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
8	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
9	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
10	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
11	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
12	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
13	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
14	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
15	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★
16	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★	★

Fig. 5: Test results of LUIIS in all NAT combinations.

5. Conclusions

The proposed scheme LUIIS can maintain the backward compatibility to P2P applications support UPnP while detecting network environment actively and finding a proper transport address for NAT traversal. A user only needs to install the LUIIS application for extending NAT traversal to UPnP-aware applications. No extra modification is needed for UPnP-aware applications.

LUIIS can help applications to traverse NAT. However, the portion of independent mapping and filtering NATs is relatively small and hence, most sessions need to relay through a TURN server. There should be a complete behaviour test for both NATs. With information about NAT behaviours, LUIIS may choose a proper scheme and transport address for establishing a direct communication.

6. Acknowledgements

This work was supported in part by National Science Council under Grants NSC 100-2221-E-009-084 and NSC 100-2221-E-009-072-MY3, and D-Link Co., Taiwan.

7. References

[1] J. Rosenberg, et. al. SIP: Session Initiation Protocol. IETF RFC 3261, 2002.

[2] P. Francis, K. Egevang. Traditional IP Network Address Translator (Traditional NAT). IETF RFC 3022, 2001.

[3] F. Audet, Ed. and C. Jennings. Network Address Translation (NAT) Behavioral Requirem. IETF RFC 4787, 2007.

[4] J. Rosenberg, J. Weinberger, C. Huitema, R. Mahy. Session Traversal Utilities for NAT (STUN). IETF RFC 5989, 2008.

[5] J. Rosenberg, R. Mahy, P. Matthews, C. Huitema. Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN). IETF RFC 5766, 2010.

[6] UPnP Device Architecture Version 1.0. UPnP Forum, www.upnp.org, 2000.

[7] M. Aurel Constantinescu, V. Croitoru, D. Oana Cernaianu. NAT/Firewall traversal for SIP: issues and solutions. *Int. Symp. on Signals, Circuits and Syst.* 2005, pp. 521-524.