財團法人台灣網路資訊中心

台灣網路資訊中心委託執行計畫

「Native IPv6 連網環境建置及 IPv6 用戶安全防護研析」

受委託單位

國立中央大學

資訊工程學系

Native IPv6 連網試驗環境建置手冊

摘要

鑑於未來 Native IPv6 的發展需求，本計畫將建立 Native IPv6 連網試驗環境及進行 Native IPv6 網路環境測試。本建置手冊詳述如何建置一套 Native IPv6 連網試驗環境，包含 NAT64/DNS64 與 464XLAT 連網架構，以校園網路作為 IPv4 與 IPv6 接取環境，並於實驗室自行架設 NAT64/DNS64 暨 464XLAT 轉換服務，以 JOOL 及 BIND9 開源軟體實現，提供 IPv6 -> IPv4 以及 IPv4 -> IPv6 -> IPv4 兩種連線情境。

本建置手冊將可提供適合未來 Native IPv6 環境的轉移技術，並提供台灣 ISP 業者部署 Native IPv6 之參考，建置成果亦可提供國內業者實際建置 464XLAT 移轉服務之參考。

目錄

圖目錄

表目錄

壹、 前言

本建置手冊將分別針對 NAT64/DNS64 及 464XLAT 連網架構，以校園網路作為 IPv4 與 IPv6 接取環境，並於實驗室自行架設 NAT64/DNS64 暨 464XLAT 轉換服務，提供 IPv6 -> IPv4 以及 IPv4 -> IPv6 -> IPv4 兩種連線情境。本計畫擬採用 JOOL 開源軟體建置 CLAT 及 PLAT 環境；DNS64 伺服器則預計採用 BIND 9 開源軟體建置。以下分別介紹及說明兩者功能：

一、 JOOL

JOOL 作為在 Linux 作業系統中實現 SIIT（Stateless IP / ICMP Translation）、IPv4/IPv6 Translation、Stateful NAT64 及 Stateless NAT64 的位址轉換功能之方案，是最廣泛被使用作為 NAT64 及 464xlat 環境建置的開源軟體之一，由 NIC Mexico 與蒙特雷科技大學開發。JOOL 以 JOOL instance 的方式來實現 IPv4/IPv6 的轉換。JOOL instance 中記錄著 IPv4/IPv6 轉換的規則，使用者可根據需求將客製化的轉發規則添加至 JOOL instance 中。而 JOOL instance 可被附加至兩種不同的 Linux 框架中—Netfilter 與 iptables，藉此可達到訊務攔截（traffic intercepting）的效果，訊務攔截後即透過 JOOL instance 中的轉換規則，將 IPv4/IPv6 的位址進行轉換，再發送至其目的地[1][2][3]。

二、 BIND

BIND 為 Berkeley Internet Name Domain 之縮寫，最初的時候是由加州大學柏克萊分校所發展出來的 BSD UNIX 中的一部份，目前則由網際網路系統協會（Internet Systems Consortium, ISC）來負責維護與發展。BIND 是用來解決網域名稱與 IP 位址對應的軟體，且是個被廣泛使用的 DNS 伺服器軟體，它提供了強大及穩定的名稱服務，因此有近九成的 DNS 伺服器主機都是使用 BIND。目前最新的版本到 BIND 9。BIND 9.8.0 以上的版本透過 dns64 的 options statement 以達到支援 DNS64 的功能。DNS64 通過添加 IPv6 前綴到原 IPv4 位址中以完成 128 位址的 IPv6 位址轉換[4]。

貳、 目的及應用範圍

雖然 IPv6 成長快速，但仍存在一些議題須持續關注，特別是 IPv6 與 IPv4 互通性問題急需解決。未來 IPv6 網路的發展方向將朝向 Native IPv6 方面演進，因此如何在 IPv4 與 IPv6 共存下連線成為一門重要課題。鑑於未來 Native IPv6 的發展需求，本計畫將建立 Native IPv6 連網試驗環境一套。具體如下：

- 464XLAT CLAT(For NAT46 功能)，以 JOOL 開源軟體建置
- 464XLAT PLAT(For NAT64 功能)，以 JOOL 開源軟體建置
- DNS64 Server(搭配 PLAT NAT64 使用)，須用 BIND DNS

Server 軟體建置

本手冊詳述建置 Native IPv6 連網試驗環境步驟，包含軟體安裝、BIND DNS Server 設定及 JOOL 設定，所得成果可進行後續 Native IPv6 網路環境測試，並可提供台灣 ISP 業者部署 Native IPv6 之參考。

參、 NAT64/DNS64 環境建置

NAT64/DNS64 環境建置規劃如圖 1。圖中左方為以 IPv6-only 連接方式連接實驗室網路之用戶，所到達網段或網域僅限於 IPv6 Internet。客戶端如預請求 IPv4 only 的網站服務（如，github.com），亟需透過 DNS64 的幫助。DNS64 架設於學校內網與外部網路之間，在收到來自客戶端有關某個域名 AAAA（IPv6 位址）紀錄的查詢時，它會像平常一般找尋答案，若該域名有 IPv6 的位址，則直接將該紀錄直接回覆給客戶端。相反地，若找不到任何此類的紀錄，即目的地域名不提供 IPv6 位址，則會嘗試查找相同域名的 A 紀錄（IPv4 位址）。DNS64 會將查詢到的 A 記錄中”合成”相等數量的 AAAA 紀錄，將 32 位元的 IPv4 位址嵌入到 128 位元的 IPv6 位址中，並加上 IPv6 的前綴（此處為 64:ff9b::/96）。DNS64 即將此 IPv6 位址回覆給用戶，自此，用戶會認為該域名網站有支援 IPv6 的服務，並且可以直接與其通信。而後，用戶將此域名（新的 IPv6 位址）傳送至 NAT64。NAT64 則通過 JOOL 中之 JOOL instance 將 IPv6 位址中之前綴位址取出，留下後綴位址（原 IPv4 位址），依此後綴即可到達該網站之 IPv4 位址，以存取該服務[5]。

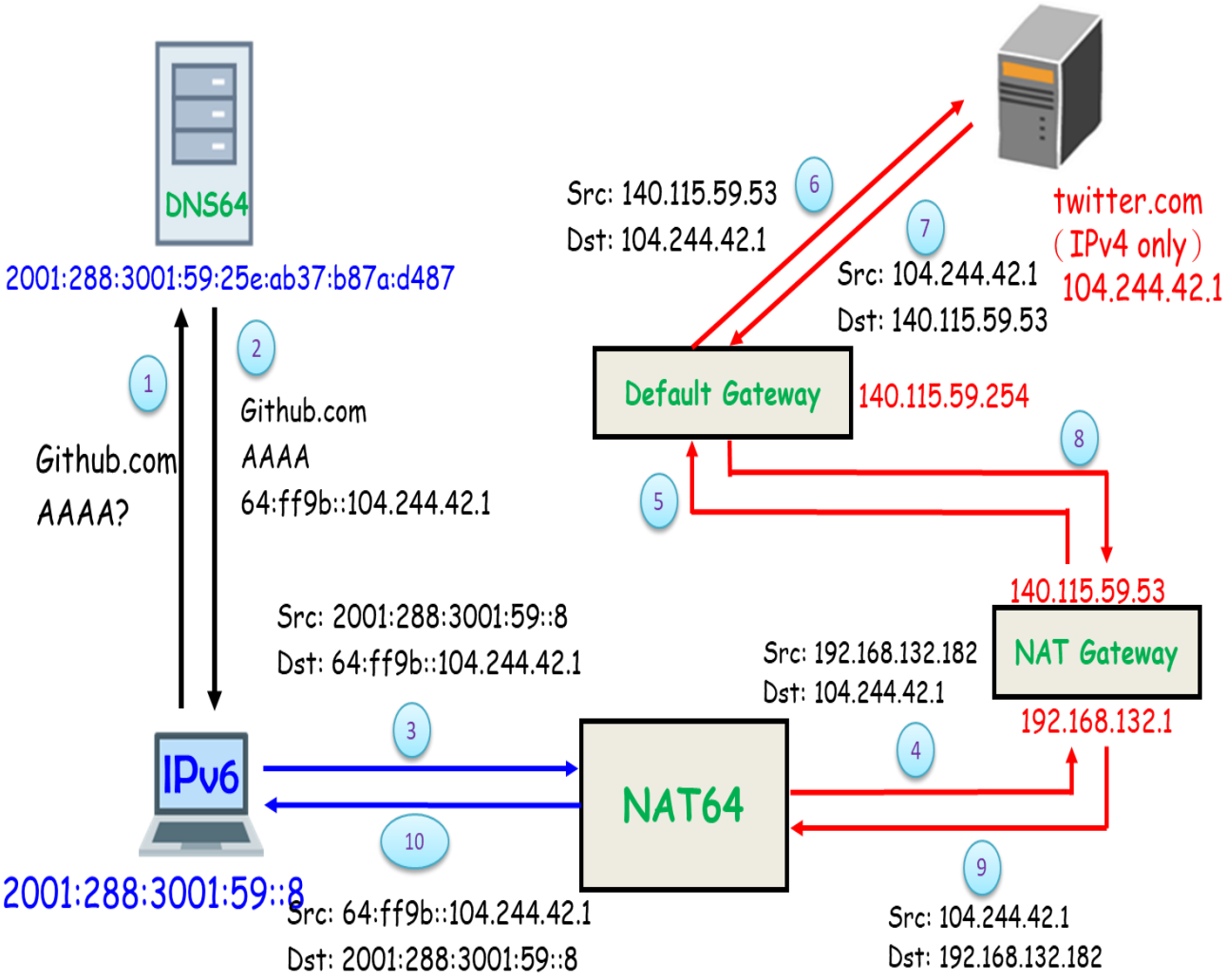

圖 1 NAT64/DNS64 建置

一、 NAT64/DNS64 安裝

NAT64/DNS64 環境將以 JOOL 4.1.1.0 開源軟體搭配 BIND 9.10.3-P4-Ubuntu 建置。NAT64 及 DNS64 皆實作於 Linux based 之虛擬機中，kernel 版本為 5.3.0-62-generic，Distribution 採用 Ubuntu18.04 版本，硬體則配置 2 核心 CPU、4G 記憶體及 60G 的

硬碟容量。詳細軟硬體配置可參考表 1。

表 1 NAT64/DNS64 資源體配置

NAT64/DNS64 資源配置			
		NAT64	**DNS64**
硬體資源配置	**CPU(cores)**	Intel(R) Xeon(R) CPU E5620 @ 2.40GHz 各配置 2 核心	
	Memory(GB)	4	4
	Disk(GB)	60	60
軟體配置	**Distribution**	Ubuntu 18.04	Ubuntu 18.04
	Kernel Version	5.3.0-62-generic	5.3.0-62-generic
	Implement	JOOL	BIND9
	Source Version	4.1.1.0	9.10.3-P4-Ubuntu

JOOL 安裝於 Ubuntu18.04 的執行步驟如下[6]：

- 步驟一：更新系統

```
$  apt update
```

```
$  apt upgrade
$  reboot
```

- 步驟二：安裝 JOOL 必要套件

```
$  apt install build-essential pkg-config
$  apt install linux-headers-$(uname -r)
$  apt install libnl-genl-3-dev
$  apt install iptables-dev
$  apt install dkms
$  apt install git autoconf libtool
$  apt install tar
```

- 步驟三：下載 JOOL 原始碼

```
$  wget   https://jool.mx/download/jool-
   4.1.1.tar.gz
$  tar -xzf jool-4.1.1.tar.gz
```

- 步驟四：安裝與編譯

```
$  /sbin/dkms install jool-4.1.1/
$  cd jool-4.1.1/
$  ./configure
$  make
$  make install
```

- 步驟五：確認安裝

```
$  /sbin/dkms status
   jool,   4.0.1.git.v4.0.1,   4.15.0-54-
   generic, x86_64: built
```

```
jool,   4.0.6.git.v4.0.6,   4.15.0-54-
generic, x86_64: installed
```

若出現以上訊息，即代表 JOOL 安裝成功。接著說明 BIND 安裝於 Ubuntu 18.04 的詳細步驟。BIND9 安裝相較於 JOOL 簡易許多，不須下載額外的相依套件，安裝指令如下[7]：

- 步驟一：下載 BIND9 套件

```
$ apt-get install bind9 bind9-doc
  dnsutils
```

- 步驟二：啟動 BIND9

```
$ service bind9 restart
```

JOOL 及 BIND9 已安裝完成，接著進行 NAT64/DNS64 環境建置的相關設定。

二、 NAT64/DNS64 設定

NAT64/DNS64 測試環境中主要設備為 NAT64 Server 一台及 DNS64 Server 一台。NAT64 Server 所接觸的網段有二，其一為 IPv6-only 網段，用戶即透過此網段連接至 NAT64；另一網段為對外網段，對接到 IPv4-only 的伺服器，因此 NAT64 共需配置兩張網卡，一張為 IPv6-only，另一則為 IPv4-only。此外，在 Linux 中 IPv4 和 IPv6 的轉換允許預設是關閉的，因此須透過 sysctl 指令對 Linux kernel 進行修改，將 NAT64 中 IPv4 和 IPv6 轉換允許設定為 True。JOOL 是以 JOOL instance 的方式來實現 IPv4/IPv6 的轉

換，將轉換規則添加至 JOOL instance 中實現 IPv4/IPv6 的轉發，而後，將此 JOOL instance 與 Linux iptables 掛接，利用 iptables 過濾、篩選封包的特性，達到 NAT64 forwarding IPv4/IPv6 封包之功能。NAT64 與 DNS64 網路配置可參考表 2。

表 2 NAT64/DNS64 網路配置

NAT64/DNS64 網路配置			
	DNS64	**NAT64**	
網卡名稱	ens160	ens160	ens192
網卡位址	192.168.132.x / 2001:288:3001:59::x	2001:288:3001:59::x	192.168.132.x
接觸網段	實驗室內網/外網	實驗室內網 (IPv6-only)	外網 (IPv4-only)
轉發設定	none	net.ipv4.conf.all.forwarding=1 net.ipv6.conf.all.forwarding=1	
JOOL Instance Attatched	none	Iptables	

DNS64 Server 相對於 NAT64 Server 來說相對單純，只需配發一張 dual stack 的網卡，對內連接實驗室內網，對外連接外網與後端 DSN Server。唯須在 BIND9 設定中增加 dns64 的 statement，用以敘述 DNS64 的功能以及配發 Prefix 時之標準。設定如圖 2。

```
options{
    ...
    ...
    listen-on-v6{any;};
    dns64 64:ff9b::/96{
        clients{any;};
        mapped{any;};
        suffix ::;
        recursive-only yes;
        break-dnssec yes;
    };
};
```

圖 2 BIND9 設定

BIND9 設定檔位於/etc/bind/目錄下的 named.conf.options 檔案，DNS64 功能設定說明如下：

$ Listen-on-v6{any;}

- BIND 可接受來自 IPv6 主機的 Query

$ dns64 64:ff9b::/96{};

- 指示 named 在沒有 AAAA 記錄時將映射的 IPv4 地址返回給 AAAA 查詢。旨在與 NAT64 結合使用。每個 dns64 定義一個 DNS64 前綴

$ clients{any;};

- 可接受所有用戶的 request

$ mapped{any;};

- 每個 dns64 支持一個可選的映射 ACL，該 ACL 選擇要在對應的 A RRset 中映射哪些 IPv4 位址，這裡設定為所有 IPv4 位址皆可

$ suffix ::;

- 後綴位址皆保留為 0

$ recursive-only yes;

- 將 dns64 設定為遞迴查詢

$ break-dnssec yes;

- 將 break-dnssec 設置為 yes，即使結果（經過驗證）將導致 DNSSEC 驗證失敗，也會進行 DNS64 合成

NAT64 設定則於 Bash 模式下設定，設定指令可參考圖 3，指令說明如下：

```
$ service network-manager stop
$ /sbin/ip link set ens160 up
$ /sbin/ip link set ens192 up
$ /sbin/sysctl -w net.ipv4.conf.all.forwarding=1
$ /sbin/sysctl -w net.ipv6.conf.all.forwarding=1
$ /sbin/modprobe jool
$ jool instance add "example" --iptables --pool6
  64:ff9b::/96
$ /sbin/ip6tables -t mangle -A PREROUTING -j JOOL
  --instance "example"
$ /sbin/iptables -t mangle -A PREROUTING -j JOOL
  --instance "example"
```

圖 3 JOOL NAT64 設定

$ Service network-manager stop

- 將 NetworkManager 自動化網路管理套件關閉

$ /sbin/ip link set ens160 up

$ /sbin/ip link set ens192 up

- 啟動網卡 ens160 及 ens192

$ /sbin/sysctl -w net.ipv4.conf.all.forwarding=1

$ /sbin/sysctl -w net.ipv6.conf.all.forwarding=1

- 利用 sysctl 將 linux kernel 的 net.ipv4.conf.all.forwarding 及 net.ipv6.conf.all.forwarding 參數設定為 true

$ /sbin/modprobe jool

- 在 linux kernel 中載入 jool module

$ jool instance add "example" --iptables --pool6 64:ff9b::/96

- 名為"example"的 jool instance 掛載到 iptables 中，prefix 為 64:ff9b::/96

$ /sbin/ip6tables -t mangle -A PREROUTING -j JOOL --instance "example"

$ /sbin/iptables -t mangle -A PREROUTING -j JOOL --instance "example"

- Iptables 中的設定，table 為 mangle，mangle 用於特定封包的修改。並搭配 PREROUTING Chain，instance 採用 example。

肆、 464XLAT 環境建置

464XLAT 環境建置規劃如圖 4。464XLAT 為具有私有位址的 IPv4 客戶端通過 IPv6 網絡連接到 IPv4 主機提供了一種簡單且可擴展的技術。我們於實驗室環境中架設一台 CLAT 及一台 PLAT 伺服器，並搭配 JOOL 開源軟體建置。CLAT 通過 JOOL instance 將 IPv4 來源地址和目標地址嵌入 IPv6 前綴中，將 IPv4 位址轉換為 IPv6，然後通過 IPv6 網路將封包發送到 PLAT。 PLAT 將位址轉換為 IPv4，然後通過 IPv4 網路將封包發送到 IPv4 主機。其中若目的地位址有支援 IPv6 服務(如，google.com)，則封包可經過 CLAT 後中途下站，不須再經過 PLAT 將位址轉為 IPv4 送出，可直接由 IPv6 方式存取該服務。而 CLAT 亦可以嵌入於 IPv6-only 的移動網路中的最終用戶移動設備上[8]，從而允許移動網路提供商為其用戶推出 IPv6 服務，並在移動設備上支援 IPv4-only 的應用程式。

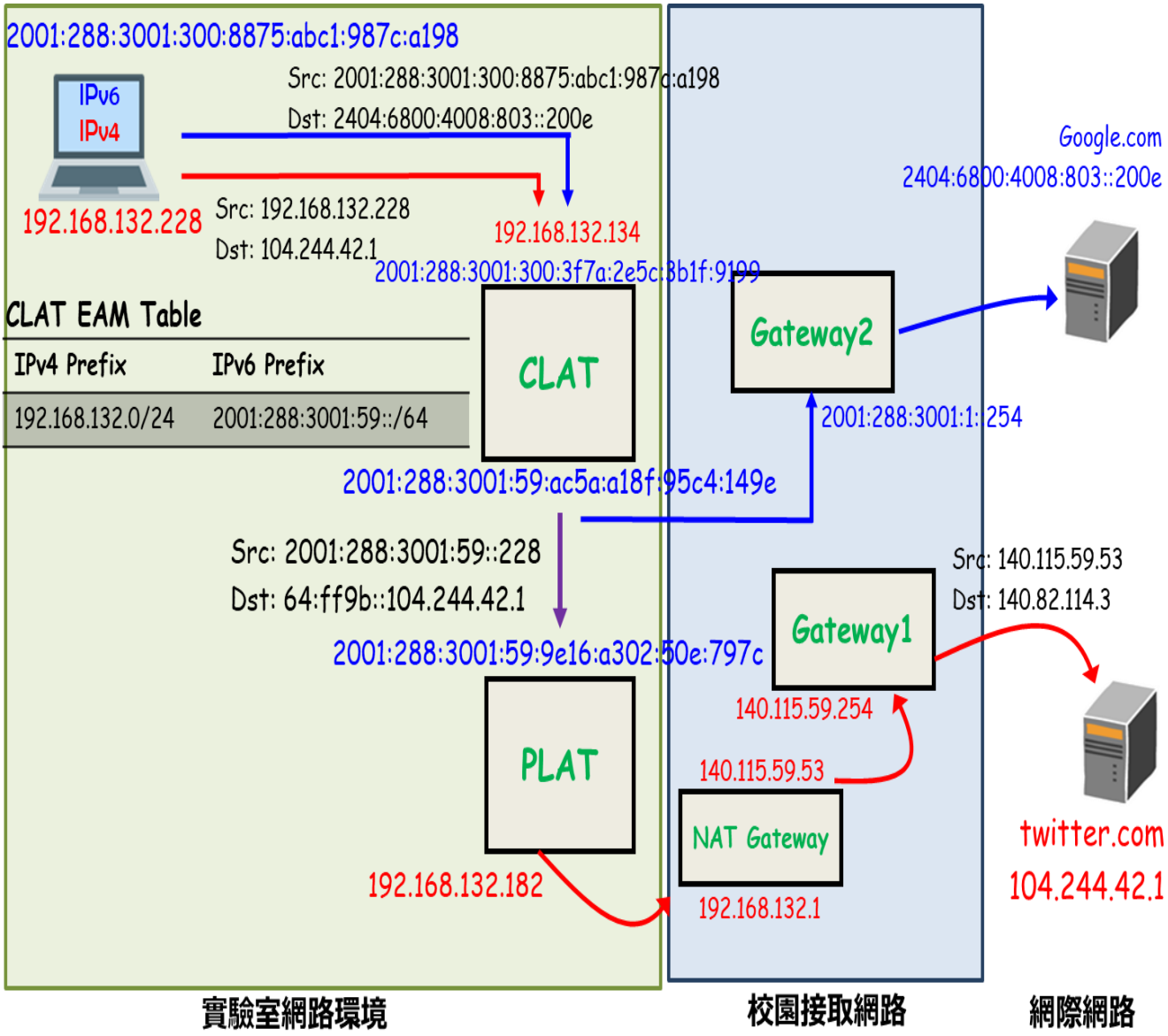

圖 4 464XLAT 建置

一、464XLAT 安裝

464XLAT 環境將以 JOOL 4.1.1.0 開源軟體建置。CLAT 及 PLAT 皆實作於 Linux based 之虛擬機中，kernel 版本為 5.3.0-62-generic，Distribution 採用 Ubuntu18.04 版本，硬體則配置 2 核心

CPU、4G 記憶體及 60G 的硬碟容量。詳細軟硬體配置參考表 3。

表 3 464XLAT 資源配置

464XLAT 軟硬體配置			
		CLAT	**PLAT**
硬體資源配置	**CPU(cores)**	Intel(R) Xeon(R) CPU E5620 @ 2.40GHz 各配置 2 核心	
	Memory(GB)	4	4
	Disk(GB)	60	60
軟體配置	**Distribution**	Ubuntu 18.04	Ubuntu 18.04
	Kernel Version	5.3.0-62-generic	5.3.0-62-generic
	Implement	JOOL	JOOL
	Source Version	4.1.1.0	4.1.1.0

JOOL 安裝如前所述，詳細步驟可參考第參章第一節 ”NAT64/DNS64 安裝”。

二、 464XLAT 設定

464XLAT 測試環境中主要設備為 CLAT Server 一台及 PLAT Server 一台。CLAT Server 所接觸的網段有二，其一為 IPv4/IPv6 共存的 dual stack 網段，IPv4 與 IPv6 用戶即透過此網段連接至 CLAT;另一網段為 IPv6-only 網段,若用戶存取目的地支援 IPv6，即從此網段直接出去，若目的地為 IPv4-only，則將封包 forward 到與 CLAT 對接的 PLAT Server。故 CLAT 共配置兩張網卡，一張為 IPv4/IPv6 dual stack，另一則為 IPv6-only。PLAT Server 亦須配置兩張網卡，一張為 IPv6-only，與 CLAT Server 對接，接收需要經過 IPv6/IPv4 位址轉換之封包；另一張為 IPv4-only，對外連接學校網路 Gateway，以 IPv4 的方式連接到目的地。464XLAT 網路配置可參考表 4。

表 4 464XLAT 網路配置

464XLAT 網路配置				
	CLAT		PLAT	
網卡名稱	ens160	ens192	ens160	ens192
網卡位址	192.168.132.x / 2001:288:3001:300::x	2001:288:3001:59::x	2001:288:3001:59::x	192.168.132.x
接觸網段	實驗室內網	實驗室內網 (IPv6-only) / 外網	實驗室內網 (IPv6-only)	外網 (IPv4-only)
轉發設定	net.ipv4.conf.all.forwarding=1 net.ipv6.conf.all.forwarding=1		net.ipv4.conf.all.forwarding=1 net.ipv6.conf.all.forwarding=1	
JOOL Instance Attatched	netfilter		netfilter	

CLAT 與 PLAT 的 JOOL 設定皆於 Bash 模式下執行。CLAT 設定指令可參考圖 5，PLAT 則參考圖 6。指令接續圖片說明。

```
$ service network-manager stop
$ ip link set ens160 up
$ ip link set ens192 up
$ ip route add 64:ff9b::/96 via
  <plat-ens160-ipv6-address>
$ sysctl -w net.ipv4.conf.all.forwarding=1
$ sysctl -w net.ipv6.conf.all.forwarding=1
$ modprobe jool_siit
$ jool_siit instance add --netfilter \
  --pool6 64:ff9b::/96
$ jool_siit eamt add 192.168.0.0/16 \
  2001:288:3001:59::/64
```

圖 5 CLAT JOOL 設定

$ Service network-manager stop

- 將 NetworkManager 自動化網路管理套件關閉

$ /sbin/ip link set ens160 up

$ /sbin/ip link set ens192 up

- 啟動網卡 ens160 及 ens192

$ ip route add 64:ff9b::/96 via <plat-ens160-ipv6-address>

- 增加路由規則。位址 prefix 為 64:ff9b::/96 者，forward 封包至 PLAT ens160 網卡

$ /sbin/sysctl -w

net.ipv4.conf.all.forwarding=1

$ /sbin/sysctl –w
net.ipv6.conf.all.forwarding=1

- 利用 sysctl 將 linux kernel 的 net.ipv4.conf.all.forwarding 及 net.ipv6.conf.all.forwarding 參數設定為 true

$ Modprobe jool_siit

- 在 linux kernel 中載入 jool module

$ jool_siit instance add --netfilter --pool6 64:ff9b::/96

- jool instance 掛載到 netfilter 中，prefix 為 64:ff9b::/96

$ jool_siit eamt add 192.168.0.0/16 2001:288:3001:59::/64

- Explicit Address Mappings Table（EAMT）是 SIIT 設備中紀錄的集合，用以描述如何轉換不同的位址[9]

```
$ service network-manager stop
$ ip link set ens160 up
$ ip route add 2001:288:3001:59::/64 via
  <clat-ens192-ipv6-address>
$ ip link set ens192 up
$ sysctl -w net.ipv4.conf.all.forwarding=1
$ sysctl -w net.ipv6.conf.all.forwarding=1
$ modprobe jool
$ jool instance add --netfilter --pool6 \
  64:ff9b::/96
$ jool pool4 add <plat-ens192-ipv4-address>
```

圖 6 PLAT JOOL 設定

$ Service network-manager stop

- 將 NetworkManager 自動化網路管理套件關閉

$ ip link set ens160 up

$ ip link set ens192 up

- 啟動網卡 ens160 及 ens192

$ ip route add 2001:288:3001:59::/64 via <clat-ens192-ipv6-address>

- prefix 為 2001:288:3001:59::/64 者從 clat ens192 gateway 出去

$ sysctl –w net.ipv4.conf.all.forwarding=1

$ sysctl –w net.ipv6.conf.all.forwarding=1

- 利用 sysctl 將 linux kernel 的 net.ipv4.conf.all.forwarding 及

net.ipv6.conf.all.forwarding 參數設定為 true

$ Modprobe jool

- 在 linux kernel 中載入 jool module

$ jool instance add --netfilter --pool6 64:ff9b::/96

- jool instance 掛載到 netfilter 中，prefix 為 64:ff9b::/96

$ jool pool4 add <plat-ens192-ipv4-address>

- IPv4 pool 是節點傳輸位址的子集，保留這些位址作為 IPv6 節點的遮罩[10]

伍、 結論

為因應未來 Native IPv6 網際網路環境演進升級的必然趨勢，本建置手冊提供 NAT64 與 464XLAT 以開源軟體建置的方法，並詳述安裝及設定步驟，作為未來 Native IPv6 環境的過渡方案。

NAT64/DNS64 環境以 BIND9 搭配 JOOL 開源軟體實現，。IPv6-only 的用戶可透過 DNS64 獲取到目的地的 IPv6 位址，再透過 NAT64 的幫忙存取到 IPv4-only 的網站或應用程式服務；464XLAT 之 CLAT 與 PLAT 位址轉換功能同樣以 JOOL 開源軟體建置，無論為 IPv4-only、IPv6-only 或 dual-stack 的用戶，皆可透過 464XLAT 的環境存取服務，若存取服務支援 IPv6，則用戶可走 Native IPv6 環境，若存取服務僅支援 IPv4，用戶亦可透過 CLAT 及 PLAT 位址轉換功能，達到存取服務的目的。因此，本手冊成果可作為國內業者實際建置 464XLAT 移轉服務之參考依據。

重要有關文獻

[1] JOOL, https://www.jool.mx/en/index.html

[2] JOOL Wiki, https://zh.wikipedia.org/wiki/NAT64

[3] Github NICMx/Jool, https://github.com/NICMx/Jool

[4] BIND9, https://www.isc.org/bind/

[5] DNS64, https://www.oreilly.com/library/view/dns-and-bind/9781449308025/ch04.html

[6] JOOL Installation, Available: https://www.jool.mx/en/install.html

[7] DNS64 Tutorial, Available: https://nicmx.github.io/Jool/en/dns64.html

[8] 464XLAT Overview, https://www.juniper.net/documentation/en_US/junos/topics/concept/nat-464xlat.html

[9] IETF RFC 7757, Explicit Address Mappings for Stateless IP/ICMP Translation, Available: https://tools.ietf.org/html/rfc7757

[10] Pool4 mode, Available: https://nicmx.github.io/Jool/en/usr-flags-pool4.html