Nonlinear Dyn (2014) 76:1–12
DOI 10.1007/s11071-013-0811-1

ORIGINAL PAPER

WiFi assisted NAT traversal scheme for surveillance patrol robot

Chien-Chao Tseng · Chia-Liang Lin · Bih-Yaw Shih · Chia-Yen Chen · Chen-Yuan Chen

Received: 19 January 2013 / Accepted: 30 January 2013 / Published online: 6 March 2013

Abstract With the advances in wireless communication technology and artificial intelligence, robots are gradually being introduced as part of our life. Previous research proposed a SIP-enabled Surveillance Patrol Robot (SSPR), which tracks a moving object actively and informs the householder of such security alarm. However, the underlying signaling protocol for communication and data streams suffer from the network address translation (NAT) traversal problem as most peer-to-peer (P2P) applications. NAT is a commonly adopted technique to share one public IPv4 address among several hosts located behind a NAT device for alleviating the exhaustion of IPv4 address. NAT devices typically block session requests originating from outside, prevent the establishment of peer-to-peer (P2P) sessions and cause NAT traversal problem. This study proposes WANTS, a WiFi Assisted NAT Traversal Scheme for SSPR. When SSPR is activated, it retrieves the topology information from a server to choose the candidate access point (AP) for handoff. Then SSPR uses the collected network context information to assist its NAT traversal procedure after handoff. Experimental results confirm that WANTS reduces not only connectivity check delay but also protocol messages as compared to the Interactivity Connectivity Establishment (ICE), which is the most acknowledged approach to creating a session across NATs.

Keywords Artificial intelligence · Intelligent robot · Object tracking · Session initiation protocol

C.-C. Tseng · C.-L. Lin
Department of Computer Science, National Chiao Tung University, No. 1001 University Rd., Hsinchu, Taiwan 300, ROC

B.-Y. Shih · C.-Y. Chen (✉)
Department & Graduate School of Computer Science, National Pingtung University of Education, No. 4-18, Ming Shen Rd., Pingtung 90003, Taiwan, ROC
e-mail: cyc@mail.npue.edu.tw

C.-Y. Chen
Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung University Rd., 811, Kaohsiung, Nanzih District, Taiwan, ROC

1 Introduction

The advent of robots has gained immense popularity in recent years and is becoming an intrinsic part of everyday life. Nonlinear systems [1–6] including robot manipulation [7–11] and machinery control [12–16] have been successfully applied to engineering [17–19]. Integrated with communication technology, nonlinear systems further provide a promising solution to construct a smart home environment [20]. As a robot equipped with sensors can be programmed to track a moving object actively, its mobility is a promising solution to organizing a smart and secure home. The previous research [21] proposed a SIP-enabled Surveillance Patrol Robot (SSPR), which is equipped with a

set of sensors and a camera, and is aware of session initiation protocol (SIP). SSPR is designed to patrol the home space periodically. Once SSPR senses a moving object, it starts tracking and initiates a SIP call to the default mobile device through wireless technology such as WiFi. The householder can get the status of house through his/her mobile device in time. When following the object from one room to another, the wireless link may break and result in the disconnection of audio and video streams. With mobility support in SIP, SSPR could perform a handoff from the original access point (AP) to another with better signal strength.

However, when SSPR moves from one AP coverage to another, it may take a few seconds to establish a new session, causing the loss of some important information during these seconds. A major portion of latency can be attributed to the traversal of network address translation (NAT) [22–24]. NAT, which is a solution to alleviate the exhaustion of IPv4 address, modifies network address information stored in packet header when packets pass through a routing device and 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 [24–26], which is a barrier to P2P applications such as the audio and video streams between SSPR and the mobile device of householder. NAT devices block session requests originated from outside and prevents the establishment of P2P sessions. An external host (EH) outside a NAT cannot send a packet to an internal host (IH) directly until an IH sends a packet to EH first. When both hosts are behind different NATs, this problem becomes worse. Many NAT traversal techniques [27–35] have been proposed to deal with NAT traversal problem. Therefore, NAT traversal is indispensable for P2P applications running in NAT environment.

Most of NAT traversal methods require a server with publicly routable IP addresses and could further divided into three major types. One kind of methods only needs the server when establishing a session (such as STUN [28, 35, 36]). Some methods demand the server involved in each packet exchange (such as TURN [29, 35, 36]), which increase both bandwidth costs and latency, and are detrimental to real-time voice and video communication. The rest of the methods do not need a server (such as UPnP [31, 33]), but these approaches require modifications or upgrades on NAT devices. The Internet Engineering Task Force (IETF) proposed Interactive Connectivity Establishment (ICE) [30, 35], which makes use of STUN and TURN, to provide NAT traversal capabilities for session-oriented protocol. ICE hosts exchange accessibility information and negotiate with each other to find potential communication paths between them. However, ICE performs a complete and systematical connectivity check before selecting a communication path, and results a long connectivity check delay and requires considerable message exchanges.

Although many NAT traversal methods are available for SSPR, no single one works well under all circumstances. Furthermore, most existing methods are not aware of topology information of APs and network context information of NATs, resulting unnecessary protocol messages and delays. To improve the mobility and applicability of SSPR, this study proposes a WiFi Assisted NAT Traversal Scheme (WANTS). WANTS integrates wireless technology, topology of APs and network context information of NATs to establish an appropriate communicating path between SSPR and the mobile device of householder when tracking a moving object. When SSPR is activated, it retrieves the geographical information of APs and the network context information of NATs from a server. Then SSPR uses the signal strength of each AP to estimate its location, selects the candidate AP for handoff, and reduces unnecessary message overhead and latency with network context information after handoff. In the case that network context information of existing NAT is not available, SSPR starts collecting such network context information and then uploads to the server for subsequent accesses. As direct communication paths save bandwidth demand and latency caused by relaying methods, direct communication ratio (DCR) serves as another important metric for evaluating a NAT traversal method. Given a set of NAT combinations, DCR is the ratio of combinations for which direct communication paths can be created to all combinations. A traversal method that leads to high DCR naturally demands low relay resource. Experiment results confirm that WANTS eliminates unnecessary protocol messages and shortens the delay of establishing sessions while maintaining the same DCR as compared to the widely adopted method ICE.

The remainder of this paper is organized as follows: we first describe the conventional NAT traversal methods and then introduce the distance calculation and location estimation methods. In the following sections,

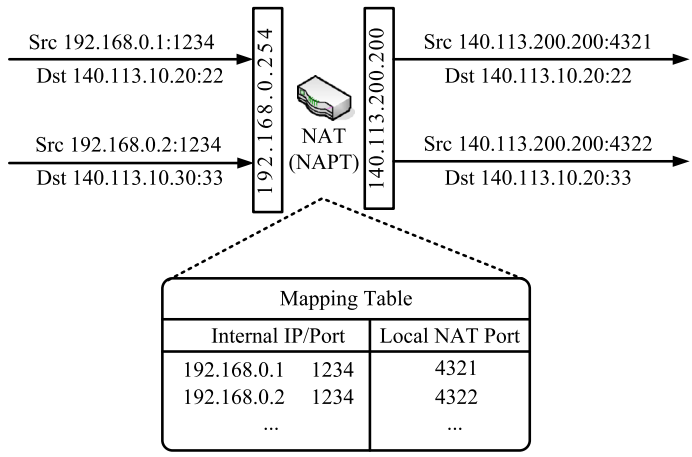

Fig. 1 An example of NAPT operation

we will describe the design of WANTS and present the simulation results. Finally, we summarize our findings in the final section.

2 Related work

2.1 NAT behaviors and traversal methods

NAT allows IHs within a private network to connect to EHs in a public network [22, 26, 38] and ameliorates IPv4 address depletion by allowing globally routable IP addresses to be either reused or shared among several hosts. The most commonly adopted NAT implementation is Network Address Port Translation (NAPT), which allows many hosts to share a single IP address through multiplexing sessions differentiated by a TCP/UDP port number. The hosts with private addresses 192.168.0.1 and 192.168.0.2 in Fig. 1 send packets from port number 1234. A NAPT device translates source addresses of these packets into a single public IP address 140.113.200.200 with different source ports 4321 and 4322. When the NAPT device receives the response traffic, it routes packets destined for port 4321 to 192.168.0.1:1234, while packets to port 4322 would go to 192.168.0.2:1234. 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 session attempt from an IN.

Different NATs may differ in the mapping and filtering behaviors. These behaviors are discussed in detail in the following.

Mapping behavior refers to how mapped-addresses are allocated to sessions (i.e., chooses an external address and a port for each session) [40]. Assume sessions are originated from the same host but are destined for different hosts and/or different ports. Independent mapping NAT uses the same address and port for all outbound packets regardless of destination address and port (Fig. 2a). Address dependent mapping captures the difference among sessions that are destined for different EHs. NAT uses the same mapped-address for packets to the same destination address (Fig. 2b). Finally, address-and-port dependent map-

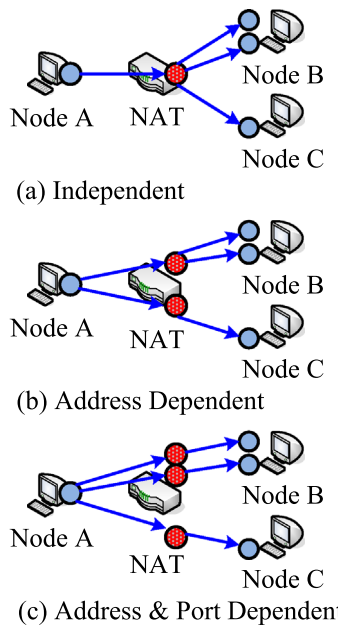

Fig. 2 NAT mapping behavior

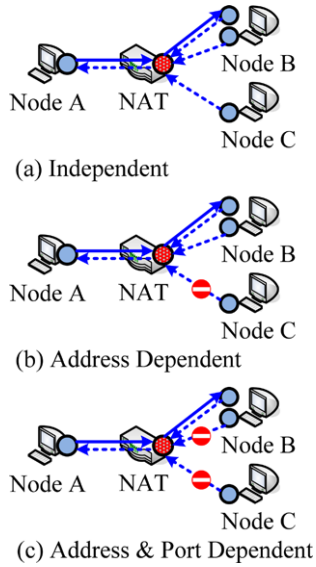

Fig. 3 NAT filtering behavior

ping NAT generates one unique mapped-address for each session (Fig. 2c).

Filtering behavior determines whether a packet sent form the external side is allowed to traverse the NAT based on the source address and port number [40]. Independent filtering allows any EH regardless of its address and port number sends packets to IHs with valid mapped-addresses (Fig. 3a). Address dependent filtering only accepts packets sent from an EH for which a mapped-address (Fig. 3b) has been created previously. NAT with address-and-port dependent filtering only allows packets with source address B and port number b if a mapped-address has already been created for that address (B) and port (b) (Fig. 3c).

NAT type [39] classifies port translation method of NAT into cone and symmetric. As long as packets originating from the same transport address of an IH, a cone NAT assigns the same mapped-address. Cone NATs could further divide into full-cone, address-restricted cone and port-restricted cone with respect to their filtering behaviors. A full cone (FC) NAT has an independent filtering behavior, an address-restricted cone (AR) NAT has address dependent filtering behavior and a port-restricted cone (PR) NAT is similar to AR NAT, except that the restriction includes port numbers. A symmetric (SY) NAT has the strictest mapping and filtering behaviors (i.e., address-and-port dependent mapping and filtering behavior).

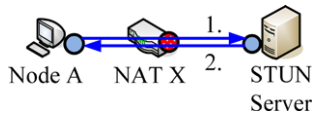

Fig. 4 STUN operation

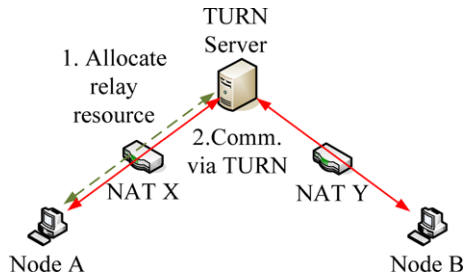

Fig. 5 TURN operation

Hole-punching is the key point of NAT traversal [39]. If an IH wants to receive packets from EHs, the IH should first send an outbound packet to create an appropriate mapped-address in the NAT. A NAT verifies the validity of the destination address of every incoming packet and ensures the existence of a corresponding entry in the translation table that maps to an IH to which the packet belongs. Although many techniques are proposed for the NAT traversal problem, no single solution works for all NAT devices and network environments. The following paragraphs review some representative techniques, including STUN [29, 36, 37], TURN [30, 36, 37] and ICE [31, 36].

STUN [29, 36, 37] is a suite of methods, which contains a network protocol used in NAT traversal. As Fig. 4 illustrates, the STUN protocol requires a third-party STUN server located on the external side of the NAT to help applications behind a NAT to discover the presence of a NAT. It also helps retrieve the mapped-address that NAT generates for the application's User Datagram Protocol (UDP) sessions.

TURN [30, 36, 37] utilizes a server in the public domain to relay packets created by two hosts behind different NATs. As Fig. 5 illustrates, Node A connects to a TURN server to request relay resource and inform Node B of the relay resource. Two hosts can communicate with each other by relaying their data through the TURN server.

ICE [31, 36] uses a set of methods including STUN and TURN for NAT transversal instead of using a single method. A host uses both techniques simultane-

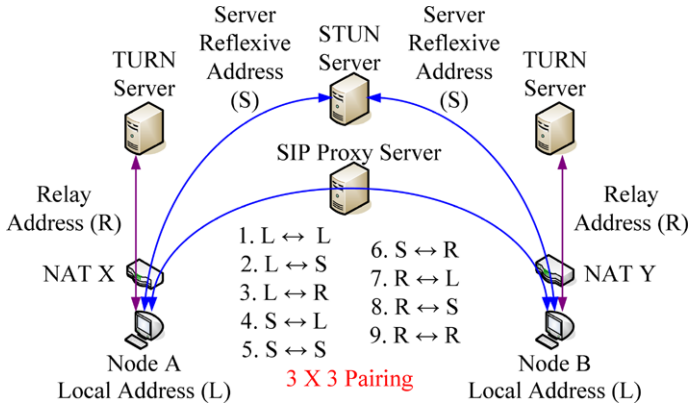

Fig. 6 ICE architecture

ously to obtain a set of possible IP addresses and ports (Fig. 6) for communication. The addresses and ports include a local address which is directly attached to the network interface, the server-reflexive address on the external side of a NAT, and the relay address allocated from a TURN server. Since each address and port represents a potential point of communications, both hosts exchange three candidate addresses after obtaining them, making total nine candidate communicating pairs. A connectivity check is then performed for each pair to verify if a session can be established with such pair.

2.2 Distance calculation and location estimation

In the design of WANTS, an SSPR should measure the received signal to find its location first. WiFi is one of the most widely adopted indoor wireless technologies. SSPR can retrieve the signal strength from the received signal strength indicator (RSSI) of beacons or probe response frames in 802.11. With the signal strength of each AP, SSPR can calculate the distance to each AP. With such distance information among SSPR and APs, SSPR can estimate its location with regard to the locations of APs. The following paragraphs illustrate how SSPR estimates its position with signal strength of each AP.

By using Free Space Propagation Formula, SSPR can calculate the distance to each AP with the received signal strength. For simplicity, we assume that both transmitting antennas and receiving antennas are located in an otherwise empty environment and are omni-directional. Neither absorbing obstacles nor reflecting surfaces are considered. The received signal power S_r at SSPR is

$$S_r = S_t \cdot G_t \cdot G_r \cdot \left(\frac{\lambda}{4\pi D_i}\right)^2, \tag{1}$$

where S_t is the transmitted power from the AP, G_t and G_r are the gain of the transmitting and receiving antenna, respectively, λ is the wavelength, and D_i is the distance between the ith AP and SSPR. For simplicity, we assume that each AP has a known fixed location and the transmitting power. After reorganizing (1), the distance D_i between SSPR and the ith AP can be represented as

$$D_i = \sqrt{\frac{S_t}{S_r} \cdot G_t \cdot G_r \cdot \left(\frac{\lambda}{4\pi}\right)^2}. \tag{2}$$

SSPR then uses the geographical information to estimate its location with regard to each AP with the obtained the distance information. Here we assume SSPR and all APs are on the same plane. One of the most common methods for SSPR to estimate its location is lateration, which is evolved from the triangulation method. With the calculated distance information (D_i) and the geographical information including the known positions (x_i, y_i) of the anchors (i.e., APs), we can derive the following series of equations:

$$\begin{aligned} &(x_1 - x)^2 + (y_1 - y)^2 = D_1^2 \\ &(x_2 - x)^2 + (y_2 - y)^2 = D_2^2 \\ &\vdots \\ &(x_i - x)^2 + (y_i - y)^2 = D_i^2 \end{aligned}, \tag{3}$$

where the unknown position (x, y) is the location of SSPR. After subtracting the last equation in (3) from

the first $i-1$ equations to linearize equations in (3), we have

$$\begin{aligned}
&x_1^2 - x_i^2 - 2(x_1 - x_i)x + y_1^2 - y_i^2 - 2(y_1 - y_i)y \\
&\quad = D_1^2 - D_i^2 \\
&x_2^2 - x_i^2 - 2(x_2 - x_i)x + y_2^2 - y_i^2 - 2(y_2 - y_i)y \\
&\quad = D_2^2 - D_i^2 \\
&\vdots \\
&x_{i-1}^2 - x_i^2 - 2(x_{i-1} - x_i)x + y_{i-1}^2 - y_i^2 \\
&\qquad - 2(y_{i-1} - y_i)y \\
&\quad = D_{i-1}^2 - D_i^2
\end{aligned} \quad . \tag{4}$$

Then we can reorder equations in (4) to derive a polynomial presentation in the form of $Ax = b$:

$$A = \begin{bmatrix} 2(x_1 - x_i) & 2(y_1 - y_i) \\ 2(x_2 - x_i) & 2(y_2 - y_i) \\ \vdots & \vdots \\ 2(x_{i-1} - x_i) & 2(y_{i-1} - y_i) \end{bmatrix}$$

$$b = \begin{bmatrix} x_1^2 - x_i^2 + y_1^2 - y_i^2 + D_i^2 - D_1^2 \\ x_2^2 - x_i^2 + y_2^2 - y_i^2 + D_i^2 - D_2^2 \\ \vdots \\ x_{i-1}^2 - x_i^2 + y_{i-1}^2 - y_i^2 + D_i^2 - D_{i-1}^2 \end{bmatrix} . \tag{5}$$

The above equation can be solved by using the least-square method, which is

$$\hat{x} = \left(A^T A\right)^{-1} A^T b. \tag{6}$$

3 WiFi Assisted NAT Traversal Scheme (WANTS) for SSPR

The basic idea of WANTS is that SSPR and the mobile device of the householder need not to try all possible communication paths after SSPR performs handoff. SSPR utilizes the topology information and network context information provided by the server to find the most promising paths for communication, while the mobile device uses the network context information only. When SSPR is activated, it initiates a request to retrieve the topology information and network context information. Then SSPR uses topology information to estimate its location and uses network context information to assist NAT traversal. As Fig. 7 illustrates, the request is embedded in the REGISTER message. After receiving such request, the Proxy Server will respond

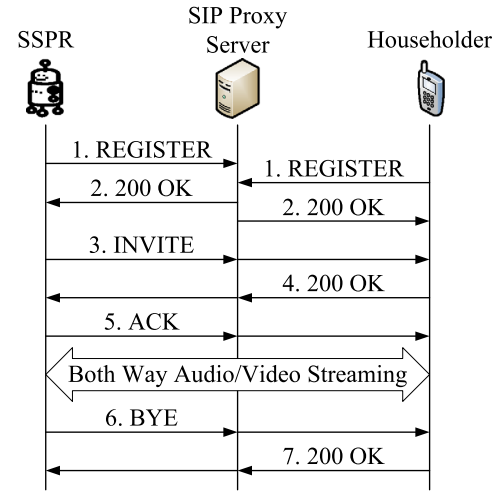

Fig. 7 Interaction between SSPR and the householder

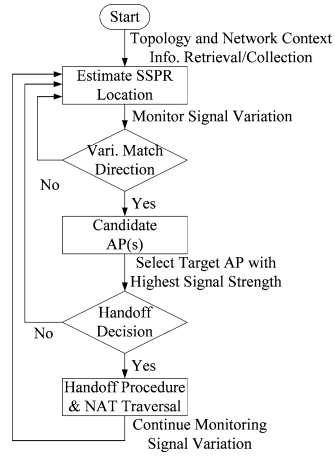

Fig. 8 WANTS operation

SSPR with topology information and network context information appended to a 200 OK message. The network context information includes host location (public/private domain) and NAT type (mapping behavior and filtering behavior) of each AP. In the case that network context information is not available, SSPR starts collecting such information and uploads to the server for subsequent access. The information retrieval procedure is identical to the mobile device, except that the topology information is useless for the mobile device.

After retrieving the position of each AP in the topology information, SSPR will estimate its location with regard to the coordinate of each AP as shown in Fig. 8. The estimation method is illustrated in the previous section. Then SSPR starts monitoring the signal

strength of each AP, comparing the variation of signal strength with its moving direction.

As Fig. 9 illustrates, if SSPR moves toward the center of AP2 and AP4, the signal strength of AP2 and AP4 increase smoothly while the signal strength of AP1 and AP3 decrease. As long as SSPR knows its location in the coordinate system and captures the variation of the signal strength, SSPR can determine the candidate APs for handoff. The candidate APs in Fig. 9 are AP2 and AP4. However, the signal strength may be interfered and thus such signal strength may not reflect the proper moving direction. Since SSPR is aware of moving direction, it compares the variation of signal strength with its moving direction to filter out the jammed signal. In the end, SSPR selects the AP with highest signal strength as the candidate AP for handoff and uses the network context information to assist NAT traversal after handoff.

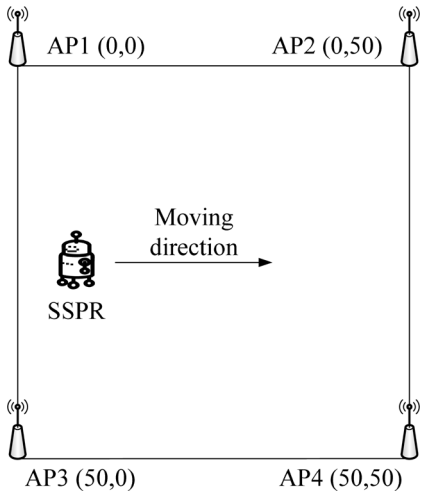

Fig. 9 Location information of each AP

When SSPR decides to perform a handoff, it chooses the AP with best signal strength and starts handoff procedure. SSPR then initiates NAT traversal after completing handoff. As Fig. 10 shows, when SSPR and the mobile device wish to setup a session, both exchange network context information through an independent signaling protocol (SIP). Based on the information, both then determine the most suitable communicating paths and the one to initiate the connectivity checks with WANTS connectivity check algorithm (WCCA). The following paragraphs illustrate the operations of WANTS with signaling protocol SIP.

1. The first step is to exchange network context information. When SSPR wishes to contact the mobile device, both can use an out-of-band signaling protocol to exchange network context information. In the case of SIP, SSPR uses an INVITE message to carry its network context information, while the mobile device uses 200 OK.
2. After acquiring complete network context information, both SSPR and the mobile device run WCCA to choose a candidate path and initiator for connectivity check. WCCA is described in the next subsection.
3. The last step of WANTS establishes the sessions so that SSPR and the mobile device can start transferring data.

3.1 WANTS Connectivity Check Algorithm (WCCA)

The WCCA uses the retrieved or collected network context information to choose the most promising paths for communication. Then both SSPR and the mobile device start connectivity check. As Fig. 11 illustrates, WCCA differentiates three different cases

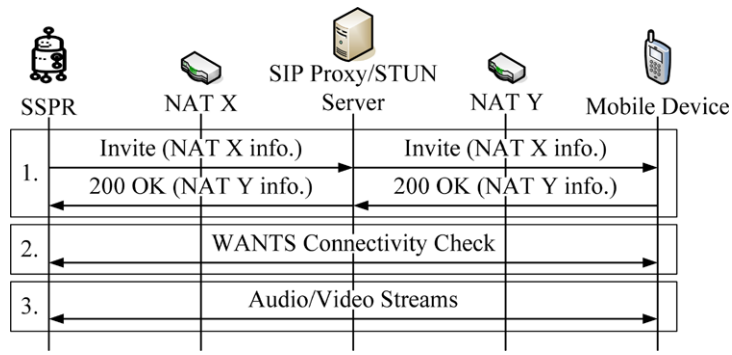

Fig. 10 WANTS operation for NAT Traversal

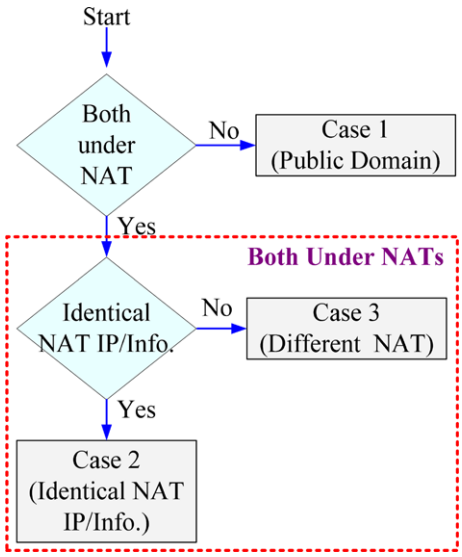

Fig. 11 Three major cases in WCCA

and tailors a specific connectivity check procedure for each case.

1. Case 1 corresponds to the condition that at least SSPR or the mobile device is in the public domain.
2. Case 2 corresponds to the condition that neither SSPR nor the mobile device is in the public domain but both have an identical server-reflexive address (i.e., both SSPR and the mobile device are behind the same NAT), or SSPR and the mobile device have different server-reflexive addresses but identical NAT information (i.e., SSPR and the mobile device *may* be behind the same NAT because some NATs have multiple public addresses).
3. Case 3 indicates the condition that both SSPR and the mobile device are behind different NATs with different NAT information.

The following paragraphs describe the detail operation of each case in WCCA.

Case 1 involves the case that at least SSPR or the mobile device is in the public domain. As Table 1 illustrates, there are two possible configurations in Case 1.

1. If both SSPR and the mobile device are in the public domain, they can use their local (public) addresses to proceed with the connectivity check.
2. If only SSPR or the mobile device is in the public domain, the one behind a NAT must use its server-reflexive address to initiate a connectivity check, while the other can use its local address.

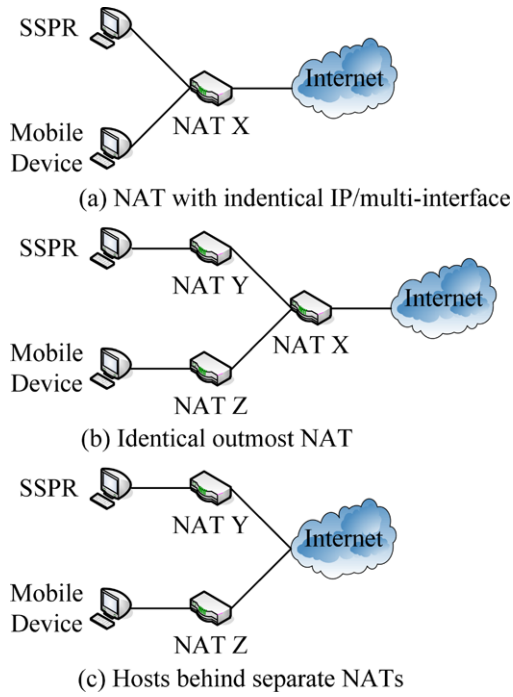

Fig. 12 Three possible topologies in Case 2

The WCCA tests only one candidate pair as compared to the original ICE which tests the nine candidate pairs. As a result, WCCA reduces both delay and message exchanges.

Case 2 indicates that SSPR and the mobile device have identical network context information (Table 1). There are three possible topologies as Fig. 12 illustrates and at most three address pairs are tested.

1. If a NAT has multiple public addresses, SSPR and the mobile device have identical network context information but different server-reflexive addresses (Fig. 12a). SSPR and mobile with identical server-reflexive address are certainly behind the same NAT. Since communications with local addresses consumes least resource, WCCA gives local-to-local (L-to-L) address pair the highest priority (Table 1).
2. In the other topologies (Fig. 12b and Fig. 12c), WCCA tries the candidate pair of two server-reflexive addresses first.
3. The relay path is the last choice when WCCA fails to find a path with server-reflexive addresses.

A SY NAT generates different mapped-addresses for different sessions and renders previously ex-

Table 1 Candidate path selection rule

	SSPR	Mobile device	Identical NAT Info.	NAT X	NAT Y	Candidate path (SSPR - M. Device)
Case 1	Public IP	Public IP	–	–	–	L-to-L
	Public IP	Private IP	–	–	–	L-to-S
	Private IP	Public IP	–	–	–	S-to-L
Case 2	Private IP	Private IP	Yes	–	–	L-to-L
				–	–	S-to-S
				SY	SY	R-to-R
				Non-SY	Non-SY	S-to-R
Case 3	Private IP	Private IP	No	SY	SY	R-to-R
				SY	PR	R-to-S
				PR	SY	S-to-R
				Non-SY	Non-SY	S-to-S

L: Local address,
S: Server-reflexive address,
R: Relay address

changed addresses invalid. As a result, the relay path selection is inevitable.

1. As Table 1 illustrates, if the outmost NAT (NAT X) is a SY NAT (Fig. 12b), SSPR and the mobile device must use their relay addresses to perform the connectivity check.
2. Otherwise, SSPR or the mobile device can use the server-reflexive address to perform the connectivity check, while the other uses the relay address. SSPR is the one to use the server-reflexive address in WCCA.

The approach of relay path selection also reduces the usage of the relay server and the latency between SSPR and the mobile device.

Case 3 considers the condition that SSPR and the mobile device are behind different NATs with different NAT information. As Table 1 illustrates, only one path is tested.

1. Both NATs are SY NATs. Since a SY NAT changes mapped-address for each session, hosts behind SY NATs cannot acquire effective mapped-addresses for connectivity check. As a result, both SSPR and the mobile device should proceed the connectivity check with relay addresses.
2. A combination of SY NAT and PR NAT. The one behind the PR NAT can use a server-reflexive address while the one behind the SY NAT should use a relay address to perform the connectivity check. This is because sessions from the same internal IP address and port generate an identical mapped-address at the PR NAT (independent mapping behavior).
3. The rest of the cases not mentioned above can use the candidate pair of server-reflexive addresses for the connectivity check.

Table 2 NAT devices used in the experiments

No.	Brand	Model
1	D-Link	DI-604
2	SMC	SMCWBR14-G2
3	Corega	CG-BARMX2
4	Planex	BLW-54MR
5	SMC	SMCWGBR-14N
6	Belkin	F5D8231TW4
7	Draytek	Vigor 2104P
8	Lemel	LM-WLG6400
9	3Com	3CRWER100-75
10	Asus	RX3041

4 Experiment results

To study the performance of the proposed approach and compare it with that of other alternative, we conducted experiments with off-the-shelf NAT devices as shown in Table 2. Our experiments use 10 NAT devices and result in total $10 * 10 = 100$ possible NAT combinations for SSPR-mobile device pairs. The alternative approach is ICE because ICE is one of the most promising solutions to NAT traversal.

We first compare WANTS with ICE and verify if they have identical direct communication ratio (DCR).

Then we proceed with experiments of connectivity check delay and resource demand (i.e., the number of connectivity check messages) to find the amount of average delay and resource that WANTS reduced. The following presents the experimental results.

4.1 Direct Communication Ratio (DCR)

Table 3 shows the behaviors of all NAT devices. With this information, SSPR and the mobile device can perform connectivity checks with appropriate path.

Figure 13 shows the test result of direct communication and indicates whether a direct communication path can be established between each SSPR-mobile device pair. A shaded area in Fig. 13 indicates that both WANTS and ICE can establish a direct communication path for the corresponding SSPR-mobile device pair, while a blank area means that neither WANTS nor ICE can. Both WANTS and ICE have identical DCR. Both DCR are $92/100 = 92\ \%$.

Table 3 NAT Behavior of each device

NAT type	NAT No.
FC	1, 2
AR	3, 4, 5
PR	6, 7, 8, 9
SY	10

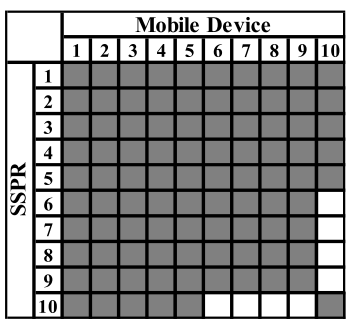

Fig. 13 The direct communication status of all NAT combinations of ICE/WANTS

4.2 Connectivity check delay

We also measured average connectivity check delay for each type of NAT combinations. The results shown in Table 4 (observed by SSPR) and Table 5 (observed by Mobile Device) reveal that the average delays of SSPR and the mobile device in ICE are 4.25 s and 4.48 s, respectively. The results of WANTS show 0.02 s average delay of SSPR and 0.35 s average delay of the mobile device. SSPR in ICE consumes more than 4.23 s than that of WANTS in average. The time that the mobile device takes for connectivity check in ICE is almost 10 times longer than that of WANTS. WANTS reduces unnecessary delay by using network context information to selectively perform connectivity checks while ICE performs connectivity checks sequentially for all (9) candidate paths and results longer delay.

4.3 Resource demand

Figure 14 shows the total number of protocol messages required by WANTS and ICE of overall connectivity checks. A successful connectivity check requires four to six messages depending on the type of the NAT pair under check, while an unsuccessful one takes six messages. Experimental results indicate that the number of messages WANTS generated is much fewer than that of ICE. This is because WANTS selects the most promising paths for connectivity checks with WCCA, while ICE performs connectivity checks for all possible paths. According to experimental results, ICE produces almost nine times the connectivity check messages than that of WANTS.

Table 4 Average connectivity check delay observed by SSPR

SSPR	Mobile device							
	FC		AR		PR		SY	
	WANTS	ICE	WANTS	ICE	WANTS	ICE	WANTS	ICE
FC	0.01 s	4.10 s	0.04 s	4.34 s	0.01 s	4.28 s	0.19 s	4.10 s
AR	0.04 s	4.29 s	0.01 s	4.19 s	0.01 s	4.29 s	0.19 s	4.19 s
PR	0.02 s	4.26 s	0.01 s	4.24 s	0.01 s	4.34 s	0.01 s	4.02 s
SY	0.01 s	4.19 s	0.01 s	4.39 s	0.01 s	4.18 s	0.01 s	4.01 s

Table 5 Average connectivity check delay observed by the mobile device

SSPR	Mobile device							
	FC		AR		PR		SY	
	WANTS	ICE	WANTS	ICE	WANTS	ICE	WANTS	ICE
FC	0.32 s	4.33 s	0.36 s	4.47 s	0.33 s	4.46 s	0.50 s	4.43 s
AR	0.36 s	4.50 s	0.34 s	4.50 s	0.33 s	4.50 s	0.50 s	4.50 s
PR	0.35 s	4.50 s	0.33 s	4.50 s	0.34 s	4.50 s	0.33 s	4.42 s
SY	0.33 s	4.49 s	0.37 s	4.50 s	0.34 s	4.50 s	0.33 s	4.00 s

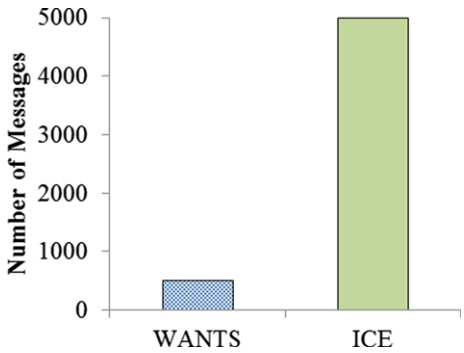

Fig. 14 Number of connectivity check messages produced by WANTS and ICE

5 Conclusion

This study proposes an improved NAT traversal scheme WANTS for SSPR. SSPR can utilize the topology information to choose the target AP for handoff and use the network context information to assist its NAT traversal. WANTS not only shortens the connectivity delay after handoff, but also reduces the required protocol messages when performing connectivity checks. Furthermore, both SSPR and the mobile device collect and store network context information when they are activated. When performing connectivity checks, both exchange their network context information to help eliminate unnecessary checks. Experimental results confirm that WANTS outperforms ICE in terms of connectivity check delay and resource demand while maintaining the same DCR as ICE.

Acknowledgements The authors acknowledge the support from the National Science Council of the Republic of China, under Grant Nos. NSC 100-2221-E-009-072-MY3, NSC 101-2219-E-009-028, NSC 100-2628-E-153-001 and NSC 101-2221-E-153-001-MY2, and D-Link Co., Taiwan. The authors are also most grateful for the kind assistance of Professor Ali H. Nayfeh, Editor of Nonlinear Dynamics, and for the constructive suggestions from the anonymous reviewers, all of which has led to the making of several corrections and have greatly aided us to improve the presentation of this paper.

References

1. Chen, C.W.: Modeling and control for nonlinear structural systems via a NN-based approach. Expert Syst. Appl. **36**, 4765–4772 (2009)
2. Chen, C.W.: GA-based adaptive neural network controllers for nonlinear systems. Int. J. Innov. Comput., Inf. Control **6**, 1793–1803 (2010)
3. Chen, C.W.: Stability analysis and robustness design of nonlinear systems: an NN-based approach. Appl. Soft Comput. **11**(2), 2735–2742 (2011)
4. Chen, C.W.: Application of fuzzy-model-based control to nonlinear structural systems with time delay: an LMI method. J. Vib. Control **16**, 1651–1672 (2010)
5. Chen, P.C., Chen, C.W., Chiang, W.L.: GA-based modified adaptive fuzzy sliding mode controller for nonlinear systems. Expert Syst. Appl. **36**, 5872–5879 (2009)
6. Gholami, A., Markazi, A.H.D.: A new adaptive fuzzy sliding mode observer for a class of MIMO nonlinear systems. Nonlinear Dyn. **70**(3), 2095–2105 (2012)
7. Chen, C.Y., Shih, B.Y., Shih, C.H., Wnag, L.H.: Design, modeling and stability control for an actuated dynamic walking planar bipedal robot. J. Vib. Control (2012). doi:10.1177/1077546311429476
8. Chen, C.Y.: The motion editor and high precision integration for optimal control of robot manipulators in dynamic structural systems. Struct. Eng. Mech. **41**, 633–644 (2012)
9. Chung, P.Y., Chen, Y.H., Walter, L., Chen, C.Y.: Influence and dynamics of a mobile robot control on mechanical components. J. Vib. Control (2012). doi:10.1177/1077546312452184
10. Pinto, C.M.A.: Stability of quadruped robots' trajectories subjected to discrete perturbations. Nonlinear Dyn. **70**(3), 2089–2094 (2012)
11. Fateh, M.M., Khorashadizadeh, S.: Optimal robust voltage control of electrically driven robot manipulators. Nonlinear Dyn. **70**(2), 1445–1458 (2012)
12. Hsiao, F.H., Chen, C.W., Liang, Y.W., Xu, S.D., Chiang, W.L.: T-s fuzzy controllers for nonlinear interconnected systems with multiple time delays. IEEE Trans. Circuits. Syst., I Regul. Pap. **52**, 1883–1893 (2005)
13. Chen, C.W., Yeh, K., Liu, K.F.R., Lin, M.L.: Applications of fuzzy control to nonlinear time-delay systems using the

linear matrix inequality fuzzy Lyapunov method. J. Vib. Control (2012). doi:10.1177/1077546311410765
14. Chen, P.C., Chen, C.W., Chiang, W.L.: Linear matrix inequality conditions of nonlinear systems by genetic algorithm-based H (infinity) adaptive fuzzy sliding mode controller. J. Vib. Control **17**(2), 163–173 (2011)
15. Fateh, M.M., Khorashadizadeh, S.: Robust control of electrically driven robots by adaptive fuzzy estimation of uncertainty. Nonlinear Dyn. **69**(3), 1465–1477 (2012)
16. Pratiher, B., Bhowmick, S.: Nonlinear dynamic analysis of a Cartesian manipulator carrying an end effector placed at an intermediate position. Nonlinear Dyn. **69**(1–2), 539–553 (2012)
17. Chen, C.W.: Adaptive fuzzy sliding mode control for seismically excited bridges with lead rubber bearing isolation. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **17**, 705–727 (2009)
18. Chen, C.W.: Modeling and fuzzy PDC control and its application to an oscillatory TLP structure. Math. Probl. Eng. (2010). doi:10.1155/2010/120403
19. Chen, C.Y.: Fuzzy control for an oceanic structure: a case study in time-delay TLP system. J. Vib. Control **16**(1), 147–160 (2010)
20. Chen, C.Y., Shih, B.Y., Shih, C.H., Chou, W.C.: The development of autonomous low-cost biped mobile surveillance robot by intelligent bricks. J. Vib. Control **18**(5), 577–586 (2012)
21. Tseng, C.C., Lin, C.L., Shih, B.Y., Chen, C.Y.: SIP-enabled surveillance patrol robot. Robot. Comput.-Integr. Manuf. **29**(2), 394–399 (2013)
22. Francis, P., Egevang, K.: Traditional IP Network Address Translator (Traditional NAT). IETF RFC 3022 (2001)
23. Stegel, T., Sterle, J., Sedlar, V., Bester, J., Kos, A.: SCTP multihoming provisioning in converged IP-based multimedia environment. Comput. Commun. **33**(14), 1725–1735 (2010)
24. Lin, Y.D., Tseng, C.C., Ho, C.Y., Wu, Y.H.: How NAT-compatible are VoIP applications? IEEE Commun. Mag. **48**(12), 58–65 (2010)
25. Aurel Constantinescu, M., Croitoru, V., Oana Cernaianu, D.: NAT/Firewall traversal for SIP: issues and solutions. In: Int. Symp. on Signals, Circuits and Syst, vol. 2, pp. 521–524 (2005)
26. Ho, C.-Y., Wang, F.-Y., Tseng, C.-C., Lin, Y.-D.: NAT-compatibility testbed: an environment to automatically verify direct connection rate. IEEE Commun. Lett. **15**(1), 4–6 (2011)
27. Yoshimi, H., Enomoto, N., Cui, Z., Takagi, K., Iwata, A.: NAT traversal technology of reducing load on relaying server for P2P connections. In: Consum. Commun. and Netw. Conf., pp. 100–104 (2007)
28. Saikat, G., Yutaka, T., Paul, F.: NUTSS: a SIP-based approach to UDP and TCP network connectivity. In: Proc. of the ACM SIGCOMM Workshop on Future Dir. in Netw. Archit (2004)
29. Rosenberg, J., Weinberger, J., Huitema, C., Mahy, R.: Session Traversal Utilities for NAT (STUN). IETF RFC 5989 (2008)
30. Rosenberg, J., Mahy, R., Matthews, P., Huitema, C.: Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN). IETF RFC 5766 (2010)
31. Rosenberg, J.: Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols. IETF RFC 5425 (2010)
32. Cuevas, R., Cuevas, Á., Cabellos-Aparicio, A., Jakab, L., Guerrero, C.: A collaborative P2P scheme for NAT traversal server discovery based on topological information. Comput. Netw. **54**(12), 2071–2085 (2010)
33. Chen, Y.-C., Jia, W.-K.: Challenge and solutions of NAT traversal for ubiquitous and pervasive applications on the Internet. J. Syst. Softw. **82**(10), 1620–1626 (2009)
34. Patro, A., Ma, Y., Panahi, F., Walker, J., Banerjee, S.: A system for audio signalling based NAT traversal. In: Third International Conference on Communication Systems and Networks, pp. 1–10 (2011)
35. Houngue, P., Damiani, E., Glitho, R.: Overcoming NAT traversal issue for SIP-based communication in P2P networks. In: 4th Joint IFIP Wireless and Mobile Networking Conference, pp. 1–8 (2011)
36. Maenpaa, J., Andersson, V., Camarillo, G., Keranen, A.: Impact of network address translator traversal on delays in peer-to-peer session initiation protocol. In: 2010 IEEE Global Telecommunications Conference, pp. 1–6 (2010)
37. Langendoen, K., Reijers, N.: Distributed localization in wireless sensor networks: a quantitative comparison. Comput. Netw. **43**(4), 499–518 (2003)
38. Ho, C.Y., Tseng, C.C., Wang, F.Y., Wang, J.T., Lin, Y.D.: To call or to be called behind NATs is sensitive in solving the direct connection problem. IEEE Commun. Lett. **15**(1), 94–96 (2010)
39. Rosenberg, J., Weinberger, J., Huitema, C., Mahy, R.: STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs). IETF RFC 3489 (2003)
40. Audet, F., Jennings, C.: Network Address Translation (NAT) Behavioral Requirem. IETF RFC 4787 (2007)