# 可穿越網路位址轉換器的第六代網際網路通訊協定隧道

**Shiang-Ming Huang Quincy Wu Whai-En Chen**
**National Chiao Tung University**
**E-mail: {smhuang, solomon, wechen}@csie.nctu.edu.tw**

## 摘要

在現有的 IPv4 網路中引入 IPv6 時，Tunneling 是經常被使用到的機制。由於 IPv6 的網路尚未普及，IPv6 網路設備往往需要經由現有的 IPv4 網路以 Tunneling 的機制傳送 IPv6 封包，才能與其他 IPv6 網路互通。然而，如果 IPv6 設備在 NAT 後端，傳統的 Tunneling 機制完全沒有辦法幫助它連上 IPv6 網路。本篇文章將就 IPv6 Tunneling 相關的議題討論，介紹 Teredo 這個穿越 NAT 的 IPv6 Tunneling 機制，同時提出我們在交通大學所發展的 Teredo 封包分析器，以及所進行的 IPv6 Tunneling 實驗。

**關鍵詞**：第六代網際網路協定、隧道、網路位址轉換器。

## Abstract

Tunneling is a popular mechanism for IPv6 network deployment. Since IPv6 network is not widely deployed all over the world, IPv6 traffic must be transported over existing IPv4 network using the tunneling mechanism. However, if the IPv6 network device locates behind NAT(s), traditional IPv6 tunneling mechanisms will fail. This paper discusses issues related to IPv6 tunneling. A tunneling mechanism, Teredo, which can successfully tunnel IPv6 packets through NATs, is introduced here. The Teredo protocol dissector is also demonstrated, and our IPv6 tunneling experiment in NCTU is described.

**Keywords**: IPv6, Tunnel, NAT.

## 1. Introduction

為使現有的 IPv4 網路能平順地轉換到 IPv6，IETF NGtrans 工作小組所擬定的方針[3]，大致可分為三個方面，分別解決不同的需求：

1. Dual-Stack

目的在使 IPv4 及 IPv6 兩個通訊協定可在相同的設備上及相同的網路中共存。如此即無需在導入 IPv6 時，造成雙倍的硬體投資。目前 router 廠商如 Cisco、Juniper、Nortel、Nokia、Hitachi 等，伺服主機作業系統如 HP、IBM、Sun、Microsoft Windows、Linux、FreeBSD 等，均已支援 IPv4/IPv6 Dual-Stack。

2. Tunneling

目的在使未直接相連的 IPv6 網路彼此間能夠互通。如此即使所有網路導入 IPv6 的時程先後不一，但依然能維持彼此間的互通。現有做法如 Configured & Automatic Tunnel (RFC 2893)、6to4 Tunnel (RFC 3056)、Tunnel Broker (RFC 3053)、ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) 等。

3. Translation

使 IPv6 網路能與現存的 IPv4 網路互通，以共享現有 IPv4 網路豐富的網路資源及使用族群。目前較流行的做法有 SIIT(RFC 2765)、NAT-PT (RFC 2766)、BIS (RFC 2767)、BIA(RFC 3338)等。

其中現有 Tunneling 的方式，不論是 Configured Tunnel 或是 6to4 Automatic Tunnel 等做法，均要求建立 Tunnel 的 client 必需具備一個 public routable IPv4 位址。但對目前許多有興趣嘗試 IPv6 的玩家而言，他們所處的網路若在 NAT(Network Address Translator)之內, 就無法以 Tunneling 的方式連到世界上的 IPv6 網路，成為在現存環境下的「孤島」。由於現有的許多連線服務，像是 WLAN 或 GPRS，大多透過 NAT 以 private IPv4 位址的方式提供服務，因此在未來所有網路設備全面支援 IPv6 的那一天到來之前，有必要為這些位於 NAT 後的 IPv6 使用者提供一個解決方案，以使他們能順利與世界上所有的 IPv6 網路相連。

為了解決 IPv6 Tunneling 因為 NAT 產生的問題，IETF(Internet Engineering Task Force)的 v6ops Working Group 中，提出一個讓 NAT 後端設備用 IPv6 Tunnel 連上 IPv6 網路的新規格[4]，我們除了已針對此通訊協定開發封包分析器及進行基本測試外，並將在交大實地佈建這套系統，提供 NAT 後端設備連上 IPv6 網路的媒介，讓 NAT 後端網路不再是 IPv6「孤島」。

本篇文章第 2 節將說明 NAT 後端電腦的 Tunneling 問題，第 3 節是 Teredo 運作機制的介紹，第 4 節介紹 Teredo IPv6 位址編碼的方式，第 5 節簡介 Teredo 服務在交大實驗的構想和初步成果，最後是結論。

## 2. Problems Caused by NATs

為什麼在 NAT 後端的電腦沒有辦法利用 IPv6-in-IPv4 Tunnel 連上 IPv6 網路？下面我們將指出 NAT 讓 IPv6-in-IPv4 Tunnel 無法運作的原因。

過去所有的 IPv6-in-IPv4 Tunnel，包括 Configured Tunnel、6to4 Tunnel 和 6over4 Tunnel，都是依據 Tunnel 兩端設備的 IPv4 位址來做設定，因此必須使用 public IPv4 位址。如果是在 NAT 內部使用 private IPv4 位址的機器，會產生什麼問題呢？我們以 6to4 Tunnel[1]為例，這種 Tunnel 不需要手動設定，它由嵌在目的 6to4 IPv6 位址中的 IPv4 位址，得到用來建 Tunnel 的目的 IPv4 位址。6to4 Tunnel 在 NAT 後面的情況如圖 1，NAT 後端有一台 IPv6 電腦 A，A 想要用 6to4 Tunnel 連上 IPv6 網路。假設 NAT 後端 6to4 Router B 的 IPv4 位址為 10.0.0.1，電腦 A 的 IPv6 位址為 2002:A00:1:210::3/64(IPv6 位址的第 16 到 47 的 32 bits 嵌入 6to4 Router B 的 IPv4 位址)；NAT 的外部 IPv4 位址是 140.113.131.1；Internet 上 6to4 Router C 的 IPv4 位址是 61.218.105.10，IPv6 網路上電腦 D 的 IPv6 位址是 2001:238:F88::1/64。A 送資料給 D 的步驟如下：

①—A 把要送給 D 的資料，加上來源位址為 2002:A00:1:210::3/64、目的位址為 2001:238:F88::1/64 的 IPv6 header，包成 IPv6 封包；這個 IPv6 封包被送到 6to4 Router B。

②—6to4 Router B 收到 IPv6 封包，把 IPv6 封包用來源位址用自己的 IPv4 位址(10.0.0.1)、目的位址為另一台 6to4 Router(61.218.105.10)的 IPv4 header 包住，將這個 IPv6 封包送往 Internet 上的 6to4 router C。

③—IPv4 封包經過 NAT，NAT 把 IPv4 header 的來源位址改成 NAT 的外部位址(140.113.131.1)，目的位址仍然為 6to4 Router B 的 IPv4 位址(如圖 1 的封包格式)。

④—6to4 Router C 收到帶有 IPv6 封包的 IPv4 封包，6to4 Router C 拆掉這個封包的 IPv4 header 後，把 IPv6 封包送上 IPv6 網路。這個 IPv6 封包被 IPv6 網路送到 D，完成 NAT 後端電腦送資料給 IPv6 網路上電腦的流程。

當 D 要送資料給 A，D 和 A 之間的傳送的流程如下：
⑤—D 把要送的資料用 IPv6 header 包住，IPv6 header 的來源位址填入 2001:238:F88::1/64，目的位址填入 2002:A00:1:210::3/64。因為目的地位址為 6to4 位址(2002::/16)，這個封包會被在 IPv6 網路上的 6to4 Router C 收下轉送。

⑥—6to4 Router C 收到這個 IPv6 封包，把藏在 IPv6 目的位址中的 IPv4 目的位址取出(0x0A000001，也就是 10.0.0.1)，把 IPv6 封包用來源位址 61.218.105.10，目的位址 10.0.0.1 的 IPv4 header 包住。因為這時 6to4 Router C 取出來的目的位址是 6to4 Router B 的 IPv4 位址(private IP 位址)，所以這個封包沒有辦法被送回到 A。換句話說，由於 NAT 的存在，Tunnel 無法被成功地建立。

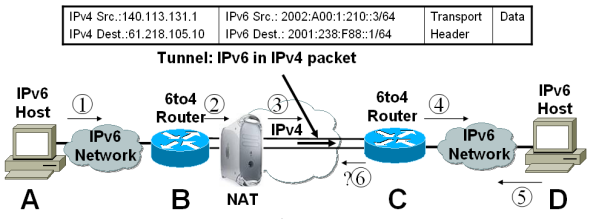


**圖 1 在 NAT 後端的 6to4 Tunnel**

## 3. Teredo Operation Model

在 IETF 的"draft-huitema-v6ops-teredo-00"文件中，提出 Teredo 這個新的 Tunneling IPv6 方法。它可以解決 NAT 後端電腦因為使用 private IPv4 位址所遭遇的 Tunneling 問題，讓在 NAT 後端的電腦可以連上 IPv6 網路。利用 Teredo，提供 NAT 後端電腦和 IPv6 網路相連接的服務，叫做 Teredo 服務。

Teredo 服務的架構圖如圖 2，它需要三個基本元件：Teredo Client、Teredo Server 和 Teredo Relay。下面介紹這三個元件在 Teredo 服務中的功能：

**Teredo Client**：在 NAT 後端，想利用 Tunnel 連上 IPv6 網路的電腦。

**Teredo Server**：有 global IPv6 和 global IPv4 位址，主要功用是幫助 Teredo Client 連上 IPv6 網路，及提供 Teredo Client 該使用的 IPv6 位址。

**Teredo Relay**：有 global IPv6 和 global IPv4 位址，功能像是一個 IPv6 Router，負責傳遞 IPv6 網路和 Teredo Client 之間的 IPv6 封包。

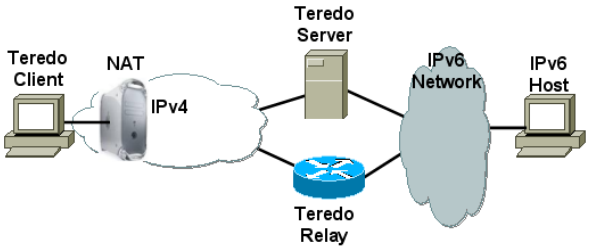


**圖 2 Teredo 服務**

它的基本運作模式是 Teredo Client 和 Teredo Server 溝通後，取得可以穿越 NAT 的 IPv6 位址(Teredo IPv6 位址)，Teredo Client 再和 Teredo Relay 之間建立 Tunnel 傳送 IPv6 封包。

當封包從 NAT 後端穿越 NAT 到外部，這個封包會在 NAT 上留下一個 IP.port 的對應：「封包來源 IP.port」<=>「NAT 外部 IP.port」<=>「封包目的 IP.port」，如果在 Internet 要送封包給 NAT 後端的「封包來源 IP.port」，必須送封包給「NAT 外部 IP.port」，由 NAT 幫忙轉送。所以，封包要穿越 NAT，port 是一定需要的資訊。傳統的 Tunneling 作法，光是用 IPv4 header 包住 IPv6 封包，只有帶有 IP 位址而沒有 port 資訊，無法從 NAT 外面回到內部。

Teredo 服務用的 Tunnel 是 IPv4+UDP Tunnel，這個 Tunnel 方式和傳統做法最大的不同是它帶有 port 資訊，讓 Tunnel 封包有能力穿越 NAT。Teredo 的封包格式如圖 3，封包外層的 IPv4 和 UDP header 讓 NAT 找到 NAT 後端的電腦，Teredo 封包被當作 UDP 的負載傳輸。

| IPv4 Header | UDP Header | Teredo Header | IPv6 packet |
|---|---|---|---|

**圖 3 Teredo 的 tunnel IPv6 封包**

## 4. Teredo Address Encoding

128 bits 的 IPv6 位址[2]足足有 IPv4 位址的四倍長，6to4 IPv6 位址、6over4 IPv6 位址都有利用 IPv6 位址夾帶建立 Tunnel 的資訊，Teredo IPv6 位址除了帶有可以連到 Teredo Client 的 IPv4 位址，還在 IPv6 的 128 bits 裡面塞了其它穿越 NAT 的必要資訊。我們可以說因為嵌入在 Teredo IPv6 位址裡面各式各樣的訊息，讓 Teredo 服務有能力讓 NAT 後端的電腦連上 IPv6 網路。Teredo IPv6 位址編碼方式如圖 4，其中各欄位所紀錄的資訊如下：

**Teredo Prefix**：32 bits，其值為 3FFE:831F::/32，標明這個 IPv6 位址在 Teredo 服務中。

**Teredo Server IPv4 Address**：32 bits，紀錄 Teredo Server 的 IPv4 位址。

**Flags**：16 bits，可以二進位表示為"C00000UG00000000"，當 C 設為"1"時 Teredo Client 判定它所在的 NAT 為 cone NAT，UG 一般都設為"00"。

**Obscured External Port**：16 bits，Teredo client 的 port 對應在 NAT 的外部 port，為隱匿後的值。

**Obscured Client Address**：32 bits，Teredo client 所在電腦的 IPv4 位址對應在 NAT 的外部 IP，為隱匿後的值。

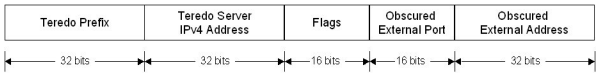


**圖 4 TeredoIPv6 位址 (Source: Microsoft Corp.)**

隱匿(Obscured)的意思，是指該欄位裡記錄的值，是實際值每一個 bit 和 1 作 XOR 的結果；舉例來說，如果 Teredo Client 的 IP.port 對應在 NAT 實際的外部 IP.port 為 140.113.131.2 (0x8C7C8302)及 7890 (0x03D2)，在 Teredo IPv6 位址的 Obscured External IP 及 Obscured External port 欄位會是 0x73837CFD 及 0xFC2D。會這麼做的原因，是因為某些「特別聰明」的 NAT 在封包經過 NAT 往外時，會搜尋 IPv4 的負載，把負載中和封包來源位址相同的 32 bits 置換成對應的 NAT 外部位址；在封包經過 NAT 向內時，則作相反的置換。把 Teredo IPv6 位址中 External IP 和 External port 隱匿起來，可以避免 Teredo 封包的 IPv6 位址被當作 IPv4 位址置換。

Teredo IPv6 位址裡面包含 Teredo Client 在 NAT 的外部 IP、外部 port，這個資訊對 Teredo Relay 非常有用。我們來看看這兩個欄位怎麼被使用。如圖 5，假設 NAT 為 cone NAT[5]，有 IPv6 封包從 IPv6 網路要送給 Teredo Client：

①—因為 IPv6 封包目的位址的 prefix 是 Teredo Prefix，這個封包會送給 Teredo Relay。

②—當 Teredo Relay 收到 IPv6 封包，Teredo Relay 把 Teredo Client 所在 NAT 的 External IP.port 從 Teredo IPv6 位址中取出，用 Teredo 的 Tunnel 方式把 IPv6 封包送到 Teredo Client 在 NAT 上的 External IP.port。

③—NAT 在收到 Tunnel 過來的封包後，根據外部 IP.port 對應到的內部 IP.port，轉送 IPv4 封包給 Teredo Client。Teredo Client 收到封包，把 IPv4 header、UDP header 和 Teredo header 拆掉後，送 IPv6 封包給上層的應用程式。

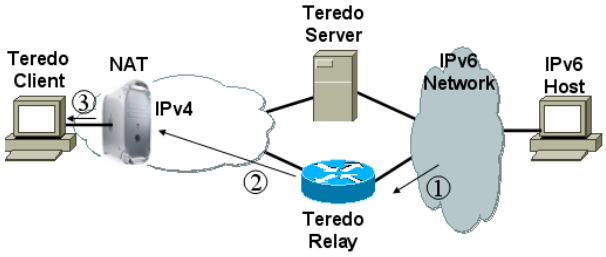


**圖 5 從 IPv6 網路送資料給 Teredo Client**

從上面的例子可以看到，Teredo Service 成功解決了 NAT 後端電腦用 Tunneling 連上 IPv6 網路的問題。(如果 NAT 為 Restricted NAT，可以參考[4]的第四節。)

## 5. Teredo Protocol Dissector and Deployment in NCTU

隨著網路設備的增加，交大計算機中心發放的 IPv4 位址對交大內部許多機構來說已經不敷使用，利用 NAT 上網在交大已經是很普遍的情形。在這種情況下，如果要部署 IPv6 網路到 NAT 後端網

路，是一件困難的事。當然如果可以讓 NAT 懂得 IPv6，變成 IPv6 router，在它後面的設備就可以輕易的透過它連上 IPv6 網路。不過由於目前大部分 NAT 是以嵌入式系統的型式存在，擴充性不高，不像由 FreeBSD、Linux 架成的 NAT 可以隨意增加新功能，要讓現有的網路設備完全升級到支援 IPv6，是非常困難的。這時 Teredo 派上用場了。下面介紹我們計畫在實驗室設置的 Teredo 服務，以及我們目前的成果。

### 5.1 Trial of Teredo in NCTU

我們計畫在實驗室架設 Teredo Server 和 Teredo Relay，如圖 6，Teredo Server 和 Teredo Relay 將會分別以兩台 FreeBSD 架設，並且都連上 HiNet 的 IPv6 網路，以及交大 IPv4 網路。

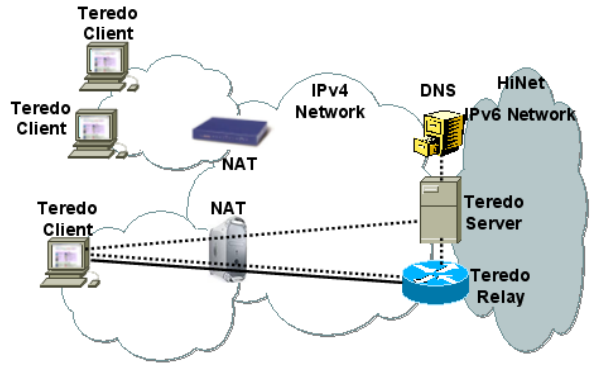


**圖 6 Teredo 在交大的構想圖**

Teredo Client 的軟體部分，Microsoft 已經實作了 Windows XP 的 Teredo Client[7]，只要安裝 Microsoft 的"Microsoft Advanced Networking Package"就可以連上指定的 Teredo 服務。如圖 7，在 Windows XP 安裝 Advanced Networking Package 後，即可從 Teredo Server 取得 Teredo IPv6 位址連上 IPv6 網路。

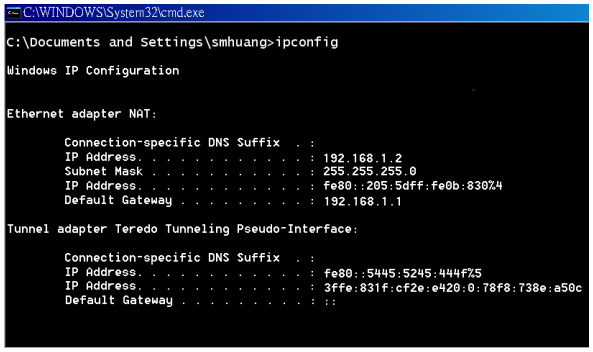


**圖 7 Windows XP 裡面的 Teredo Client**

除了已經實作 Teredo Client 的 Windows 平臺，我們也計畫在 UNIX 平台實作 Teredo Client，讓在 NAT 後端的 UNIX 系統也可以連上 IPv6 網路；另外，我們希望架一台可以讓 IPv4 網路及 IPv6 網路查詢的 DNS，提供 Teredo Client IPv6 位址的 DNS 正反查，DNS 的內容由 Teredo Server 動態更新。

### 5.2 The Teredo Protocol Dissector

從前面的介紹可以知道，Teredo IPv6-in-IPv4 Tunnel 和傳統 IPv6-in-IPv4 Tunnel 的不同，利用現有的網路竊聽器、網路分析器分析封包，只能判斷出 UDP 封包，必須另外分析 UDP 封包裡的 bit 型式，才能知道這個 UDP 封包是不是 Teredo 封包。

為了有效率的分析網路上的 Teredo 封包，我們在 Ethereal 網路分析器[6]上實作 Teredo 封包分析器，增加 Ethereal 分析 Teredo IPv6-in-IPv4 Tunnel 封包的支援，對往後建置 Teredo 服務時的除錯非常有幫助。Ethereal 是一個 Open Source 的網路分析器，它已經被移植在很多平台，也可以在不同平台很輕鬆的編譯出執行檔；最方便的是 Ethereal 提供新增封包分析器的 API，對通訊協定設計者來說，是一個非常有用的軟體。

利用 Ethereal 抓取網路上的封包，加上我們寫的 Teredo 封包分析器，抓到的 Teredo 訊息如圖 8。我們可以清楚的看到，在 IPv4 及 UDP 之上才是 Teredo 訊息，IPv6 封包被 Teredo 訊息包在上層。

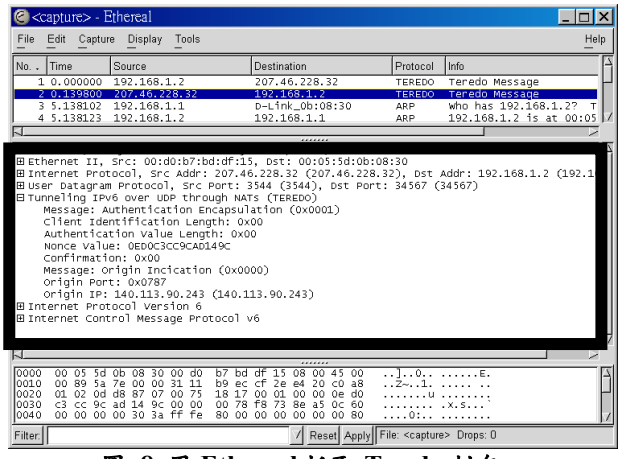


**圖 8 用 Ethereal 抓取 Teredo 封包**

## 6. Conclusion and Future Work

在 NAT 後端，利用 Windows XP 連上 Microsoft 的 Teredo Service，經測試後，Windows XP 可以靠著 Teredo Client 和 IPv6 網路互通。但由於 Microsoft 的 Teredo Relay 目前還沒有對外界開放，因此我們應在國內，特別是 TANet 各區域網路中心，都能佈建 Teredo Relay Router，以提供國內 NAT 的使用者在所有網路設備全面支援 IPv6 的那一天到來之前，也能利用 Teredo 服務連上 IPv6 網路。

在成功架設 Teredo Service 後，值得進一步研究的是下面這些問題：

a) 建立 AAA（Authentication, Authorization, Accounting）機制，只有安裝 NCTU Teredo key 的使用者才能連上 NCTU Teredo Server。

b) 和 STUN protocol 一樣，Teredo 機制可以穿越傳統的 NAT，但遇到 Symmetric NAT 就不通了[5]。如何改進 Teredo 通訊協定，以穿越 Symmetric NAT 甚至是防火牆，是未來在實用上重要的課題。

c) 目前 Teredo tunnel 是用 IPv4+UDP 所建立的，若改用 IPv4+TCP、或 IPv4+SCTP 來建 IPv6 tunnel，效能及可靠度能否有所改善，值得做進一步的深入研究。

## 7. Acknowledgement

感謝 NICI IPv6 R&D 分組計畫支持，以及交通大學及 HiNet 的網路環境支援，使相關實驗得以順利進行。

## References

[1] B. Carpenter, K. Moore, “Connection of IPv6 Domains via IPv4 Clouds”, RFC 3056, February 2001.

[2] S. Deering, R Hinden, “Internet Protocol, Version 6 (IPv6) specifications”, RFC 2460, December 1998.

[3] R. Gilligan and E. Nordmark, “Transition mechanisms for IPv6 Hosts and Routers”, RFC 2893, August 2000.

[4] C. Huitema, “Teredo: Tunneling IPv6 over UDP through NATs”, Internet Draft, draft-huitema-v6ops-teredo-00.txt (Work in Progress), June 2003.

[5] J. Rosenberg, J. Weinberger, C. Huitema and R. Mahy, “STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)”, RFC 3489, March 2003.

[6] http://www.ethereal.com/

[7] http://www.microsoft.com/windowsxp/p2p/