Sensors and Materials, Vol. 34, No. 5 (2022) 1791–1801
MYU Tokyo

S & M 2929

Session Traversal Utilities for Network Address Translator (STUN)-based Traversal Approach Using Port Assignment Prediction Mechanism

Shaw-Hwa Hwang[1] and Cheng-Yu Yeh[2*]

[1]Department of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, 1001, Daxue Rd., East Dist., Hsinchu 300093, Taiwan
[2]Department of Electrical Engineering, National Chin-Yi University of Technology, 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 41170, Taiwan

(Received Nobember 26, 2021; accepted March 9, 2022)

Keywords: network address translator (NAT), NAT traversal, session traversal utilities for NAT (STUN), real-time transport protocol (RTP), session initiation protocol (SIP)

This paper presents a STUN-based approach as a way to improve the NAT traversal success rate. As a preliminary step, 50 commercially available NATs are categorized according to the mapping and filtering rules. The port number assignment mechanism for each type of NAT is then deeply probed, and it is found to exert a strong influence on the success rate. On this basis, the combined use of STUN and the probed mechanism is presented as an effective way to improve the success rate. The high performance of this approach is demonstrated by a success rate of 94.36%, compared with 86.6% using STUN and 91.36% using a multi-hole punching counterpart. This approach is expected to be widely applied to peer-to-peer (P2P) communication apps, such as those used in voice and video streaming over IP (V^2oIP), Internet of Things (IoT), and many more.

1. Introduction

With the rapidly increasing number of internet users and apps, apps of voice and video streaming over IP (V^2oIP)[(1–4)] have gained popularity, including peer-to-peer (P2P) apps.[(5–10)] However, owing to an insufficient number of public IP addresses and a lack of discussion on hacker and virus issues, network address translators (NATs)[(11–14)] are widely employed over the internet.

Resources in public networks can be accessed by computers in private networks using NAT techniques. Nevertheless, this incurs the so-called NAT traversal problems. For instance, a direct computer connection cannot be built between an external computer and an internal one behind a NAT router unless the communication is initiated first by the internal one. Another problem is that communication of P2P apps can be blocked. For example, there is no way that a voice streaming communication can be made directly between two users without P2P techniques, resulting in the low efficiency of VoIP apps.

*Corresponding author: e-mail: cy.yeh@ncut.edu.tw
https://doi.org/10.18494/SAM3750

ISSN 0914-4935
https://myukk.org/

For this reason, many studies[15–24] have addressed the NAT traversal problems. In the earliest studies, several NAT traversal protocols were proposed, such as session traversal utilities for NAT (STUN),[15–17] traversal using relays around NAT (TURN),[18] and interactive connectivity establishment (ICE).[19] The STUN approach was developed to identify the type of NAT and mapped address, i.e., an external public IP address and a port number in a NAT, but demonstrates poor performance. Although the TURN protocol allows two hosts to exchange packets through a relay server, there is a considerable bandwidth cost of the relay server, particularly when dealing with a large number of operations. The ICE protocol employs both STUN and TURN to establish a connection, either a P2P connection or through a relay, between two hosts.

As in Refs. 11 and 12, a NAT is characterized and then categorized to improve its traversal performance. As illustrated in Sect. 2 and explicitly stated in Ref. 11, a NAT is classified according to the mapping and filtering rules. Moreover, multi-hole punching-based techniques have been proposed[22–24] to improve the success rate of NAT traversal, where Internet Control Message Protocol (ICMP) was used together with a low time-to-live value to keep ports open. However, a smooth operation cannot be achieved as long as peers lie behind multiple layers of NAT routers.

Although the classification of NAT types has been presented, there is no information on the correlation between the currently assigned port and the next one with respect to the mapping rules. To overcome this deficiency, in this study, we attempt to probe in detail the port number assignment mechanism for each type of NAT. This analysis is expected to improve the success rate of NAT traversal as well as further reduce the bandwidth cost of the relay server for P2P communications. The results of this study are expected to be widely applied to P2P communication apps, such as those in V^2oIP, Internet of Things (IoT), and many more.

In this study, 50 commercially available NAT routers are collected as the testing objects, and NAT types are identified according to the classification stated in Ref. 11. Furthermore, the successively assigned port number of a NAT is analyzed by practical tests to find the correlation between the rule for port number assignment and the NAT type. This analysis will help to increase the accuracy of port number prediction. Consequently, the combined use of STUN and the port assignment prediction mechanism is presented and experimentally validated to be an effective way to improve the NAT traversal success rate.

This paper is outlined as follows. The classification of NAT types is briefly described in Sect. 2. Section 3 presents the proposed traversal algorithm and port number prediction mechanism. Section 4 gives the experimental results and discussion. Finally, this work is summarized in Sect. 5.

2. Classification of NAT Types

For communication purposes, a port is assigned to a packet waiting for transmission in a NAT. Moreover, relevant information is also recorded for management purposes, including the source IP (SrcIP), source port (SPort), destination IP (DstIP), and destination port (DPort). The recorded information is then compared to decide whether to accept a packet according to the adopted filtering rule.

NATs are classified according to the port assignment and packet filtering rules. As illustrated in Ref. 11, mapping rules refer to the way a port is assigned according to the DstIP and DPort associated with a transmitted packet in a NAT, and can be categorized into endpoint independent (EI), address dependent (AD), and address and port dependent (APD), as shown in Fig. 1. All requests from the same internal IP address and port to any DstIP or DPort are mapped to the same external global IP address and port when the mapping rule of a NAT is of the EI type, as illustrated in Fig. 1(a), where IP_A and P_A represent the internal IP address and port; IP_B/IP_C and P_{B1}/P_{B2}/P_{C1}/P_{C2} are two random DstIPs and four random DPorts, respectively. In Fig. 1(b), all requests from the same internal IP address and port to the same DstIP are mapped to the same external global IP address and port when the mapping rule of a NAT is of the AD type. Moreover, all requests from the same internal IP address and port to a specific DstIP and DPort are mapped to a unique external IP address and port when the mapping rule of a NAT is of the APD type, as presented in Fig. 1(c).

In contrast, filtering rules refer to the way that a packet awaiting admission is filtered according to SrcIP and SPort, and can be classified into EI, AD, and APD as in the mapping rules (Fig. 2). Any external node can send a packet to the internal node by sending a packet to the mapped address, as illustrated in Fig. 2(a). An external node can send a packet to the internal node only if the internal node has previously sent a packet to the external node, such as User B

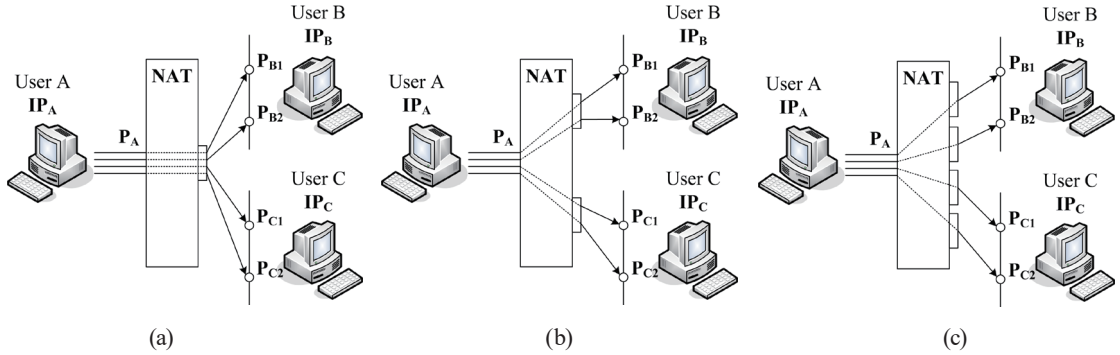

Fig. 1. Mapping rules of NAT: (a) EI, (b) AD, and (c) APD.

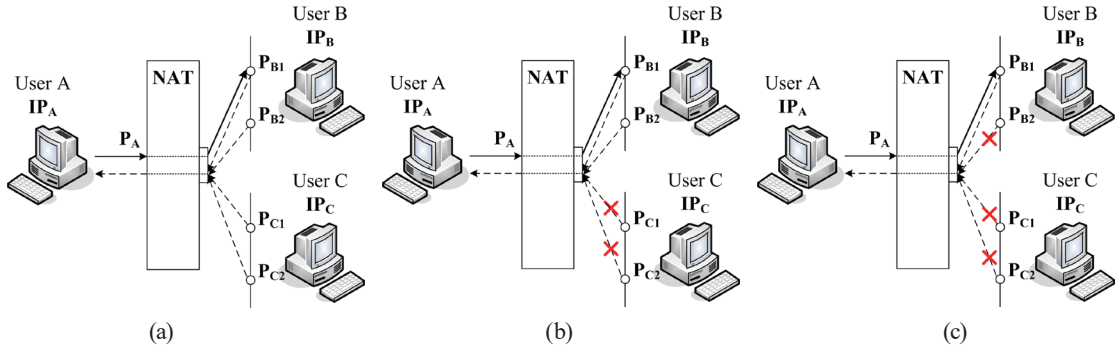

Fig. 2. (Color online) Filtering rules of NAT: (a) EI, (b) AD, and (c) APD.

with address IP_B in Fig. 2(b). In addition, a specific port of an external node can send a packet to the internal node only if the internal node has previously sent a packet to the specific port of the external node, such as User B with address IP_B and port P_{B1} in Fig. 2(c).

Consequently, Table 1 gives nine types of NATs corresponding to different combinations of the mapping and filtering rules. Full, restricted, port restricted, and symmetric cones, as classified in Ref. 15, are essentially Types 1, 2, 3, and 9, respectively, in Table 1.

3. Proposed NAT Traversal Approach

In this paper, the combined use of STUN and the port assignment prediction mechanism is presented as an efficient way to improve the NAT traversal success rate. As the first step, 50 commercially available NATs are categorized according to the mapping and filtering rules. The port number assignment mechanism for each type of NAT is then deeply probed. On the basis of this analysis, a STUN-based NAT traversal approach to increase the traversal success rate is proposed.

3.1 Analysis on port assignment of NATs

Although there are three mapping rules to assign a port in a NAT, there is no information on the correlation between the currently assigned port and the next one. To solve this problem, this issue is deeply probed. As the first step, EI, AD, and APD mappings are respectively represented as

$$MT_{EI}(NPort)\text{: } \{\ SrcIP, SPort\ \}, \tag{1}$$

$$MT_{AD}(NPort)\text{: } \{\ SrcIP, SPort, DstIP\ \}, \tag{2}$$

$$MT_{APD}(NPort)\text{: } \{\ SrcIP, SPort, DstIP, DPort\ \} \tag{3}$$

to express the correlation between an assigned port and the relevant packet information, and a mapping table is prebuilt for management purposes. For example, in the EI mapping, *SrcIP* and *SPort*, rather than *DstIP* and *DPort*, are referenced by a NAT for the port assignment, designated as *NPort*, as in Eq. (1). In other words, *NPort* is dependent on *SrcIP* and *SPort*. Likewise, in the APD mapping, *SrcIP*, *SPort*, *DstIP*, and *DPort* are all referenced for the port assignment, as expressed in Eq. (3), that is, a change in either *SrcIP*, *SPort*, *DstIP*, or *DPort* is reflected by a change in *NPort*.

Table 1
NAT type classification.

Mapping/Filtering	EI	AD	APD
EI	Type 1	Type 2	Type 3
AD	Type 4	Type 5	Type 6
APD	Type 7	Type 8	Type 9

With *SrcIP* and *SPort* remaining invariant, we define the quantity

$$\Delta(n) = NPort(n) - NPort(n-1), \tag{4}$$

where n represents the current time. Taking the EI mapping as an example, $\Delta(n) = 0$ for all n, that is, *NPort* is kept constant during the port assignment.

In this paper, a total of 50 commercially available NATs are collected as the testing objects. Each product brand and model number thereof are given in Table 2. Subsequently, Table 3 gives the respective NAT types and "port steps", i.e., $\Delta(n)$ defined in Eq. (4), obtained by our practical testing. Note that the proxy server must bind multiple public IP addresses to probe the port number assignment mechanism of NATs.

Table 3 reveals that 44 out of the 50 port steps are zero, meaning that EI mapping is employed in the vast majority of the collected NATs, whereas the random port step in NATs 2, 6, and 7, all belonging to Type 9, indicates that there is no mapping rule in these cases. Note that a NAT traversal benefits from the condition $\Delta(n) =$ constant. On the basis of Table 3, an improved NAT traversal approach is presented in the next section.

Table 2
Brand and model number of 50 commercial NAT devices as the testing objects.

Item	Brand (model number)	Item	Brand (model number)
1	3COM (3CRWER100-75)	26	GigaByte (GN-BR02G)
2	AboCom (FSM410)	27	IO DATA (ETG-R)
3	AboCom (WB02N)	28	IO DATA (ETX-R)
4	ASUS (RT-N12E)	29	IO DATA (NP-BBRM)
5	ASUS (RT-N12)	30	Lemel (LM-IS6400B)
6	ASUS (Rx3041)	31	LevelOne (FBR-1418TX)
7	ASUS (Rx3081)	32	LevelOne (WBR-3405TX)
8	BELKIN (F5D8235-4 v2)	33	LINKSYS (BEFSR41W)
9	BELKIN (F5D8235-4 v3)	34	LINKSYS (E2000)
10	BELKIN (F7D1301 v3)	35	LINKSYS (WRT150N)
11	BUFFALO (WZR-HP-G300NH)	36	LINKSYS (WRT160NL)
12	BUFFALO (WZR-HP-G300NH2)	37	NETGEAR (WGR614)
13	Corega (CG-BARMX2)	38	NETGEAR (WNDR3400)
14	D-Link (DI-604)	39	NETGEAR (WNDR3700v2)
15	D-Link (DI-707P)	40	SAPIDO (RB-1802)
16	D-Link (DI-LB604)	41	SAPIDO (RB-3001)
17	D-Link (DIR-101)	42	SMC (SMCBR14)
18	D-Link (DIR-320)	43	SMC (SMCWBRS14-N2-TW)
19	D-Link (DIR-615)	44	TP-LINK (TL-R402M)
20	D-Link (DIR-655)	45	TP-LINK (TL-R460)
21	DrayTek (Vigor2110)	46	TP-LINK (TL-WR841N)
22	DrayTek (Vigor2200V/VG)	47	TOTO-LINK (N150RT)
23	EDIMAX (BR-6204Wg)	48	Zonet (ZSR0104B)
24	EDIMAX (BR-6228nS)	49	ZyXEL (Prestige 334)
25	EDIMAX (BR-6314K)	50	ZyXEL (NBG-4115)

Table 3
Behavior property for each NAT device.

Item	NAT type	Port step	Item	NAT type	Port step
1	3	0	26	1	0
2	9	Random	27	2	0
3	3	0	28	1	0
4	3	0	29	2	0
5	3	0	30	3	0
6	9	Random	31	1	0
7	9	Random	32	1	0
8	1	0	33	5	1
9	1	0	34	3	0
10	2	0	35	3	0
11	3	0	36	3	0
12	3	0	37	2	0
13	2	0	38	2	0
14	1	0	39	2	0
15	1	0	40	3	0
16	3	0	41	3	0
17	3	0	42	1	0
18	2	0	43	7	1
19	3	0	44	1	0
20	1	0	45	1	0
21	3	0	46	3	0
22	3	0	47	3	0
23	1	0	48	2	0
24	3	0	49	9	1
25	3	0	50	2	0

3.2 Proposed approach

The combined use of STUN and a port number prediction mechanism is applied to NAT traversal. Using the results listed in Table 3 and the predicted port step ahead of a P2P connection, the NAT traversal success rate is improved. Our proposed approach is illustrated in Fig. 3, where IP2–IP4 denote public IP addresses, whereas IP1 and IP5 stand for the private ones in this proposal, that is, Clients 1 and 2 are located under NATs 1 and 2, respectively.

As illustrated in Fig. 3, steps 1, 2, 5, and 6 are part of the STUN protocol. Through these steps, NAT types on both sides are identified and respective port numbers are predicted. For illustration purposes, it is assumed that the current port number P2, assigned by NAT 1, is discovered in steps 1 and 2, and the current port number P4, assigned by NAT 2, is found in steps 5 and 6. In contrast, steps 3, 4, and 7–12 are part of the session initiation protocol (SIP). In simple terms, NAT types at both ends are first identified, then transmitted to the other using SIP signaling via steps 1–12. In this manner, the port number assigned by a NAT can be predicted by the other for NAT traversal purposes. Note that the prediction results in steps 1, 2, 5, and 6 are not affected by those in steps 3, 4, and 7–12 because the connection for SIP signaling is prebuilt for an earlier SIP registration process. In other words, steps 3, 4, and 7–12 are uncorrelated with the port number prediction mechanism in steps 1, 2, 5, and 6 and also with the subsequent NAT traversal.

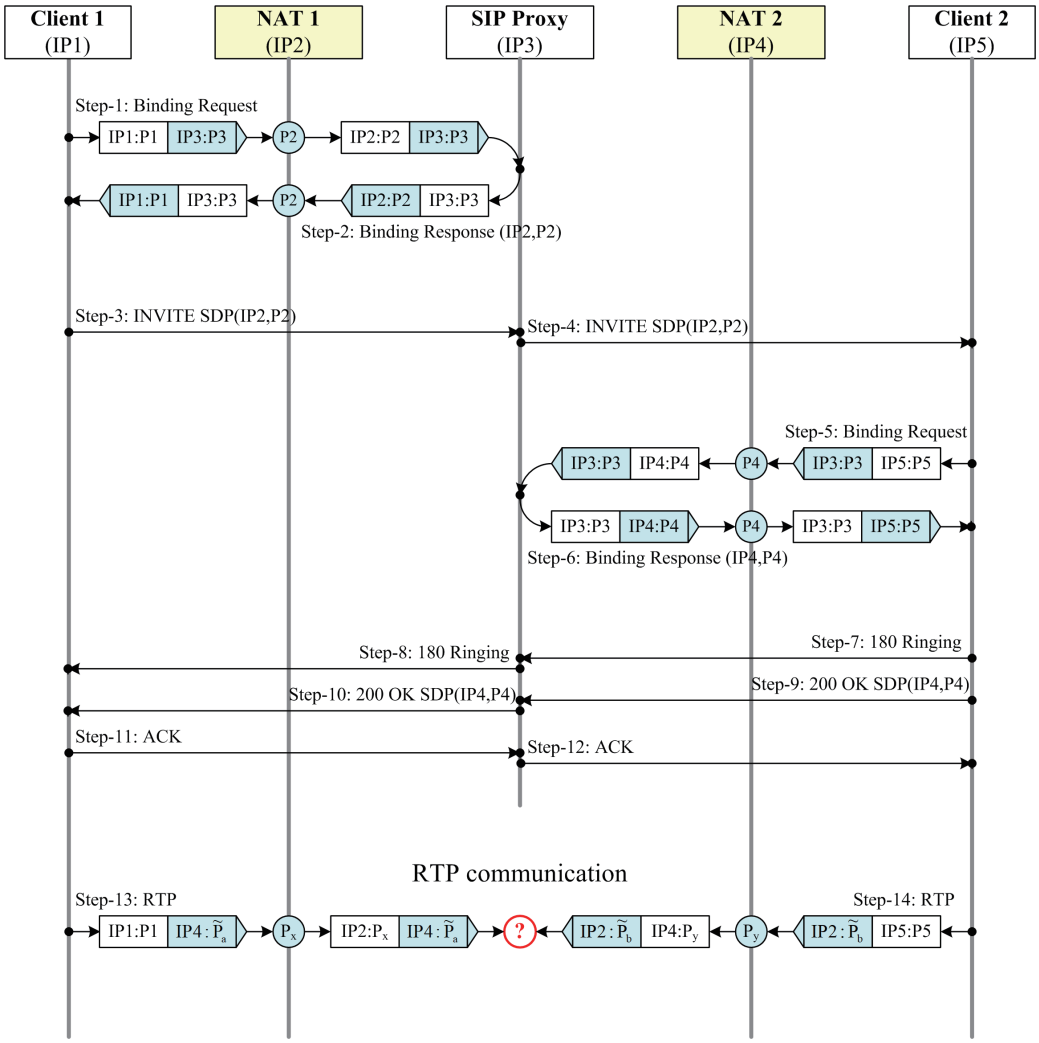

Fig. 3. (Color online) Schematic diagram of the proposed approach for NAT traversal.

In steps 13 and 14, a packet is sent by a NAT to the other using real-time transport protocol (RTP) for NAT traversal purposes. A complete traversal process is detailed as follows. In step 13, Client 1 sends a packet to NAT 2, where (destination IP, destination port) of the packet is represented as (IP4, $\tilde{P}_a$). As the packet passes through NAT 1, a port (P_x in Fig. 3) is assigned, and (source IP, source port) of the packet is simultaneously updated as (IP2, P_x). In the same manner, Client 2 sends a packet to NAT 1 in step 14, that is, (destination IP, destination port) of the packet is denoted by (IP2, $\tilde{P}_b$). As the packet passes through NAT 2, port P_y is assigned and (source IP, source port) of the packet is simultaneously updated as (IP4, P_y). Both $\tilde{P}_a$ and $\tilde{P}_b$ in steps 13 and 14 are the predicted port numbers assigned by both NATs. A NAT traversal is built once the conditions $\tilde{P}_a = P_y$ and $\tilde{P}_b = P_x$ are satisfied, meaning that a P2P connection is made between both clients accordingly.

For illustration purposes, suppose that NAT 1 is identified as Type 3 via steps 1–12, and Eq. (4) gives $\Delta(n) = 0$ for all n, whereas NAT 2 is identified as Type 5 and $\Delta(n) = 1$. Subsequently,

Client 1 sends a packet with destination port $\tilde{P}_a$ = P4 + 1, and port P_x assigned by NAT 1 remains P2, i.e., P_x = P2. On the other hand, Client 2 sends a packet with destination port $\tilde{P}_a$ = P2 at the same time. In this case, port P_y = P4 + 1 is assigned by NAT 2 because there is a difference between the destination IP addresses of the current packet and the previous one. As a consequence, (source IP, source port, destination IP, destination port) assigned by NATs 1 and 2 are (IP2, P2, IP4, P4 + 1) and (IP4, P4 + 1, IP2, P2), respectively, meaning that an assigned port is accurately predicted by the other. In other words, a P2P connection is established accordingly.

Alternatively, there is no rule to follow for port assignment provided that both NATs are identified as type 9 and the port step is specified as random in Table 3. In this context, an assigned port number cannot be accurately predicted, meaning that a packet is blocked by the other NAT, that is, there is a failure to build a P2P connection.

4. Experimental Results

In this section, the NAT traversal success rate is compared among the proposed, STUN, and multi-hole punching[24] approaches. The 50 NATs listed in Table 2 are categorized in Table 4 according to the mapping and filtering rules employed, and graphically shown in Fig. 4.

Table 4 reveals the existence of one Type 5 NAT and one Type 7 NAT, but the vast majority of NATs belong to Types 1–3 with none belonging to Types 4, 6, and 8, meaning that a minority of commercially available NATs belong to Types 4, 6, and 8. As illustrated in Fig. 4, EI mapping is employed by 44 NATs, accounting for 88% of the total, due its easy implementation. On the other hand, APD and APD+AD filtering are adopted by 25 and 36 NATs, accounting for 50 and 72% of the total, respectively, due to their high security owing to their blocking of irrelevant packets.

Fifty NATs are deployed at the caller and callee ends, that is, there are 50 × 50 = 2500 testing cases. As can be seen in Table 5, the Type 9 NAT is further classified into Types 9 and 9R. The former denotes that a certain rule is followed for port number assignment, i.e., item 49 in Table 3, whereas the latter denotes that port numbers are assigned randomly, i.e., items 2, 6, and 7 in Table 3. Table 5 gives the number of testing cases for the combinations of caller and callee types. For instance, the (1, 1) entry is 13 × 13 = 169 since there are 13 Type 1 NATs deployed at the caller and callee ends. In the same manner, the (1, 7) entry is 13 × 3 = 39 because there are 13 Type 1 NATs and 3 Type 9R NATs deployed at the caller and callee ends, respectively. Note that Table 5 omits Types 4, 6, and 8 since they were not involved in the tests.

Tables 6–8 respectively give the P2P connection test results for the combinations of callers and callees in Table 5 using the STUN, multi-hole punching, and proposed approaches, where

Table 4
Statistics for items in Table 3 according to the mapping and filtering rules.

Number of devices		Filtering		
		EI	AD	APD
Mapping	EI	13	10	21
	AD	0	1	0
	APD	1	0	4

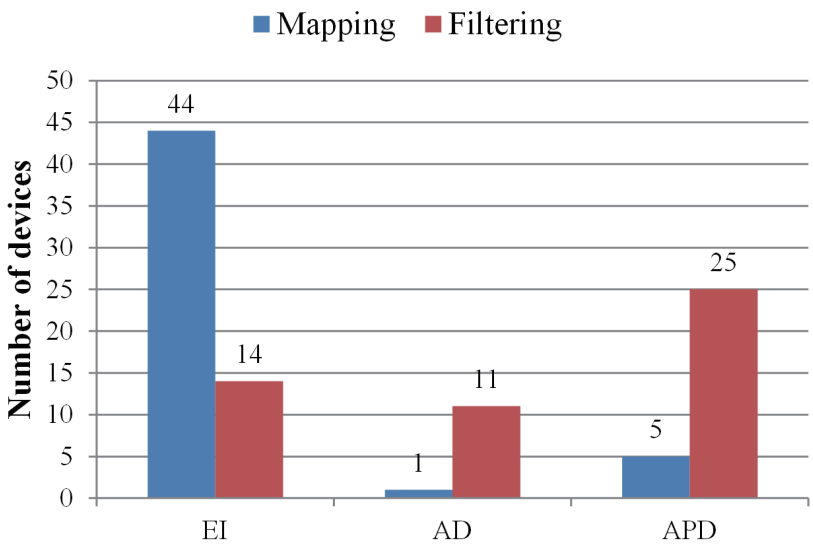

Fig. 4. (Color online) Graphical representation of Table 4.

Table 5
Number of testing cases for various combinations of caller and callee types.

NAT caller type	NAT callee type						
	1	2	3	5	7	9	9R
1	169	130	273	13	13	13	39
2	130	100	210	10	10	10	30
3	273	210	441	21	21	21	63
5	13	10	21	1	1	1	3
7	13	10	21	1	1	1	3
9	13	10	21	1	1	1	3
9R	39	30	63	3	3	3	9

Table 6
P2P connection test results for the cases in Table 5 using STUN.

NAT caller type	NAT callee type						
	1	2	3	5	7	9	9R
1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1
3	1	1	1	0	0	0	0
5	1	0	0	0	1	0	0
7	1	1	0	1	1	0	0
9	1	0	0	0	0	0	0
9R	1	0	0	0	0	0	0

"1" and "0" represent successful and failed connections, respectively. As shown in Table 8, a P2P connection can be successfully made by the proposed approach except for the (3, 9R), (9, 9R), (9R, 3), (9R, 9), and (9R, 9R) combinations. The results in Tables 5–8 are illustrated as success rates in Fig. 5 for performance comparison. The proposed approach has a success rate of 94.36%, outperforming the STUN and the multi-hole approaches (86.6 and 91.36%, respectively).

Table 7
Counterpart of Table 6 for multi-hole punching technique.

NAT caller type	NAT callee type						
	1	2	3	5	7	9	9R
1	1	1	1	1	1	1	0
2	1	1	1	1	1	1	0
3	1	1	1	1	1	1	0
5	1	1	1	1	1	1	0
7	1	1	1	1	1	1	0
9	1	1	1	1	1	1	0
9R	1	1	0	1	1	0	0

Table 8
Counterpart of Table 6 using the proposed approach.

NAT caller type	NAT callee type						
	1	2	3	5	7	9	9R
1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1
3	1	1	1	1	1	1	0
5	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1
9	1	1	1	1	1	1	0
9R	1	1	0	1	1	0	0

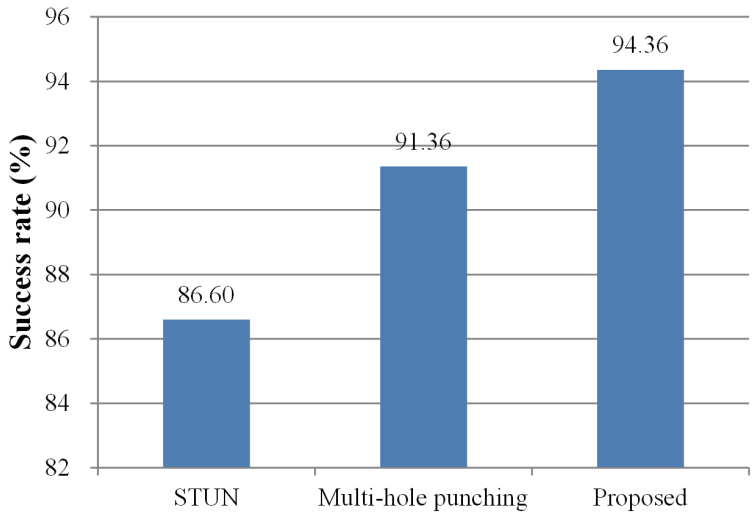

Fig. 5. (Color online) Comparison of success rate among various traversal approaches.

5. Conclusions

There are two main contributions in this paper. First, the port number assignment mechanism for each type of NAT has been probed. Second, an approach combining the use of STUN and the port assignment prediction mechanism is presented as an effective way to improve the NAT

traversal success rate, that is, the bandwidth cost of a relay server for P2P communication can be further reduced. As a preliminary step of this work, 50 commercially available NATs were categorized according to the mapping and filtering rules employed in port number assignment. With the 50 NATs as testing objects, the NAT traversal success rate was measured for various combinations of caller and callee types. The performance was compared among the STUN, multi-hole punching, and proposed approaches in terms of the success rate. The proposed approach had a success rate of 94.36%, outperforming the counterparts, and is expected to be widely applied to P2P communication apps, such as those in V^2oIP, IoT, and many more. NAT traversal remains a key issue for P2P communication in the future.

In future work, we will attempt to find different ways to fit NATs such as NATs 2, 6, and 7 in Table 3 with a complicated port assignment mechanism to further improve the NAT traversal success rate of the proposed approach.

Acknowledgments

This research was financially supported by the Ministry of Economic Affairs, Taiwan, under grant number 110-EC-17-A-02-S5-008.

References

1 T. J. Jung and K. D. Seo: J. Real-Time Image Process. **12** (2016) 455.
2 X. Che, B. Ip, and L. Lin: IEEE Multimedia **22** (2015) 56.
3 H. M. Dermanilian, F. Saab, I. H. Elhajj, A. Kayssi, and A. Chehab: Int. J. Netw. Secur. **17** (2015) 7.
4 T. Anouari and A. Haqiq: J. Mob. Multimedia **9** (2014) 230.
5 T. Rohmer, A. Nakib, and A. Nafaa: IEEE Network **29** (2015) 4.
6 I. Woungang, F. H. Tseng, Y. H. Lin, L. D. Chou, H. C. Chao, and M. S. Obaidat: IEEE Syst. J. **9** (2015) 743.
7 G. Wang, C. Zhang, X. Qiu, and Z. Zeng: J. Internet Technol. **16** (2015) 61.
8 T. Qin, L. Wang, Z. Liu, and X. Guan: Knowl.-Based Syst. **82** (2015) 152.
9 S. Traverso, C. Kiraly, E. Leonardi, and M. Mellia: Comput. Networks **69** (2014) 101.
10 C. C. Huang-Fu, Y. B. Lin, and H. Rao: IEEE Wirel. Commun. **16** (2009) 30.
11 Y. D. Lin, C. C. Tseng, C. Y. Ho, and Y. H. Wu: IEEE Commun. Mag. **48** (2010) 58.
12 A. Müller, G. Carle, and A. Klenk: IEEE Network **22** (2008) 14.
13 C. Park, K. Jeong, S. Kim, and Y. Lee: IEEE Network **22** (2008) 48.
14 S. P. Shieh, F. S. Ho, Y. L. Hung, and J. N. Luo: IEEE Internet Comput. **4** (2000) 42.
15 J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy: IETF RFC 3489 (2003).
16 J. Rosenberg, R. Mahy, P. Matthews, and D. Wing: IETF RFC 5389 (2008).
17 D. MacDonald and B. Lowekamp: IETF RFC 5780 (2010).
18 R. Mahy, P. Matthews, and J. Rosenberg: IETF RFC 5766 (2010).
19 J. Rosenberg: IETF RFC 5245 (2010).
20 C. Y. Ho, F. Y. Wang, C. C. Tseng, and Y. D. Lin: IEEE Commun. Lett. **15** (2011) 4.
21 C. Y. Ho, C. C. Tseng, F. Y. Wang, J. T. Wang, and Y. D. Lin: IEEE Commun. Lett. **15** (2011) 94.
22 K. H. Choi, K. S. Kong, K. S. Chung, D. S. Park, and J. M. Gil: Lect. Notes Electr. Eng. **308** (2014) 147.
23 H. Tran Thi Thu, J. Park, Y. Won, and J. Kim: Proc. IEEE Int. Conf. IT Convergence and Security (IEEE, 2014) 1. https://doi.org/10.1109/ICITCS.2014.7021753
24 Y. Wei, D. Yamada, S. Yoshida, and S. Goto: Proc. APAN Network Research Workshop (2008) 1. http://www.goto.info.waseda.ac.jp