Journal of Knowledge Information Technology and Systems(JKITS), Vol. 9, No. 5, pp. 555~562, October 2014

Journal of Knowledge Information Technology and Systems

ISSN 1975-7700

http://www.kkits.or.kr

Improved STUN Algorithm for Testing Network Address Translation

Tran Thi Thu Ha, Bae-Ho Lee, Jinsul Kim*

School of Electronics& Computer Engineering, Chonnam National University

ABSTRACT

Peer-to-Peer (P2P) network is a group of computers communicating directly with each other without through a central server. P2P network is used for file sharing, multi-player online games, to avoid the expense and delay of traffic at the server. However, P2P communication has problems dealing with Network Address Translation (NAT) traversal. Conventional Simple Traversal of UDP through NATs (STUN) protocol was given to discover the type of NATs behind Clients. In conventional STUN processing, the tests follow the sequence of time that means the first test is needed to get the result before performing the second test and so on. Thus, the system testing has the delay of time. So, in this paper, we propose an improved STUN algorithm that separating tests of the conventional STUN algorithm and perform all tests in parallel. So, the execution time will be shortened significantly. We also have reality experiment in which is applied improved STUN algorithm for testing NATs.

KEYWORDS: Types of NATs, STUN protocol, Peer-to-Peer, improved STUN, NAT traversal.

ARTICLE INFO: Received 1 August 2014, Revised 10 October 2014, Accepted 10 October 2014.

1. Introduction

The Internet is based on 32 bits Internet Protocol (IP) addresses that means the maximum number of computers on the Internet is around four billions. In fact, the Internet may be only a few years away from running out of IP addresses. A technique known as NAT [1], [2], [3] was developed to allow the use of a single IP address

*Corresponding author is with the School of Electronics & Computer Engineering, Chonnam National University, Gwangju, 500-757 Republic of Korea.
E-mail address: jsworld@jnu.ac.kr

for a whole network of computers. STUN protocol was released in 2003 to discover the type of NATs behind Clients. However, conventional STUN has a disadvantage about execution time because all test following sequence. In this paper, we propose an idea to improve STUN algorithm that separating tests of the conventional STUN algorithm and perform all tests in parallel then apply this algorithm for testing NATs behind Peers. Our solutions can be applied to many applications from P2P networks such as instant messaging, streaming media. Structure of remain paper as following: section 2: related work, section 3: proposed idea and system design, section 4: implementation and experiment results, section 5: conclusion.

2. Related Works

2.1 Network Address Translation Terminologies

A NAT is located between the public Internet and the network, works by rewriting IP addresses and port numbers in IP headers from the single public IP address of the NAT device instead of the actual source or destination [4, 5].

There are several ways of implementing network address and port translation such as STUN protocol [6]. It classifies NAT implementation as full-cone NAT, restricted-cone NAT, port-restricted cone NAT or symmetric NAT as shown in <Figure 1>.

Full-cone NAT is one where all requests from the same internal IP address and port are napped to the same external IP address and port.

Restricted cone NAT unlike full-cone NAT, an external host can send a packet to the internal host only if the internal host had previously sent a packet to external IP address.

Port-restricted cone NAT is like restricted cone NAT, but the restriction include ports numbers.

Symmetric NAT is one where all requests from the same internal IP address and port, to a specific destination IP address and port, are mapped to the same external IP address and port.

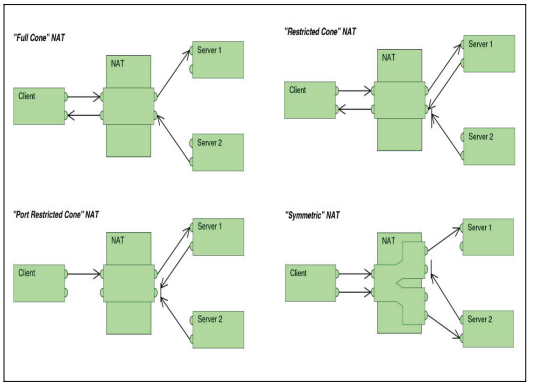

그림 1. NAT의 4가지 유형
Figure 1. Four types of NATs

2.2 Simple Traversal of UDP through NATs Protocol (STUN)

Simple Traversal of User Datagram Protocol through Network address translation is lightweight protocol that allows application to discover the presence and types of NATs and firewalls between them and the public Internet. It also provides the ability for application to determine the public Internet Protocol addresses allocated to

them by the NAT. STUN can be used to discover type of NAT on the path between a client and server, and attempt to discover the type of NAT by a structured sequence of requests and responses [7]. In STUN, an algorithm proposed for testing NAT of a device accordingly as shown in <Figure 2>. The client sends an initial request to the STUN server.

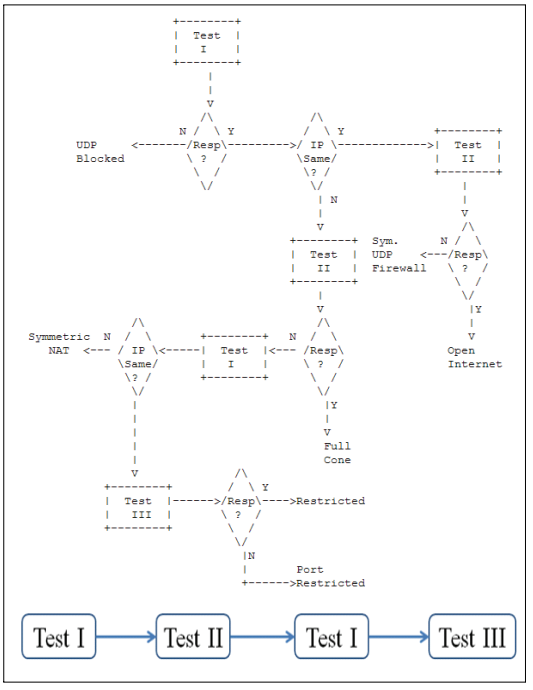

그림 2. 기존 STUN 프로토콜을 사용한 NAT 검색 과정

Figure 2. NAT discovery process using conventional STUN protocol

The flow make use of three tests. In the test I, the client send STUN Binding request to a server. In the test II, the client send a Binding request with both the change IP and change port flag. In the test III, the client send a Binding request with only the change port flag.

If the public address and port in the return response are the same as the local address, then the client can conclude hat there is no NAT in the path between the client and the server [8, 9].

If the values differ, the client can conclude that there is a NAT on the path. STUN then uses subsequent requests to determine the type of NAT. If the STUN client receives this alternative source response, then it can conclude that it is behind a full-cone NAT. If no response is received tho the second request, then the STUN client sends the original probe request. If the returned address and port values relating to the new NAT binding, then the client can conclude that it is behind a symmetric NAT. If the address and port are the same, the client is either behind a restricted or port restricted NAT. To make a determination about which one is it behind, the client initial test III. If response is received, its behind a restricted NAT, if not, its behind a port restricted NAT [10, 11, 12].

3. Proposed Idea and System Design

3.1 Setup Reality Scenario

The scenario included two Peers, two Servers for discovering type of NAT behind each Peer.

Two Peers belongs to different private networks, with Peer A's private/public IP address: 192.168.92.128/168.131.39.188 and Peer B's private/public IP address: 192.168.161.128/ 14.162.247.234.

Two Servers: Server TEST_NAT 1 IP address: 128.199.252.55 and Server TEST_NAT 2 IP address: 173.254.246.181.

3.2 Testing NATs with Improved STUN Algorithm

From conventional STUN algorithm is given in <Figure 2>, we propose idea to improve this algorithm to separate tests of the conventional STUN algorithm and perform all tests in parallel as show in <Figure 3>.

In experiments, we sent packet STUN has structure of header and content shown in <Figure 4> from Clients to Server 1, 2 and converse.

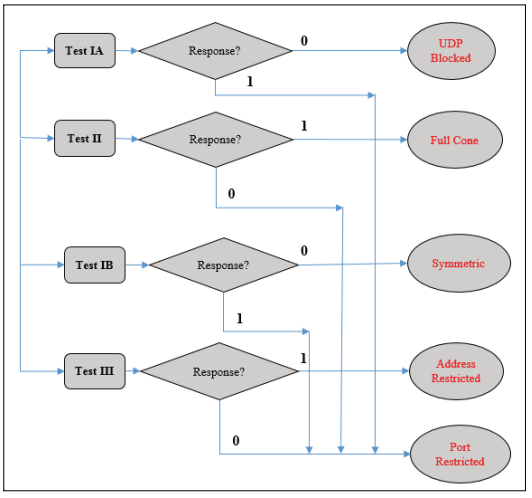

그림 3. 향상된 STUN 프로토콜을 사용한 NAT 검색 과정

Figure 3. NAT discovery process using improved STUN protocol

With detail of STUN packages as below: we have four tests which independence from each other and perform parallel.

Test IA with packet STUNTEST_1A is sent from Client to Server TEST_NAT 1.

Test III with packet STUNTEST_3 is sent from Server TEST_NAT 1 to Client.

Test II with packet STUNTEST_2 is sent from Server TEST_NAT 2 to Client.

Test IB with packet STUNTEST_1B is sent from Client to Server TEST_NAT 2.

These tests don't need following sequence time like conventional STUN algorithm. They can be performed parallel without depend on each other.

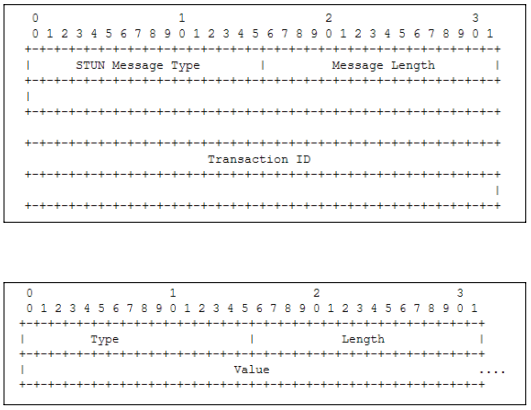

그림 4. 헤더 및 STUN 패키지 내용

Figure 4. Header and Content of STUN package

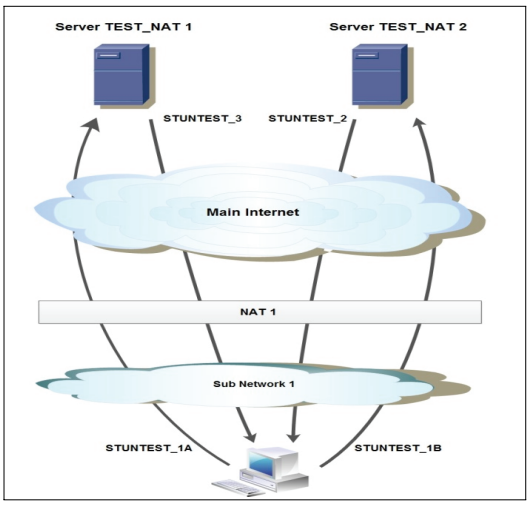

그림 5. NAT 테스트를 위한 동작

Figure 5. Actions for testing NATs

From the results obtained from test IA, test III, test II, test IB following actions as shown in <Figure 5> and based on <Table 1>, we can discover the type of NATs behinds Peers.

표 1. 각각의 테스트 및 NAT결과의 매개변수

Table 1. Parameter of each Test and NAT results

Test IA	Test II	Test IB	Test III	NAT result
0	–	–	–	Block UDP
1	1	–	–	Full-cone NAT
1	0	0	–	Symmetric NAT
1	0	1	1	Restricted NAT
1	0	1	0	Port Restricted NAT

Where: “1”: response, “0”: no response, “-”: don’t care.

4. Implementation and Experiment Results

Development environment: Ubuntu 12.04 LTS, programming tool: Eclipse CDT, programming language C/C++ compiler by GCC/G++. We obtained some experiment results which will be shown as in <Figure 6, 7, 8, 9>.

4.1 Server NAT 1 and Server NAT 2

<Figure 6, 7> show log file of Server NAT 1 and Server NAT 2 with actions receive and send packages STUN with two Peers. As shown in <Figure 6>, Server NAT 1 received STUNTEST_1A from Clients and sent STUNTEST_3 to Clients. Similarly, after Server NAT 2 received Client’s IP from Server NAT 1, it sent STUNTEST_2 to Clients and received STUNTEST_1B from Clients.

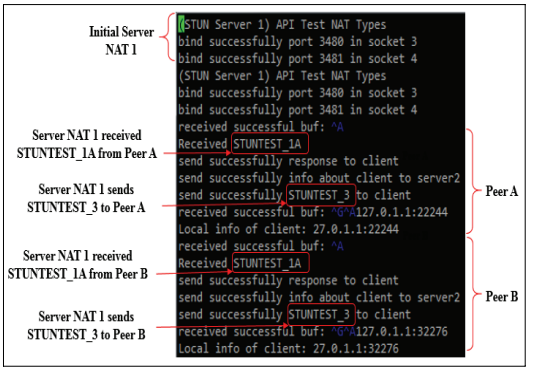

그림 6. 서버 NAT1의 로그파일

Figure 6. Log file of Server NAT 1

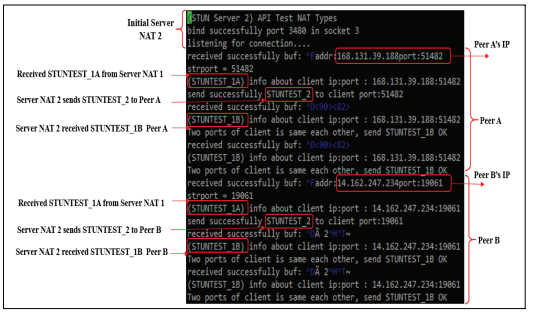

그림 7. 서버 NAT2의 로그파일

Figure 7. Log file of Server NAT 2

4.2 Peer A Side and Peer B Side

<Figure 8, 9> describe the actions of Peer A and Peer B with two Server in testing processing. As seen from <Figure 8>, all tests don’t following sequence for example STUNTEST_3 response earlier than STUNTEST_2. Based on result tests and refer to Table 1, NAT behinds Peer A, Peer B is Full-cone NAT and Symmetric NAT respectively.

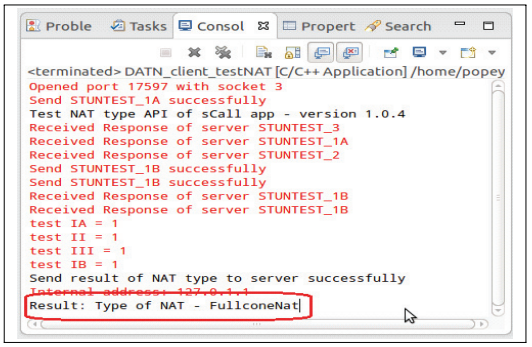

그림 8. 피어 A의 NAT 테스트

Figure 8. Testing NAT of Peer A

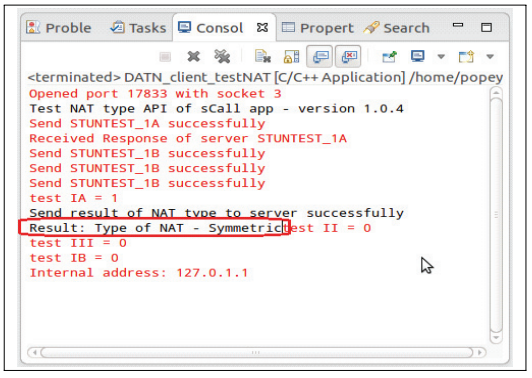

그림 9. 피어 B의 NAT 테스트

Figure 9. Testing NAT of Peer B

4.3 Crossing NAT, establish direct connection between Peer A and Peer B

그림 10. 피어 A의 터미널

Figure 10. Peer A's terminal

그림 11. 피어 B의 터미널

Figure 11. Peer B's Terminal

After knowing kind of NAT behinds each Peer, P2P connection is established between 2 Peers. <Figure 10> shows the Peer A's terminal when this Peer tries to connect with Peer B in the P2P network. Similarly, as shown in <Figure 11> is Peer B's terminal.

5. Conclusion

In this paper, we proposed an idea to improve STUN algorithm with performing all tests in parallel. In experiments, we used improved STUN algorithm for testing NATs behind Peers. From the experimental results, we can recognize many benefits of improved STUN protocol such as: simplicity, high mobility and quickly. Our idea can be applied into P2P networks for social applications, online data sharing.

References

[1] P. Srisuresh, and M. Holdrege, *IP network address translator (NAT) terminology and considerations,* RFC-2663, 1999.

[2] L. D'Acunto, J.A Pouwelse, and H.J. Sips, *A measurement of NAT & firewall characteristics in peer to peer*, In Proceedings of the ASCI Conference, 2009.

[3] K. Egevang, and P. Francis, *The IP network address translator,* RFC-1631, 1994.

[4] P. Srisuresh, B.Ford, and D.Kegel, *State of Peer-to-peer (P2P) communication across network address translators (NATs)*, RFC-5128, 2008.

[5] Bryan Ford, Pyda Srisuresh, and Dan Kegel, *Peer-to-peer communication across network address translators,* Proceedings of the annual conference on USENIX Annual Technical Conference, 2008.

[6] J. Rosenberg, J. Weinberger-Dynamicsoft, C. Huitema-Microsoft, and R. Mahy-Cisco, *STUN-simple traversal of user datagram protocol through network address translators*, RFC-3489, 2003.

[7] Marc-André Poulin, Lucas Rioux Maldague, Alexandre Daigle, and Francois Gagnon, *NAT traversal in peer-to-peer architecture* - Technical Report SCE-12-04 of Carleton University, Canada, 2012.

[8] A. Mueller, A. Klenk, and G. Carle, *Behavior and classification of NAT devices and implications for NAT- traversal*, IEEE Network Special Issue on Implications and Control of Middleboxes in the Internet, 2008.

[9] A. Mueller, A. Klenk, and G. Carle, *ANTS-A framework for nnowledge based NAT traversal, in IEEE globecom next-generation networking and internet symposium*, USA, 2009.

[10] Geoff Huston, *Anatomy: A look inside network address translators*, The Internet protocol Journal, Vol. 7, No. 3, pp. 1~40, 2004.

[11] M. Boucadair, R. Penno, and D. Wing, *Universal plug and play (UPnP) internet gateway device- port control protocol interworking function (IGD-PCP IWF)*, RFC-6970, 2013.

[12] Lisa Sherwin, *UPnP specifications named international standard for device interoperability for IP-based network devices*, 2009.

네트워크 주소 변환 테스트를 위한 향상된 STUN 알고리즘

Tran Thi Thu Ha, 이 배 호, 김 진 술

전남대학교 전자컴퓨터공학부

요 약

P2P 네트워크는 중앙 서버를 통하지 않고, 각각 직접적으로 연결하는 컴퓨터 통신 그룹이다. P2P 네트워크는 파일공유와 멀티 플레이어 온라인 게임에 사용되기도 하고, 서버 통신의 지연과 비용을 막기 위해 사용된다. 하지만 P2P 통신은 NAT 탐색에 있어 문제를 가지고 있다. NATs(STUN) 프로토콜을 통한 UDP의 기존의 탐색은 사용자를 배재한 채 NATs의 유형이 발견된다. 기존의 STUN 과정에서는, 테스트들의 시간 순서를 따르는데, 이순서는 두 번째 테스트와 그 이후의 테스트들 전에 첫 번째 테스트의 결과를 얻어낼 필요가 있으며, 테스트는 시간의 지연을 가지고 있다. 본 논문에서는 기존의 STUN 알고리즘 테스트를 분리시켜 향상된 STUN 알고리즘을 제안하고, 모든 테스트들을 동시에 수행한다. 따라서 처리시간은 현저히 줄어들며, NATs 테스트를 위한 개발된 STUN을 적용하는 실험을 실시한다.

Acknowledgments

This study was financially supported by

Chonnam National University, 2009 and Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(NRF-2013R1A1A2013740).

Tran Thi Thu Ha is currently a M.S. candidate at Smart Mobile and Media Computing Laboratory, School of Electronics and Computer Engineering, Chonnam National University, South Korea. She received B.S. degree from the School of Electronics and Telecommunications, HaNoi University of Science and Technology, VietNam in 2011. She was a solution engineer at Asian Communication Solution, JSC in 2012. Her major interests are in the research areas of Mobile Cloud Computing, Next Generation of Mobile Platform, Mobile Operating System, Peer-to-Peer Network.

E-mail address: thuhabkhn@gmail.com

Bae-Ho Lee received the B.S. degree in Electronics from Hanyang University in 1978, the M.S. degrees in Electrical and Electronics from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea in 1980 and the Ph.D. degrees in Electronics and Electrical Communications Engineering (EECE) from University of Missouri at Columbia, USA in 1993. From 1993 to current, he is a professor in Division of Electronics and Computer Engineering, Chonnam National University, Gwangju, Korea. His research interests include Artificial Intelligence, Multimedia, Digital Image Processing, and Computer Vision.

E-mail address: bhlee@jnu.ac.kr

JinSul Kim received the B.S. Degree in computer science from University of Utah, Salt Lake City, Utah, USA, in 2001, and the M.S. and Ph.D. degrees in digital media engineering, department of information and communications from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2005 and 2008. He worked as a researcher in IPTV Infrastructure Technology Research Laboratory, Broadcasting/ Telecommunications Convergence Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea from 2005 to 2008. He worked as a professor in Korea Nazarene University, Chon-an, Korea from 2009 to 2011. Currently, he is a professor in Chonnam National University, Gwangju, Korea. He has been invited reviewer for IEEE Trans. Multimedia since 2008. He has been invited for TPC(Technical Program Committee), IWITMA2009/2010, and PC(Program Chair), ICCCT2011 His research interests include QoS/QoE, Measurement/ Management, IPTV, Mobile IPTV, Smart TV, Multimedia Communication and Digital Media Arts.

E-mail address: jsworld@jnu.ac.kr