# Enhancing Teredo IPv6 Tunneling to Traverse the Symmetric NAT

Shiang-Ming Huang, Quincy Wu, and Yi-Bing Lin, *Fellow, IEEE*

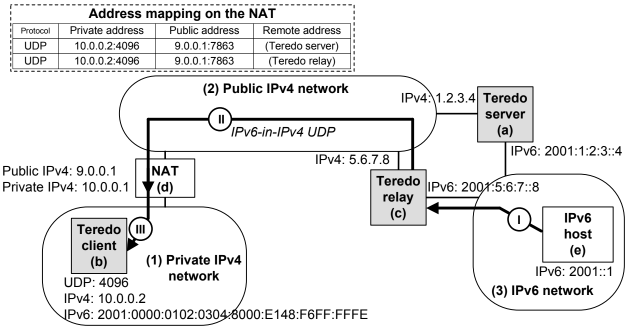
***Abstract*— By tunneling IPv6 packets over IPv4 UDP, Teredo supports IPv4/IPv6 dual-stack nodes in private IPv4 networks behind Network Address Translation (NAT) to access IPv6 networks. However, the current Teredo protocol does not work with symmetric NAT. This letter proposes *SymTeredo*, an extension of Teredo with capability to traverse the symmetric NAT. Our extension preserves the Teredo architecture, and offers backward compatibility with the original Teredo protocol.**

***Index Terms*— IPv6, NAT, Teredo, tunneling.**

## I. Introduction

Internet Protocol version 6 (IPv6) has been developed as the next generation Internet protocol. Compared with the Internet Protocol version 4 (IPv4), IPv6 provides larger address space, more efficient routing mechanism, better support for security and quality of service. During IPv6 deployment, it is required to upgrade IPv4 networks to IPv6 networks. To facilitate the transition for IPv4-to-IPv6 migration, *tunneling* techniques are utilized to connect isolated IPv6 nodes through IPv4 networks [1], [2]. Traditional tunneling methods carry IPv6 packets as payload of IPv4 packets. However, these methods fail when one endpoint of the tunnel is located in a private IPv4 network behind a Network Address Translation (NAT) server.

Several IPv6 tunneling solutions [1] have been proposed to resolve the NAT traversal issue. Among these solutions, IETF v6ops Working Group chooses Teredo as the protocol for clients to traverse the NAT and automatically establish IPv6 tunnels in an unmanaged network. A major advantage of Teredo is its load-balancing design that utilizes a centralized Teredo server for signaling and several Teredo relays for data packets delivery. In this architecture, the user traffics can be distributed among Teredo relays to reduce the bottleneck effect [2].



| Protocol | Private address | Public address | Remote address |
|---|---|---|---|
| UDP | 10.0.0.2:4096 | 9.0.0.1:7863 | (Teredo server) |
| UDP | 10.0.0.2:4096 | 9.0.0.1:7863 | (Teredo relay) |

Fig. 1. Teredo architecture.

The Teredo architecture consists of a Teredo server (Fig. 1 (a)), several Teredo clients (Fig. 1 (b)) and Teredo relays (Fig. 1 (c)). A Teredo client runs at an IPv4/IPv6 dual-stack host in a private IPv4 network (Fig. 1 (1)) that connects to public IPv4 networks (Fig. 1 (2)) through an NAT (Fig. 1 (d)). A Teredo server helps Teredo clients to access IPv6 networks (Fig. 1 (3)). A Teredo relay forwards IPv6 traffic between a Teredo client and a host in the IPv6 network. A Teredo client (with IPv4 address 10.0.0.2 and UDP port number 4096; see Fig. 1 (b)) exchanges IPv4 UDP messages with a Teredo server to detect the NAT type and its mapped public IPv4 address and UDP port number on the NAT (9.0.0.1 and 7863; see Fig. 1 (d)). The mapped address and port number are encoded in the Teredo client's IPv6 address (e.g., 2001:0000:0102:0304:8000:E148:F6FF:FFFE, where 9.0.0.1 and 7863 are represented by the 32-bit value 0xF6FFFFFE and 16-bit value 0xE148, respectively) to identify the NAT. After the Teredo client has acquired an IPv6 address from the Teredo server, this IPv6 address is known to the hosts in the IPv6 networks via mechanisms such as dynamic DNS. For IPv6 packets sent from an IPv6 host (Fig. 1 (e)) in an IPv6 network, a Teredo relay closest to the IPv6 host is dynamically chosen based on standard IPv6 routing protocols. The Teredo relay utilizes an IPv6-in-IPv4 UDP tunnel to deliver the IPv6 packet to the Teredo client, as described in the following steps.

Step I: The IPv6 host (Fig. 1 (e)) in the IPv6 network attempts to send an IPv6 packet to the Teredo client. The packet is first sent to the chosen Teredo relay.

Step II: The Teredo relay encapsulates the IPv6 packet in an IPv4 UDP packet. The destination IPv4 address and port number for the UDP packet are automatically derived from the destination IPv6 address, which are the mapped IPv4 address and port number on the NAT for the Teredo client (9.0.0.1 and 7863 in Fig. 1). The Teredo relay sends the encapsulated packet to the NAT.

Step III: When the NAT receives the packet, it uses the port number 7863 as the key to retrieve the private IPv4 address 10.0.0.2 and port number 4096 from its mapping table. Then it sends the IPv4 UDP packet to the Teredo client in the private IPv4 network.

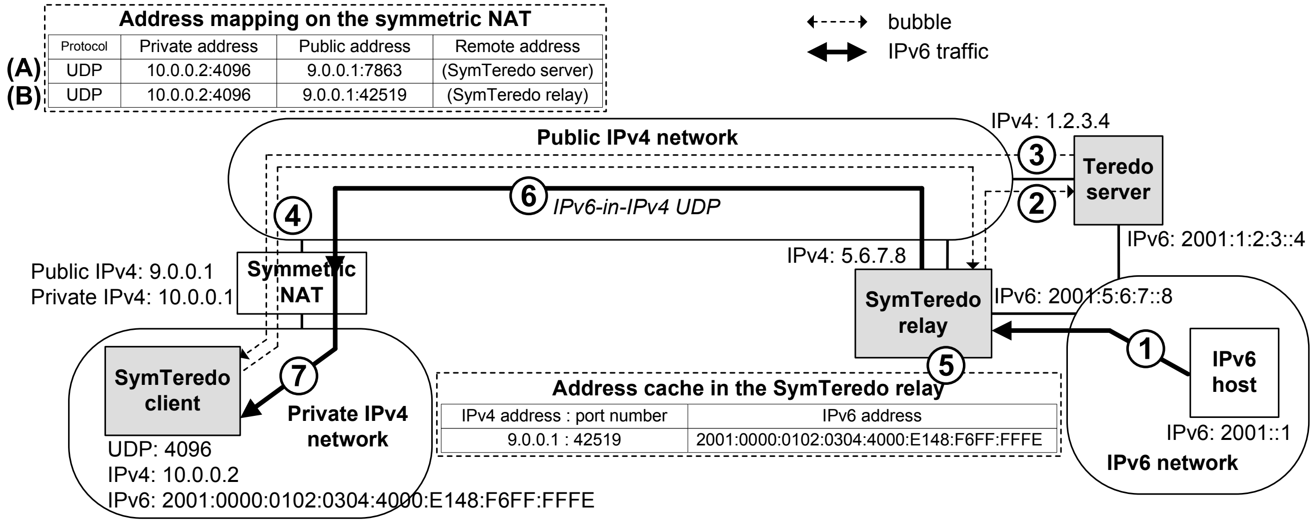
Upon receipt of the packet, the Teredo client decapsulates

Manuscript received November 28, 2005. The associate editor coordinating the review of this paper and approving it for publication was Dr. Nikos Nikolaou.

S.-M. Hunag and Y.-B. Lin are with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan (e-mail: {smhuang, liny}@csie.nctu.edu.tw).

Q. Wu is with the Graduate Institute of Communication Engineering, National Chi Nan University, Nantou, Taiwan. (e-mail: solomon@ipv6.club.tw).

Digital Object Identifier 10.1109/LCOMM.2006.05004.



Address mapping on the symmetric NAT

| | Protocol | Private address | Public address | Remote address |
|---|---|---|---|---|
| (A) | UDP | 10.0.0.2:4096 | 9.0.0.1:7863 | (SymTeredo server) |
| (B) | UDP | 10.0.0.2:4096 | 9.0.0.1:42519 | (SymTeredo relay) |

Address cache in the SymTeredo relay

| IPv4 address : port number | IPv6 address |
|---|---|
| 9.0.0.1 : 42519 | 2001:0000:0102:0304:4000:E148:F6FF:FFFE |

Fig. 2. Packet delivery through symmetric NAT using SymTeredo.

the IPv4 UDP packet to obtain the IPv6 packet.

There is a serious problem about current Teredo protocol. Among the four major NAT types [3], Teredo fails in traversing symmetric NAT [2]. One possible solution was proposed in [4] which routes IPv6 packets through the Teredo server instead of the Teredo relays. However, this approach violates the load-balancing design of Teredo and imposes heavy loading on the Teredo server. Therefore, it was not adopted by the IETF v6ops Working Group. In this letter, we propose *SymTeredo*, an extension of Teredo with symmetric NAT traversal capability. SymTeredo only requires minor modifications to the Teredo relay and the Teredo client components. The SymTeredo provides backward compatibility with the current Teredo protocol without violating the distributed load-balancing design.

## II. Packet Delivery using SymTeredo

As we previously mentioned, when the Teredo client acquires an IPv6 address (e.g., 2001:0000:0102:0304:8000:E148:F6FF:FFFE) from the Teredo server, the mapped public IPv4 address and UDP port number (e.g., 9.0.0.1 and 7863) are encoded in this IPv6 address. The encoded address will be used by the Teredo relay as the destination of the encapsulated packet. However, a symmetric NAT server will assign different mapped port numbers for each pass-through flow. That is, two IPv4 UDP packets sent from the same private IPv4 host to different public IPv4 hosts are translated to the same mapped public IPv4 address but different port numbers. Therefore, the address mapping created for the flow between the Teredo client and the Teredo server (stored in the symmetric NAT server; see (A) in Fig. 2) is different from that (stored in the symmetric NAT server; see (B) in Fig. 2) for the flow between the Teredo client and the Teredo relay. Unfortunately, the mapped public IPv4 address and port number (see (A) in Fig. 2)) encoded in the Teredo client's IPv6 address will be used by the Teredo relay to determine the target address. Hence the Teredo relay forwards IPv6 packets to the Teredo client using address mapping (A) instead of (B) without success. To fix this problem, we need to find out the correct mapped public IPv4 address and port number (see (B) in Fig. 2) on the symmetric NAT so that the Teredo relay knows the correct destination for packet delivery.

To support symmetric NAT traversal, SymTeredo slightly modifies the Teredo relay and the Teredo client without modifying the Teredo server. We use an example to illustrate how SymTeredo works. In Fig. 2, the modified Teredo client (SymTeredo client) has acquired an IPv6 address (2001:0000:0102:0304:4000:E148:F6FF:FFFE in Fig. 2) and created an address mapping on the NAT (Fig. 2 (A)) using the current Teredo protocol described in Section I. The acquired IPv6 address includes a symmetric flag (0x4000) to indicate that the NAT server is symmetric. With an *address cache* implemented in the modified Teredo relay (SymTeredo relay), an IPv6 host in the IPv6 network will be able to deliver an IPv6 packet to the SymTeredo client through the following steps.

Step 1:The IPv6 packet is sent to a SymTeredo relay. This SymTeredo relay is chosen according to the standard Teredo protocol described in Section I.

Step 2:The SymTeredo relay detects the symmetric flag in the destination SymTeredo client's IPv6 address. The SymTeredo relay buffers this IPv6 packet for later transmission (at Step 5), and sends a bubble to the Teredo server following the standard Teredo bubble transmission procedure [2]. The bubble is an IPv6-in-IPv4 UDP encapsulation packet, which contains the IPv6 address of the destination SymTeredo client and the IPv6 address of the source SymTeredo relay.

Step 3:Upon receipt of the bubble, the Teredo server inserts the source IPv4 address and UDP port number of the SymTeredo relay where this bubble is sent from, and then forwards this bubble to the mapped IPv4 address and port number encoded in the destination SymTeredo client's IPv6 address (see Step II in Section I). The NAT translates this packet and forwards it to the SymTeredo client.

Step 4:Upon receipt of this modified bubble, the SymTeredo client sends a response to the SymTeredo relay through the symmetric NAT. In this step, an address mapping (Fig. 2 (B)) is created on the NAT. This new address mapping will be used by the SymTeredo relay to reach the SymTeredo client (at Step 6).

Step 5: Upon receipt of the response, the SymTeredo relay stores the source IPv4 address and UDP port number of this packet (9.0.0.1 and 42519) in an *address cache* where the SymTeredo client's IPv6 address (2001:0000:0102:0304:4000:E148:F6FF:FFFE) is used as a key to search the *address cache* for retrieving these values. After the SymTeredo relay obtains this mapped address and port number, the previously buffered IPv6 packet (at Step 2) is retrieved for transmission.

Step 6: The SymTeredo relay encapsulates the IPv6 packet in an IPv4 UDP packet. The destination IPv4 address and UDP port number are retrieved from the *address cache* using the destination SymTeredo client's IPv6 address as the index key. The SymTeredo relay sends the encapsulated packet to the NAT.

Step 7: The NAT translates the packet according to the address mapping (Fig. 2 (B)), and then sends it to the SymTeredo client.

With the above steps, an IPv6 packet sent from an IPv6 host to the SymTeredo client can successfully pass through the symmetric NAT. For subsequent IPv6 packets sent from this IPv6 host to the SymTeredo client, the bubble transmission (described in Steps 2-5 above) need not be executed because the SymTeredo relay has already obtained the correct mapped IPv4 address and port number in its *address cache*. For SymTeredo clients in private IPv4 networks behind non-symmetric NATs (i.e., the symmetric flags in their IPv6 addresses are not set), the SymTeredo relay forwards packets as an ordinary Teredo relay described in Steps I-III in Section I. This enables SymTeredo to offer backward compatibility with the current Teredo protocol.

## III. Modifications Made to SymTeredo

To support SymTeredo, four minor modifications in the current Teredo protocol are needed.

1) In SymTeredo client's IPv6 address format, a bit in the reserved field is used to represent the symmetric flag. The SymTeredo client sets this flag in its IPv6 address when the NAT is detected to be symmetric.
2) In the SymTeredo relay, the standard Teredo bubble transmission procedure is executed before packet delivery when the symmetric flag in the destination IPv6 address is set.
3) In the SymTeredo relay, an *address cache* is implemented to store the mapped IPv4 addresses and UDP port numbers for SymTeredo clients. The SymTeredo relay determines the IPv4 UDP destination of an encapsulated packet according to the corresponding mapping stored in the *address cache* when the symmetric flag in the destination IPv6 address is set.
4) In the SymTeredo relay, checking on the source port numbers of incoming IPv4 UDP packets is skipped when the symmetric flag in the source IPv6 address is set. This check skipping may bring additional security threats to the SymTeredo relay. Fortunately, strategies proposed in [2] can eliminate the potential problems.

## IV. Summary

Teredo allows IPv6 users in private IPv4 networks behind NATs to access IPv6 networks. Current Teredo solution does not work with symmetric NAT. This letter proposed SymTeredo to support Teredo for symmetric NAT traversal. Our solution is backward compatible with the standard protocol.

## References

[1] S.-M. Huang, Q. Wu, and Y.-B. Lin, "Tunneling IPv6 through NAT with Teredo mechanism," in *Proc. IEEE 19th International Conference on Advanced Information Networking and Applications (AINA 2005)*, vol. 2, pp. 813-818.

[2] C. Huitema, "Teredo: tunneling IPv6 over UDP through NATs," IETF RFC 4380, Feb. 2006.

[3] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) through network address translators (NATs)," IETF RFC 3489, Mar. 2003.

[4] C. Huitema, "Teredo and symmetric NAT traversal," http://ops.ietf.org/lists/v6ops