

『 前瞻及完整之 IPv6 生態系推動方案 』

附錄七

『 IPv6 動態 DNS 服務軟體開發 』

109 年委託開發報告

109 年度
「IPv6 動態 DNS 服務軟體開發」
期末報告

計畫委託機關：台灣網路資訊中心

中華民國 109 年 10 月

109 年委託開發報告

109 年度「IPv6 動態 DNS 服務軟體開發」

受委託單位

趣比比股份有限公司

計畫主持人

林韋廷

主要負責人員

張珮馨、馮俊杰、金沅禹、童冠瑜

開發時程：中華民國 109 年 5 月至 109 年 12 月

中華民國 109 年 10 月

目 次

目 次.....	I
表 次.....	III
提 要.....	I
第一章 動態 DNS 服務架構規劃	5
第一節 執行內容說明.....	5
第二節 計畫說明.....	10
第三節 資源運用檢討.....	14
第二章 成果說明	16
第一節 開發 IPv6 DDNS SERVER 端軟體.....	16
第二節 開發 IPv6 DDNS CLIENT 軟體	45
第三節 IPv6 DDNS SERVER 及 CLIENT 軟體安裝及使用	63
第四節 軟體推廣與下載統計.....	65
第三章 結論與建議	73
第一節 結論說明.....	73
第二節 建議事項.....	75
參考資料.....	77
中英專有名詞對照	79

表 次

表 1 IPv6 動態 DNS 服務軟體開發各項工作執行進度.....	10
表 2 開發 IPv6 DDNS CLIENT 軟體各項工作執行進度	10
表 3 開發交付之手冊執行進度.....	11
表 4 各項查核進度表.....	12
表 5 執行人力表.....	14
表 6 合計總經費運用情形統計表.....	15
表 7 BIND 各版本支援時間表	16
表 8 BIND 9 與 BIND 10 比較表	17
表 9 BIND 9 特殊功能.....	20
表 10 CLIENT 開發使用程式語言分析	49
表 11 UWP 跟 WPF 的差別	51
表 12 IPv4 環境測試檢查表	56
表 13 IPv6 環境測試檢查表	59
表 14 DDNS 重新連線的測試檢查	62

圖 次

圖 1 YOUTUBE 上 DDNS 安裝示範影片	8
圖 2 DDNS 運作流程	36
圖 3 在家用環境下架設測試環境	42
圖 4 測試設備	42
圖 5 DDNS CLIENT 與 DDNS SERVER 串接架構	45
圖 6 DDNS IP 位址取得流程	46
圖 7 DDNS CLIENT 更新資料的流程	48
圖 8 測試環境	53
圖 9 測試環境(與 IPv4 測試方式相同)	57
圖 10 測試的網路架構圖	60
圖 11 IPv6 軟體下載推廣	65
圖 12 啟動 DNS 訊息畫面	66
圖 13 新增主機畫面	68
圖 14 主機列表畫面	68
圖 15 新增帳號介面	69
圖 16 帳號管理介面	69
圖 17 金鑰管理介面	69
圖 18 DDNS CLIENT 介面	72

圖 19 設定更新頻率	72
-------------------	----

提 要

關鍵詞： IPv6、DDNS(動態 DNS)

一、 專案描述

隨著 IPv4 的 Public IP 發放完畢，各國政府積極推廣 IPv6。使用 IPv6 可以激發產業將更多的創意與 IPv6 結合，提供更多有價值的服務給用戶。目前幾乎所有的網路服務，都需要透過網域名稱讓用戶存取，包括網站、雲端空間、共享服務等。當服務商提供的服務是建置在動態的 IPv6 位址時，勢必需要使用 DDNS(動態 DNS) 才有辦法讓網站 IP 隨時保持更新。台灣的 IPv6 普及度名列世界前茅，為了增加 IPv6 的可用性，本專案將開發同時支援 IPv4 跟 IPv6 的 DDNS 套件，供使用者下載安裝，建立自己的服務。

市面上常見的類似服務整理說明如下：

表格 1 常見 DDNS 軟體

產品名稱	說明
OpenDNS	不是開源軟體，也不支援自行設定 DNS 服務。使用 OpenDNS 的固定 DNS 為 208.67.222.222 跟 208.67.220.220。不是用來給使用者自行架設動態 DNS 使用。
Godns	只有 Client 端的功能，支援 AliDNS、Cloudflare、Google Domains、DNSPod、HE.net、DuckDNS、DreamHost 等，使用 Go 語言開發，沒有 Server 軟體可以下載。
Docker Ddns	運作在 Docker 上，必須使用 Docker 才可以使用，對於 Docker 不熟悉的使用者，需要先了解 Docker 的用法。
pboehm/ddns	尚未支援 API，需要透過 docker 跟 docker-composer 才能安裝。使用者需要熟悉 docker 的操作。設定方式為 yml，需要懂 yml 才知道如何設定。尚未支援 IPv6。

Vinyldns	使用 Docker 安裝，docker 內建 DNS，由於系統被封裝在 docker 內，要修改 DNS 平台有困難。
-----------------	---

目前(民國 109 年 10 月 27 日)在 Github 上的 DDNS 專案共有 2,918 個，有支援 IPv6 的只有 79 個，而支援 IPv6 且有提供 Server 跟 Client 原始碼並且還在維護中的是 0 個。因此當用戶需要自行架設支援 IPv6 的 DDNS 服務時，如果有原始碼可以自行增添或修改功能，只有本專案開發的這一套。

希望隨著原始碼開放出來，而開發者可以依此基礎上延續開發出更多更強大的功能，不僅可以支援 IPv6 的發展，也可以讓以 IP 為基礎的應用能夠有更多的新服務產生。

二、 本案執行範圍

1. 開發及交付在 CentOS 7.5 以上運作之 IPv6 動態 DNS Server 軟體一套。
2. 開發 IPv6 DDNS Client 軟體。
3. 開發交付之手冊。

三、 計畫管考報告

(一) 期中報告繳交日期，109/06/20(星期六)。

(二) 期中報告繳交日之前繳交 IPv6 動態 DNS 服務架構規劃報

告一份，並將計畫執行成果提列於期中報告內容。

(三) 期末報告繳交日期，109/10/20(星期二)。

(四) 期末報告繳交日之前繳交「IPv6 動態 DNS Server 軟體」、

「IPv6 動態 DNS Client 軟體」各一套(皆含程式原始碼)及

「IPv6 動態 DNS Server 及 Client 軟體開發報告」一份、

「IPv6 動態 DNS Server 軟體安裝建置手冊」跟「IPv6 動態

DNS Client 軟體安裝設定手冊」各一份，並將計畫執行成果

提列於期末報告內容。

(五) 每月 25 日前繳交工作月報。

(六) 視需要召開進度討論會議。

第一章 動態 DNS 服務架構規劃

第一節 執行內容說明

一、 背景分析

動態 DNS(英文簡稱 Dynamic DNS 或稱為 DDNS)是域名系統 (DNS 是 Domain Name System 縮寫)中的一種自動更新域名伺服器 (Name Server)內容的技術。根據網際網路對於域名的規範，每一個域名對應的 IP 位址是固定的。但如果是動態 IP 配置方式，主機的位置可能會常變動，此時，就可以利用一種即時更新機制來更新域名對應的 IP 位址，使得用戶可以透過域名連到 IP 位址所屬的主機，並使用該主機提供的服務。

二、 計畫動機與主題

對於需要架設網站對外提供服務的用戶而言，當網路服務商提供給用戶的 IP 位址是非固定式 IP 位址時，用戶要架設對外服務的網站，就需要使用 DDNS 服務，本計畫將開發一套 IPv6 動態 DNS 服務軟體，以提供有架設網站需求但無固定 IP 者的一個解決方案。

三、 工作架構與施行方法

本計畫開發範圍包括一套支援 IPv6 的 DDNS Server、一套與 DDNS Server 溝通的 DDNS Client 軟體。

目前市場上市佔率最高的 DNS 軟體為 BIND，而 BIND 目前最普及的版本為 BIND 9，雖然維護 BIND 的組織有推出 BIND 10，但在推出後由於維護人力不足，因此 BIND 10 已經處理終止狀態。

目前維護 BIND 的團隊都集中精力在維護 BIND 9，包括各種 bug 的修復、新功能的添加都集中在 BIND 9 上。BIND 10 已經無人維護，在 github 上已經正式宣告將該專案保持在終止狀態，故本計畫以 BIND 9 為主要的 DNS 軟體。

而資料庫在本計畫中負責儲存資料，包括帳號及 IP 位址還有對應的網域名稱資訊，而 MySQL 目前在資料庫領域是市佔率最高的資料庫軟體，不僅使用及維護的族群眾多，技術文件及推廣文件也是最豐富，故資料庫的部分我們選擇使用 MySQL 資料庫。

除了使用 BIND9 及 MySQL 之外，在管理平台的部分，我們使用 Facebook 及其他大型網站使用的 PHP 程式語言來開發管理平台，PHP7 的效能優異，市佔率也是最高，故 DDNS Server 平台以 PHP7 為主要開發的程式語言。

本計畫的 DDNS Server 是建置在 CentOS 作業系統上，CentOS 目前市面上流通的版本有 CentOS-6、CentOS-7 跟 CentOS-8 版本。其

中 CentOS-7 的整體更新支援在 2020 年 Q4 結束，而 CentOS-8 則是到 2024 年的 Q2。由於 CentOS-8 已經很穩定，因此建議安裝者應該安裝新版的 CentOS-8 版本，不要安裝舊的版本，減少日後還要升級的困擾。

本計畫將依據 DDNS 的機制，建置一套可以支援 IPv6 的 DDNS Server，並提供使用者下載自行安裝使用。本計畫也會提供安裝及教學手冊，協助使用者快速上手。手冊會隨著使用者下載 DDNS Server 原始碼一併提供，也可以在 Github 下載。

此外，為了方便使用者安裝時可以有參考依據，我們錄製了一段 DDNS Server 的安裝影片放在 Youtube 上，影片網址為 <https://youtu.be/aTS2Dk3zpQY>，使用者可以在安裝之前先參考 Youtube 上的示範影片。

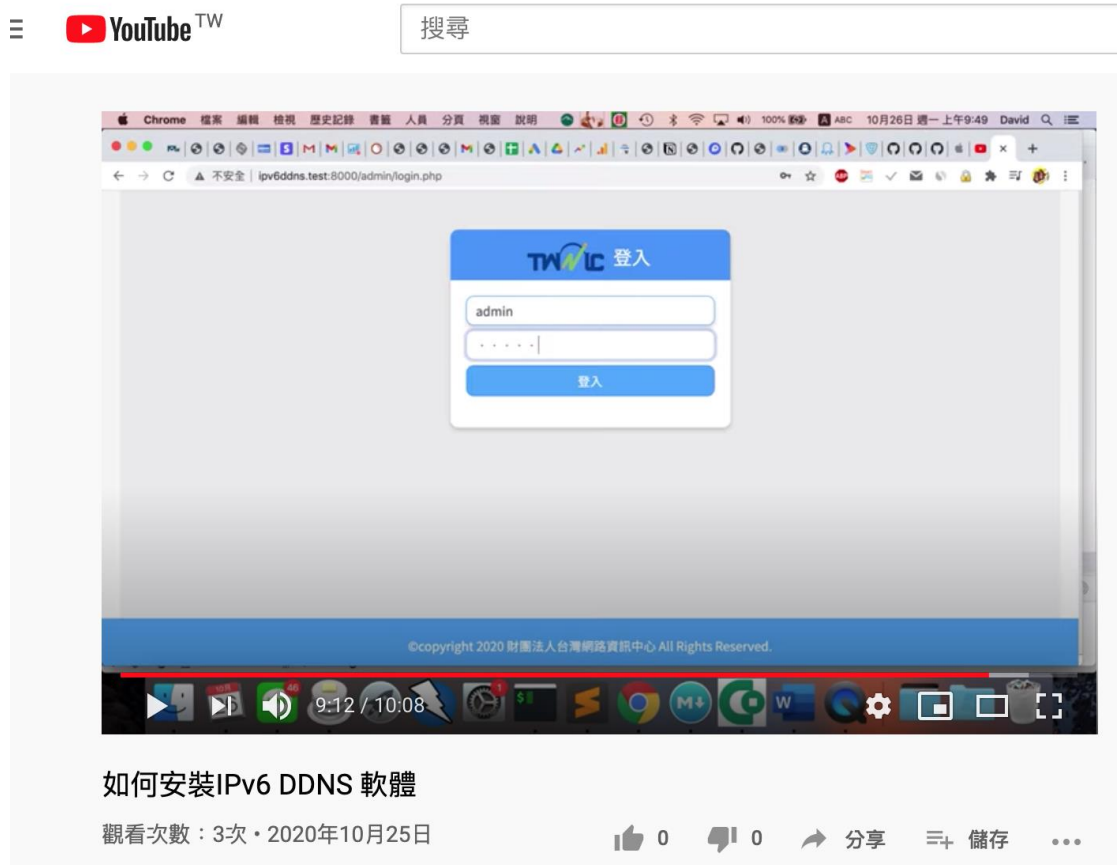


圖 1 Youtube 上 DDNS 安裝示範影片

四、 預期成果

(一) 開發 IPv6 DDNS Server 端軟體

1. 開發及交付在 CentOS 7.5 上運作之 IPv6 動態 DNS Server 端軟體一套(含程式原始碼)
2. 開發 DDNS 設定管理平台
3. DDNS 管理平台介面功能

(二) 開發 IPv6 DDNS Client 軟體：

1. 開發及交付在 Windows 10 平台運作之 IPv6 動態 DNS Client

軟體一套(含程式原始碼)

2. DDNS Client 軟體介面功能

- 帳號及密碼登入
- 顯示目前連線狀態及網域名稱
- 自動更新主機與 IP 的對應資訊
- 設定更新頻率

(三) 交付開發手冊

1. IPv6 動態 DNS Server 及 Client 軟體開發報告一份。
2. IPv6 動態 DNS Server 軟體安裝建置手冊一份。
3. IPv6 動態 DNS Client 軟體安裝設定手冊一份。

第二節 計畫說明

一、 目標達成狀況

依照計畫內容將工作項目涵蓋以下項目：

1. 開發 IPv6 DDNS Server 端軟體。
2. 開發 IPv6 DDNS Client 軟體。
3. 開發交付之手冊。

各項工作執行進度如下表所示：

IPv6 動態 DNS 服務軟體開發各項工作執行進度

表 1 IPv6 動態 DNS 服務軟體開發各項工作執行進度

	工作項目	開始日期	完成日期	進度
開發 IPv6 DDNS Server 端軟體				
1.1	BIND 版本的研究	2020/05/01	2020/05/30	100%
1.2	BIND9 與 MySQL 串接測試	2020/06/01	2020/07/30	100%
1.3	DDNS Server 資料庫結構設定	2020/07/01	2020/07/30	100%
1.4	DDNS Server API 開發	2020/07/01	2020/08/15	100%
1.5	DDNS Server 開發	2020/07/15	2020/10/10	100%
1.6	DDNS Server 測試	2020/09/15	2020/10/10	100%

開發 IPv6 DDNS Client 軟體各項工作執行進度

表 2 開發 IPv6 DDNS Client 軟體各項工作執行進度

	工作項目	開始日期	完成日期	進度
開發 IPv6 DDNS Client 軟體				
2.1	DDNS Client 端運作流程	2020/05/01	2020/05/30	100%
2.2	DDNS Client 連接 DDNS Server	2020/06/01	2020/07/30	100%
2.3	開發 Windows 10 DDNS Client 軟體	2020/07/01	2020/08/30	100%
2.4	測試 DDNS Client 在 IPv4 環境下運作	2020/09/01	2020/10/10	100%
2.5	測試 DDNS Client 在 IPv6 環境下運作	2020/09/01	2020/10/10	100%

開發交付之手冊工作執行進度

表 3 開發交付之手冊執行進度

	工作項目	開始日期	完成日期	進度
開發交付之手冊				
3.1	IPv6 動態 DNS Server 及 DNS Client 軟體開發報告一份	2020/09/01	2020/09/30	100%
3.2	IPv6 動態 DNS Server 軟體安裝建置手冊一份	2020/09/01	2020/10/10	100%
3.3	IPv6 動態 DNS Client 軟體安裝設定手冊一份	2020/09/01	2020/10/10	100%

二、 進度符合情形

各項工作的查核點進度如下表所示：

表 4 各項查核進度表

分類	工作項目	進度超前	進度符合	進度落後	補充說明
開發 IPv6 DDNS Server 端軟體					
1.1	BIND 版本的研究		V		
1.2	BIND9 與 MySQL 串接測試		V		
1.3	DDNS Server 資料庫結構設定		V		
1.4	DDNS Server API 開發		V		
1.5	DDNS Server 開發		V		
1.6	DDNS Server 測試		V		
開發 IPv6 DDNS Client 軟體					
2.1	DDNS Client 端運作流程		V		
2.2	DDNS Client 連接 DDNS Server		V		
2.3	開發 Windows 10 DDNS Client 軟體		V		
2.4	測試 DDNS Client 在 IPv4 環境下運作		V		
2.5	測試 DDNS Client 在		V		

分類	工作項目	進度超前	進度符合	進度落後	補充說明
	IPv6 環境下運作				
開發交付之手冊					
3.1	IPv6 動態 DNS Server 及 DNS Client 軟體開發報告一份		V		
3.2	IPv6 動態 DNS Server 軟體安裝建置手冊一份		V		
3.3	IPv6 動態 DNS Client 軟體安裝設定手冊一份		V		

第三節 資源運用檢討

一、 人力運用情形

本研究計畫共投入總執行人力 7 人，包含專職人員 1 人，及兼職人員 6 人，各人力擔任之工作如下表所示：

表 5 執行人力表

類別	姓名	職位	最高學歷	在本計畫中擔任之工作
主持人 (專任)	林韋廷	技術主管	碩士	負責本專案之統籌、規劃及人力調配與資源配置 DDNS Server 開發工程師
研究助理 (兼任)	張珮馨	企劃	學士	負責規劃及專案控管
研究助理 (兼任)	馮俊杰	工程師	學士	DDNS Server 開發工程師 (負責開發 CentOS 上的平台)
研究助理 (兼任)	金沅禹	工程師	學士	DDNS Client 開發工程師(負責開發 Windows Client)
研究助理 (兼任)	童冠瑜	工程師	學士	QA 工程師(負責功能及壓力測試)
研究助理 (兼任)	黃心怡	工程師	學士	負責 Linux 及 Windows 環境架設
研究助理 (兼任)	蔡采璇	研究助理	學士	負責報告及手冊

二、 經費運用情形

依照目前本計畫進度，依據工作規畫執行各項經費，經費運用情形與進度相當，各項經費使用如下表所示：

表 6 合計總經費運用情形統計表

項 目	預算金額	使用金額	使用率
人事費用	860,000	860,000	100%
業務費	60,000	60,000	100%
合 計	920,000	920,000	100%

第二章 成果說明

第一節 開發 IPv6 DDNS Server 端軟體

一、 BIND 版本的研究

(一)、 BIND 各版本的結束支援時間點(End-of-life，又稱 EOL)如下表。

表 7 BIND 各版本支援時間表

BIND 版本	停止支援時間表
9.11	2022/Q1
9.12	2091/Q2
9.13	2019/Q2
9.14	2020/Q1
9.15	2020/Q1
9.16	2024 之後
9.17	2021/Q2
9.18	2022/Q2
9.19	2022/Q2
9.20	2025 之後
9.21	2023/Q2

1. 關於各版本的 EOL 時間，請參考[1]。
2. 推薦的版本：9.11(最穩定)或 9.16(持續有新功能)，其他版本不推薦。

3. 如果不是使用 9.11 或 9.16 版本，建議升級以確保使用的安全性跟穩定性及可以得到維護單位 ISC(Internet Systems Consortium) 的持續支援。

(二)、 BIND 9 與 BIND10 比較

表 8 BIND 9 與 BIND 10 比較表

差異	BIND 9	BIND 10
維護單位	Internet Software Consortium (ISC)	社群
維護狀態	持續更新	已經終止
特殊功能	支援 DLZ	支援驗證、DDNS、動態管理等

1. 從 2009 年開始，BIND 9 的維護組織 Internet Software Consortium (ISC) 開啟了一個 BIND10 計畫，希望用 BIND 10 來取代 BIND 9，後來 BIND10 重新命名為 Bundy，原因是 ISC 不再繼續支援 BIND10，改由社群支援，因此為了切割，故在 BIND 10 的 1.2.0 版本時，將 BIND 10 改名為 Bundy，而且 Internet Software Consortium (ISC) 不再參與跟維護。

BIND 10 一開始的目標是要設計一個更具有擴充性、更安全的系統，且將更多功能變成模組化，讓 BIND 的功能可以擴充跟增加。

2. BIND 9 在運作時，只有一個服務叫做 named，但是在 Bundy 的規劃中，需要啟動的服務包括 bundy-auth (驗證用)、bundy-cfgmgr(設定管理)、bundy-cmdctl (指令及控制服務)、bundy-ddns (動態 DDNS 更新服務)、bundy-msgq(訊息傳遞服務)、bundy-resolver(解析服務)、bund-sockcreator (網路創建服務)、bund-stats(數據分析及收集服務)、bund-stats-httpd (產生報表的 HTTP Server)、bund-xfrin(傳入的 Zone 傳輸服務)、bund-xfrout (送出的 Zone 傳輸服務)、bund-zonemgr(第二個 zone 管理員)，更多說明可以參考 BIND 9 與 BIND10 比較 [2]。

3. 依據目前放在 github 的 Bundy Repository 記錄，Bundy 版本已經無人在維護，請參考說明文件。[18]

(三)、 BIND 9 的特殊功能。

BIND 9 有許多版本，包括最穩定的 9.11 跟最新版的 9.16 都有支援許多重要的功能，關於這些功能的說明可以參考 BIND 9 特殊功能說明[3]。

表 9 BIND 9 特殊功能

特殊功能	說明
通知	當資料有異動時，讓 Master DNS Server 通知 Slave DNS Server，讓 Slave DNS Server 去抓取新的資料
遞增型的 Zone 傳輸	又稱為 Incremental Zone Transfers (IXFR)，主要用途是讓 Slave DNS Server 只需要請求跟接手有異動的資料
分割的 DNS	為了讓內部 DNS Client 跟外部 DNS Client 看到的 DNS 資訊不同，目的偏向於安全考量
域名系統安全擴充	提供工具用來設定 DNSSEC
動態信任管理	允許 named 能夠一直追蹤重要的 DNSSEC keys 的異動而不需要手動去維護設定檔案
公鑰加密標準支援	支援硬體的加密機制
支援從外部資料庫獲取 Zone 資料	讓 BIND 9 可以支援外部的資料庫，所謂的外部資料庫如 MySQL 跟 PostgreSQL

動態資料庫	允許從外部的資料庫抓取 zone 的資料
Zone 的分類	zones 目錄管理，用來簡化跟加速主要 DNS Server 跟 Slave DNS Server 之間的溝通的傳遞
IPv6 支援	BIND 9 Server 支援 AAAA records 跟 A6 records

1. 通知：此功能是當 Master DNS 資料有異動時，讓 Master DNS Server 通知 Slave DNS Server 去抓取異動的資料，抓取的方式可能是全部抓取或者只要抓取有異動的部分。所謂的異動包括新增、修改及刪除。關於通知機制的說明，可以參考 RFC 1996 的說明[17]。
2. 遞增型的 Zone 傳輸：又稱為 Incremental Zone Transfers (IXFR)，主要用途是讓 Slave DNS Server 只需要請求跟接收有異動的資料，不需要把全部的資料都從 Master DNS Server 傳過來，這樣可以提升 Slave DNS Server 的運作效率，也可以減少 Master DNS Server 跟 Slave DNS Server 之間資料傳遞的傳輸量。關於 IXFR 通訊協定的說明，可以參考文件[16]也就是 RFC 1995 的說明。
3. 分割的 DNS：為了讓內部 DNS Client 跟外部 Client 看到的 DNS 資訊不同，此時就可以使用 Split DNS。使用的時機點在於安全性考量，避免駭客看到內部 DNS Server 設定的主機，也可以便於管理者控管 DNS 主機的查詢，避免將內部的主機洩漏到外部去。關於更多 Split DNS 的用法，可以參考[15]。

4. 域名系統安全擴充：依據 RFC 4033、RFC 4034 跟 RFC 4035，可以實作 DNS Security(又稱為 DNSSEC)。在 BIND 9 版本之後，有提供工具用來設定 DNSSEC。包括利用 `dnssec-keygen` 來產生資料傳遞用的金鑰，用 `dnssec-signzone` 來對一個 zone 簽章，用 `dnssec-validation` 選項來自動驗證資料的正確性等。關於 DNSSEC 的說明，可以參考文件[14]。
5. 動態信任管理：BIND 9 支援 RFC 5011 的金鑰管理，這個功能允許 `named` 能夠一直追蹤重要的 DNSSEC keys 的異動而不需要手動去維護設定檔案。
6. 公鑰加密標準支援：又稱為 PKCS#11 (Public Key Cryptography Standard #11) 支援，主要用途是定義跟平台無關的 API，用來支援硬體的加密機制。BIND 9 目前測試過有支援的硬體加密機制包括 AEP Keyper、Thales nShield 跟 SoftHSM 三種。更多說明請參考[13]。
7. 支援從外部資料庫獲取 Zone 資料：又稱為 Dynamically Loadable Zones (DLZ)本身是一個 BIND 9 的 extension，主要用途是可以讓 BIND 9 可以支援

外部的資料庫，所謂的外部資料庫如 MySQL 跟 PostgreSQL。廣泛來說，可以支援各種外部資料庫，只是需要有人去開發對應的模組，就可以進行串接。為了使用 DLZ 功能，在 BIND 9 要拿原始碼進行重新的編譯(compile)，並且把 DLZ 的支援加進去，編譯方式範例為

```
./configure --prefix=/usr/local/bind  
--with-dlz-mysql=/usr/local/mysql --enable-threads  
--with-openssl
```

做法可以參考文件[12]。在編譯時加進去的話，就變成靜態連結，也就是每次 BIND 9 啟動時，就會支援 DLZ 的功能。使用 DLZ 跟下面要說的 Dynamic Database 相比的優點是，網路上教學文件較多，可以參考別人的教學文件以瞭解如何設定跟安裝 BIND 9 跟 DLZ。

不過 DLZ 有一些缺點，例如 DLZ 的純文字格式要轉換為 named 的格式，這種轉換是很慢的，效能不好。因此如果是要架設一個供超大流量使用的 DNS Server，就不建議用 DLZ，應該改用動態資料庫 (Dynamic Database)。

此外，DLZ 不支援通知的功能，因此如果有架設 Master-Slave 架構的 DNS Server 時，如果 Master DNS 使用 DLZ 時，當 Master DNS zone 資料有更新時，Slave DNS Server 不會接受到通知，因此更新會不同步。

8. 動態資料庫：又稱為 Dynamic Database 或者簡稱為 DynDB，就跟 DLZ(Dynamically Loadable Zones)[11] 相似，允許從外部的資料庫抓取 zone 的資料，這裡所謂的外部資料庫可以像是 MySQL。不過 DynDB 跟 DLZ 不同，DynDB 提供完整的資料庫介面，因此效能上比 DLZ 好很多。DynDB 會把存在資料庫內的所有 zones 資料站存在記憶體內，因此效能跟 native BIND 一樣好，但 DLZ 就不行了，DLZ 的 API 效能差很多，而且 DLZ 有可能無法支援 DNSSEC-signed 的資料。關於動態資料庫的部分，請參考文件[10]。依據參考資料[10]的說明，Dynamic Database 其實很適合用在 DDNS 上，不過在網路上很少有人使用 BIND 9 跟 Dynamic DB，反倒是比較多人使用 BIND 9

搭配 DLZ，BIND 9 使用 Dynamic Database 的教學文件是很少的，也很少人這樣實作。

9. Zone 的分類：又稱為 Catalog Zones[8]，是一種用 zones 目錄管理，用來簡化跟加速 Master DNS Server 跟 Slave Server 之間的溝通的傳遞。當在 Master DNS Server 設定 Catalog Zones 時，這些設定在 Catalog zones 又稱為 member zones，當設定完成之後，只要 Master DNS Server 更新，則 Slave DNS Server 收到之後，如果沒有該記錄，就會自動產生。如果在 Master DNS Server 上的 member zones 更新時(例如新增、修改、刪除)，則 Slave DNS Server 會同步更新，資料傳遞的方式則是透過 AXFR/IXFR zone 傳輸機制。關於 AXFR/IXFR 的傳輸機制，可以參考 ISC 的官方說明[9]。簡單而言，AXRF 是將完整的 zone 資料傳遞過去，而 IXFR 只有傳遞跟上次相比有差異的部分。
10. 對於 IPv6 支援：對於正解的查詢，BIND 9 Server 支援 AAAA records 跟 A6 records，不過在 RFC 6563[4]已經將 A6 records 從 IPv6 移除，雖然 BIND 9 Server 支援 AAAA 跟 A6，但是未來將不再支援 A6 records。

在網域的反解部分，BIND 9 支援傳統的 nibble 格式，所謂的 nibble 格式舉例說明如下，例如 2001:db8::1 轉換為 nibble 格式之後，會變成 1.0.8.b.d.0.1.0.0.2.ip6.arpa.，關於 IPv6 位址與 nibble 格式的轉換方式，可以參考資料[5]。

不過 nibble 格式在 RFC 3363 已經不再支援，因此請不要再繼續使用 nibble 格式。另外，某些舊版的 BIND 9 支援二進位標籤(binary label 或稱為 bitstring)，但因為在 RFC 3363[6] 已經宣告不再使用，而且 BIND 9 內許多軟體也不支援，因此也不要使用 binary label。

二、 BIND 9 與 MySQL 串接

(一)、 BIND 9 支援 DLZ

關於 BIND 9 支援 DLZ 的部分可以在 DLZ 的官網看到更多的詳細資料[19]。在介紹 DLZ 之前，先說明 BIND 的一些限制，這些限制包括：

1. BIND 9 是利用文字檔案儲存 DNS 資料，使用文字檔案儲存資料的缺點是很容易打錯字，而打錯字或者設定錯誤，會造成 BIND 9 Server 無法啟動。
2. BIND 9 將所有的 DNS records 放在記憶體內，因此有時候會發生記憶體不夠用的情況。而且為了讓作業系統可以支援 BIND 9 使用更多的記憶體，你有可能需要重新編譯作業系統核心(Kernel)。
3. BIND 9 會在一開始啟動時，立即讀取跟解析所有 zone 檔案，如果遇到檔案很多時，讀取速度會很久。
4. 如果有修改或者變更 zone files 設定，你需要重啟或者重新載入 BIND 9，這樣很浪費時間。

(二)、 DLZ (Dynamically Loadable Zones)

DLZ (Dynamically Loadable Zones) 是一個 BIND 9 的 patch，目的是可以減少 BIND 的記憶體使用量、加速啟

動時間跟減少錯誤及易於管理。請注意，DLZ 並不是用來取代 BIND 9，而是解決 BIND 9 的一些問題。

另外還有一點是，BIND 9 已經被許多人廣泛的架設跟使用在線上環境，為了降低管理者的負擔，DLZ 一開始開發的方向就是變成一個 patch，然後套用到現有的 BIND 9 原始碼內，這樣就可以讓 BIND 9 使用 DLZ 這項功能，而不會影響到線上使用的功能。

DLZ 一開始在設計時，就希望可以廣泛的支援各種常用的外部資料庫，例如 PostgreSQL、MySQL、Berkeley DB、ODBC (提供給 Firebird、DB2、Oracle、Sybase、SAPDB 使用) 及 LDAP。

(三)、 BIND 9 加上 DLZ 與 MySQL 串接方式

要讓 BIND 9 使用 DLZ 連上 MySQL，你需要下載 MySQL 的 client libraries 跟 headers 等檔案，而 BIND 9 的原始碼在編譯時，需要指定使用 MySQL 並提供 MySQL 編譯用檔案的位址，因此使用的參數如下：

```
--with-dlz-mysql=/path/to/files
```

如果你找不到 MySQL headers 檔案的路徑，可以使用下面指令：

```
mysql_config --cflags
```

如果你找不到 MySQL 的 libraries，可以使用以下指令：

```
mysql_config --libs
```

由於 MySQL 本身支援 thread(執行緒)，但是 BIND 9 本身是單一 thread(執行緒)，因此在啟動 BIND 9 時，請記得加上 "-n 1" 來讓 BIND 9 只會啟動一個 thread。如果不想在每次啟動 BIND 9 時都加上 "-n 1" 這個參數，可以在編譯 BIND 9 時，加上

```
--enable-threads=no
```

把 thread 關閉，這樣 BIND 9 就不會支援 thread，避免後續串連 BIND 9 跟 MySQL 出問題。至於如何設定 BIND 9 來使用 MySQL，可以參考 DLZ 的官方文件[20]。

依據 DLZ 官方的測試，BIND 9 搭配 MySQL 可以達到每秒 689 次查詢，也就是 DLZ 的 MySQL driver 支援每秒 689 次查詢，效能對一般的 DNS Server 需求已經相當足夠。

要讓 BIND 9 透過 DLZ 可以使用 MySQL 時，除了 BIND 9 需要安裝 DLZ 及支援 MySQL driver 之外，還需要在 MySQL 端建立所需要的資料結構，需要開的欄位請參考 DLZ 的官方說明範例[21]。

三、 DDNS Server 資料庫結構設定

1. DDNS Server 的資料庫包括帳號管理、網域管理、系統日誌及 BIND 9 結合 DLZ 使用的資料庫表格，由於數據量不大，因此可以將這些資料庫表格全部集中在同一個資料庫內，各自的資訊分別記錄在各自的表格內。

由於資料庫使用 MySQL，而 MySQL 開始預設的 character set 及 collation 分別為 latin1 及 latin1_swedish_ci，為了確保資訊可以正確被儲存，我們需要在建立資料庫時，將新建的資料庫 character 指定為 utf8mb4，而 collation 則指定為 utf8mb4_unicode_ci，這樣當儲存中文時，才不會出現問題。建立一個 utf8 的資料庫語法如

```
CREATE DATABASE mydb CHARACTER SET utf8  
COLLATE utf8_unicode_ci
```

2. 在 MySQL 資料庫的 collation(校對)有許多選擇，這個值會因為你操作連線到 MySQL 的軟體預設值而有所不同，例如使用 Squel Pro 時，此軟體預設偏好的 collation 為 utf8mb4_0900_ai_ci，而 character 則為 utfmb4。使用 utf8mb4 已經是趨勢，主要是支援多國語言時，可以更精確的儲存資料。

但是在 collation 的部分，如果使用 utf8mb4_0900_ai_ci

時，可能會造成要將資料轉移在之間轉換時出問題。

因此為了保險起見，我們會在 utfmb8_general_ci 及

utf8mb4_unicode_ci 兩者之間擇一。

選擇 utf8mb8_general_ci 的優點則是其速度比

utf8mb4_unicode_ci 稍快，但是在某些特殊字元時，顯示

不如 utf8mb4_unicode_ci 精確，由於資料的正確性對於系

統後續的維護很重要，因此我們優先採用

utf8mb4_unicode_ci 作為 DDNS Server 的 collation 設定。

而 character set 的部分，預設為 utf8 或 utf8mb4，差別在

於 utf8 是用 3 bytes 儲存資料，而 utf8mb4 是用 4 bytes

儲存資料，也就是說，utf8mb4 在儲存資料時，比 utf8 更

精確，因此我們也是選擇使用 utf8mb4。

四、 DDNS Server API 規劃

(一)、 API 的用途

在網路相連的 Client 與 Server 端之間通常會由 Server 端提供介接的介面給 Client 連結，讓 Client 端可以完成包括登入、登出、修改資料、新增資料等行為。API 的全名為應用程式介面(英語：Application Programming Interface，縮寫：API)，透過 API 可以讓不同的部分互相銜接，但需要事先定義銜接的規格，例如使用的通訊協定方式、Port、資料的格式(XML、JSON)及資料交換的欄位等。在網頁開發上，一個 API 通常是指一系列的代碼功能，方便使用者開發第三方的軟體可以跟網站溝通實現互動。關於 API 的更多定義，請參考[22]的說明。

(二)、 DDNS API 設計

1. DDNS 的 API 主要是用來讓 DDNS Client 可以透過 API 跟 DDNS Server 溝通，由於 DDNS Client 為 Windows 軟體，而 DDNS Server 為 Linux 作業系統且是一個網站，因此這種情況下用 API 交換資料可以達成資料交換即時，且可以在規範好的架構下完成資訊的傳遞。在 DDNS API 的設計中，需要的功能包括 (1)

登入與登出 (2)變更 hostname 對應的 IPv4 位址 (3)

變更 hostname 對應的 IPv6 位址。(4)取得目前系統可用的網域資訊。

2. 資料交換的格式包括自訂格式(文字檔或 binary 二進位檔案)、XML 格式、JSON 格式，目前市面上的 DDNS Client 使用的資料交換格式都是 JSON，因此我們也遵循使用 JSON 格式。使用 JSON 格式的優點是支援 UTF-8 且結構方便簡單，各種程式語言都可以在不額外安裝特殊套件的情況下就可以直接閱讀，此外，網路上也有許多線上免費的 JSON Parser 或者 Editor，這也是 JSON 格式會變成資料交換主流格式的原因。
3. JSON 又稱為 JavaScript Object Notation，雖然 JSON 原本是為了 Javascript 而設計，但 JSON 是獨立於程式語言之外的一種文字資料格式，所以 JSON 是與程式語言無關的。關於 JSON 的更多資訊，可以參考 [23]。
4. 資料的交換協定部分，由於 DDNS Server 本身是一台 Server，因此我們使用 HTTP 方式交換資料。HTTP

有包括 HTTP 跟 HTTPS 兩種，而 HTTPS 是在 HTTP 上加上 SSL 加密，由於 SSL 需要有 SSL 憑證，而 SSL 的憑證目前大都需要額外購買，因此如果使用者在有 SSL 憑證的情況下，是可以使用 HTTPS 傳遞資料。如果使用者不願意額外花錢購買 SSL 憑證，則可以使用 HTTP 傳遞資料。

五、 DDNS Server 規劃

(一)、 DDNS Server 架構設計

為簡化 BIND 9、MySQL 與 DDNS Server 之間的資料溝通複雜度，也方便使用者拿到之後可以快速的建置自己的 DDNS Server 服務，運用在自己的網站或者服務主機上，我們讓平台跟 DNS 共用同一個 MySQL 資料庫，但開多個不同的資料表格(Tables)用以存放各自不同的資料內容。BIND 9 使用自己的資料表格存放 zone 資訊，而 DDNS Server 使用自己的資料表格，存放包括帳號、網域資訊、IP 資訊及操作記錄。共用同一個資料庫的好處包括備份容易、資料交換容易、只要一台機器就可以安裝所需套件，節省硬體成本。

DDNS運作流程

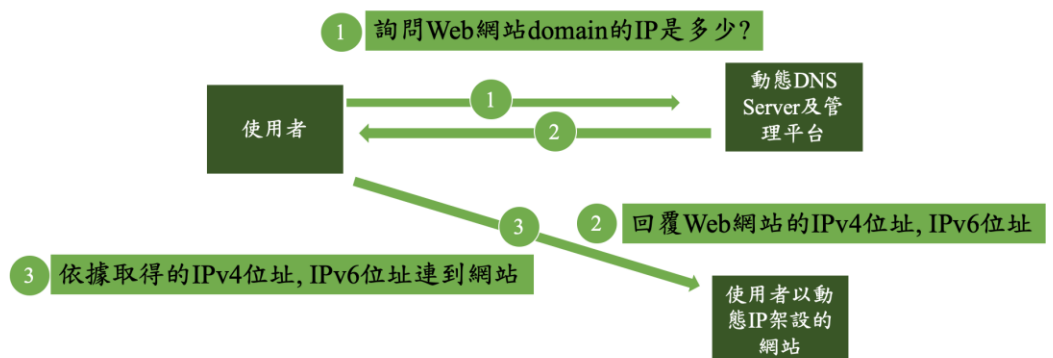


圖 2 DDNS 運作流程

(二)、 資料格式用 XML 與 JSON 的比較

XML 又稱為擴充套件標記語言 (Extensible Markup Language, XML) ，使用 XML 時，可以搭配 XSD(XML schema) 來對 XML 內容進行驗證。使用上是跨平台，格式統一，如果要交換結構化的資料時，使用 XML 是一個好的選項。但 XML 有一個最大的缺點，大部分的程式語言預設都不支援 XML 的解析，因為 XML 的格式是可以由開發系統的人員自訂，因此 XML 的彈性非常大，可以變成很複雜的架構，但越複雜的 XML 的格式，在解讀時會變得非常困難，而且大部分的 XML 解析都需要在程式語言之外再額外安裝套件才有辦法處理。處理 XML 檔案時，必須先了解當初定義 XML 格式的規則，及每一個欄位的用途，不然無法處理。JSON(JavaScript Object Notation)一種輕量級的資料交換格式，相對於 XML 支援完整又複雜的結構，JSON 是一個類似陣列的結構，只是這個陣列可以很多層，不像 XML 需要搭配 XSD 進行驗證，JSON 因為本身是純文字格式，而且是 UTF-8 編碼，因此可以用任何的文書編輯開啟，必要時可以用許多線上免費的 JSON Parser 讀取，使用上非常方便。雖然 JSON

本身是從 Javascript 產生的，不過 JSON 本身也是跨平台且與程式語言無關。整體相比之下，JSON 檔案格式具有佔用頻寬小、格式壓縮、易於讀寫、簡單等特性，更多關於 XML 與 JSON 的比較，可以參考資料[30]。基於上述考量，DDNS Server 端使用 JSON 與 DDNS Client 交換資料，並讓 DDNS Client 透過 JSON 格式的資料進行包括登入/登出、新增/修改/刪除資訊。

(三)、 DDNS Server 的備份機制規劃

DDNS Server 的備份機制：DDNS Server 分為程式碼、資料庫及 BIND 9 設定檔案，檔案的備份可以利用 rsync 或使用 cron job(在 Linux 主機上)定期備份，備份的週期如果需要頻繁，可以設定為每天備份一次，備份的儲存設備可以用外接隨身碟、外接硬碟或 NAS(網路連結儲存設備)，必要時，還可以將檔案備份到光碟片。而程式碼部分，為考量到程式碼異動及維護，建議將程式碼以 git 做版本控制跟備份功能，目前 GitHub[25] 跟 gitlab[26] 都提供私人 git repository 免費使用，因此建議使用者可以把程式碼下載之後，上傳到 github 或者 gitlab。當自己要維護自己的版本(例如新增一些自己的功能，或者修改介

面)，可以用 github 或 gitlab 來維護跟管理程式碼，如果程式碼有故障時，也可以透過程式碼的更新軌跡(github 跟 gitlab 都有提供)，必要時可以回溯到舊的版本。

關於資料庫的備份，由於資料庫是一直在運作中，如果可以停機做備份，則可以使用 mysqldump 進行資料庫的備份，備份出來的檔案再保存到外接硬碟、另一台主機或者燒錄到光片上。如果不能做停機備份時，可以利用 MySQL 現有的 Replication 機制[24]，找第二台電腦安裝 MySQL，利用 MySQL 支援的 Replication 機制，建立 Master-Slave 的架構。如果需要更高的穩定度或者有對外提供重要服務時，可以考慮建置 MySQL Cluster 環境。

硬碟本身也有備份機制，我們稱之為 RAID，利用 RAID 將資料區塊分散儲存在不同的硬碟，當硬碟有壞軌或者故障時，仍可正常運作。

(四)、 DDNS Server 的多國語言支援規劃

由於 DDNS Server 採用 PHP 跟 Smarty 開發，而這兩者本身都已經有支援多國語言的機制。以 PHP 來說，對於需要被翻譯的字串，可以使用 _("字串") 的方式來處理，這個做法是利用 gettext 產生 PO 或者 POT 檔案，最後再以

可以處理 PO 或者 POT 的翻譯系統進行多國語言的翻譯。而 Smarty 需要被翻譯的部分則是頁面 Template 的部分，可以用 `{% trans "字串" %}` 來完成，再透過 PHP 將這些 Template 產生成暫存的 PHP 檔案，再以 gettext 來產生 PO 或 POT 檔案。翻譯的工作可以利用人工翻譯，例如使用 POEdit[27] 或者使用 Crowdin[28] 等雲端翻譯工具處理，或者透過 Google Cloud Translation API[29] 來達成也可以。

六、 DDNS Server 測試規劃

(一)、 DDNS Server 測試範圍

包括網頁功能的測試，例如帳號的新增、修改跟刪除，網域的維護(同樣也是包括新增/修改及刪除等行為)。此外，系統有會將重要的行為記錄到日誌內，例如新增帳號或者修改網域名稱等，這些行為都被系統記錄到資料庫內，作為查詢及稽核使用。此外，也需要測試 DDNS Client 是否可以跟 DDNS Server 溝通，將正確的資訊傳遞給 DDNS Server。而 DDNS Server 除了負責功能的運作之外，也需要跟 BIND 9 搭配 DLZ 共同運作，因此這些都列入在 DDNS Server 的測試範圍內。

(二)、 DDNS Server 的測試過程

要測試 DDNS Server 的功能是否正常運作，需要先使用系統預設的管理者帳號跟密碼登入，登入之後，再逐步測試以下功能：(1)帳號的管理功能 (2)網域的管理功能 (3)與 DDNS Client 串接的運作是否正常 (4)與 BIND 9 的資料更新是否正常。(5)BIND 9 對外的連線是否正常。



圖 3 在家用環境下架設測試環境

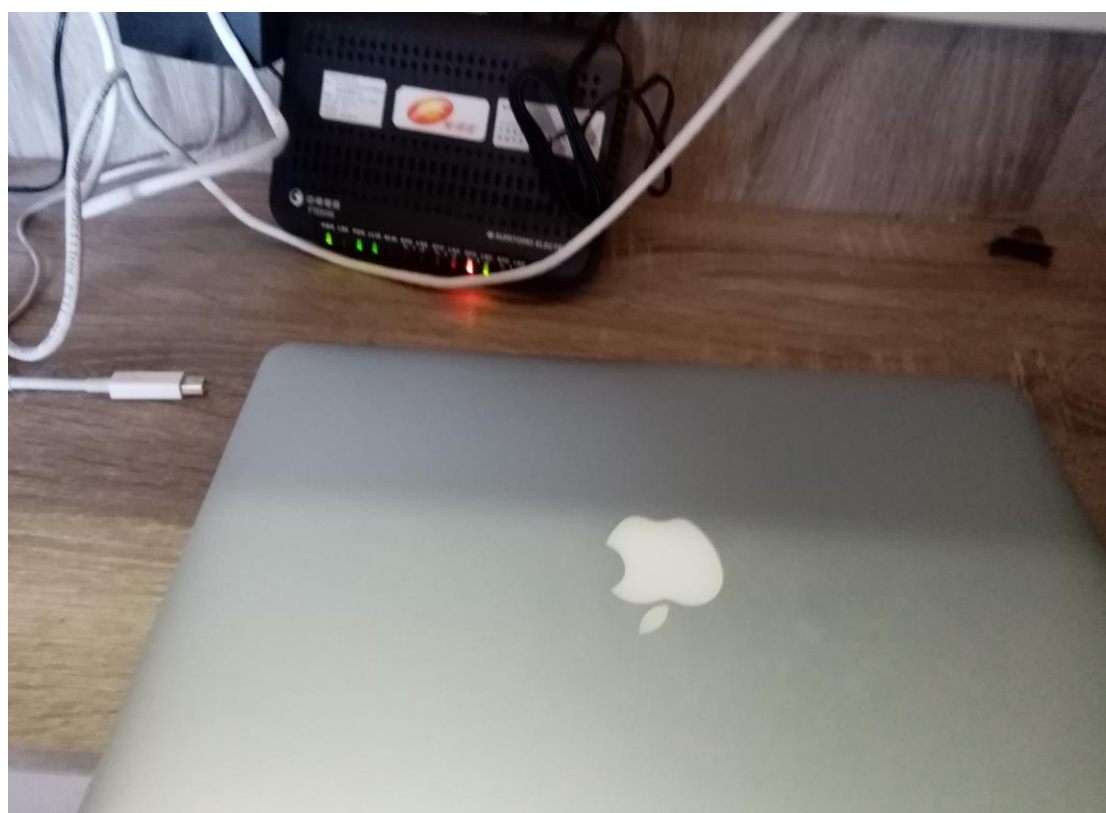


圖 4 測試設備

在進行測試時，務必記得取得 IPv4 不可以是私有 IP，私有的 IPv4 位址有 10.0.0.0/8、172.16.0.0/12、192.168.0.0/16 等網段。在私有 IP 下無法進行測試。

測試過程主要為幾個部分：

測試方式	目的
輸入指令 <code>sudo /usr/local/bind/sbin/named -g -c /usr/local/bind/etc/named.conf</code>	確保 MySQL driver 可以被正確啟用。輸出畫面請參考 IPv6 動態 DNS Server 軟體安裝建置手冊第 62 頁。
<code>/usr/local/bind/sbin/named-checkconf -z /usr/local/bind/etc/named.conf</code>	檢查 DNS 設定檔案是否正常。輸出畫面請參考 IPv6 動態 DNS Server 軟體安裝建置手冊第 50 頁。
<code>dig @localhost www.ddns.idv.tw</code> <code>dig @localhost google.com</code> <code>dig @localhost ns1.ddns.idv.tw</code> <code>dig @localhost ns2.ddns.idv.tw</code>	測試 DDNS 是否有正常運作。輸出畫面

host ddns.idv.tw localhost host ns1.ddns.idv.tw localhost host ns2.ddns.idv.tw localhost	請參考 IPv6 動態 DNS Server 軟體安 裝建置手冊第 72 頁。
設定購買 DNS 的網站上 DNS 的 IP。	讓 DDNS 查詢知道 該向哪台主機查 詢。畫面請參考 IPv6 動態 DNS Server 軟 體安裝建置手冊第 77 頁。
測試是否可以正確取得 Server 端的 資料庫，且與資料庫內的資料一致。	確定 DDNS Client 可以與 DDNS Server 溝通。輸出畫 面請參考 IPv6 動態 DNS Client 軟體安 裝設定手冊第 12 頁。

第二節 開發 IPv6 DDNS Client 軟體

一、 DDNS Client 端運作流程規劃

(一)、 DDNS Client 與 DDNS Server 串接架構

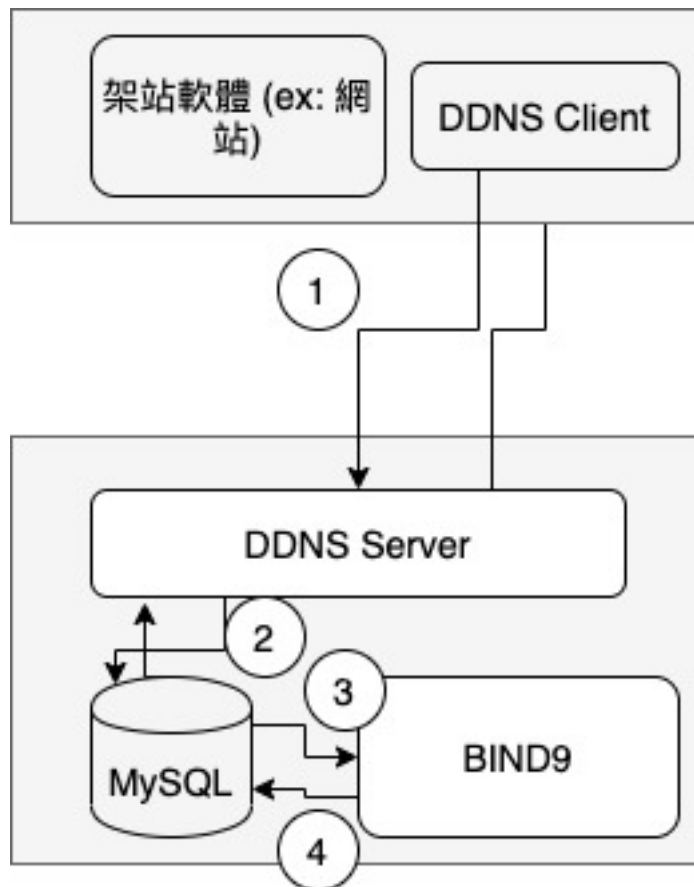


圖 5 DDNS CLIENT 與 DDNS SERVER 串接架構

上圖說明 DDNS Client 與 DDNS Server 串接的一個範例。

流程如下：

1. DDNS Client 取得自己主機上最新的 IP 位址
2. DDNS Client 將自己主機的 IP 位址及對應的網域名稱

傳遞給 DDNS Server

3. DDNS Server 收到 DDNS Client 傳過來的 IP 跟網域的對應資料之後，存入到資料庫。
4. BIND 9 透過資料庫取得最新的 IP 位址與網域的對應關係。

(二)、訪客如何透過 DDNS 連到網站

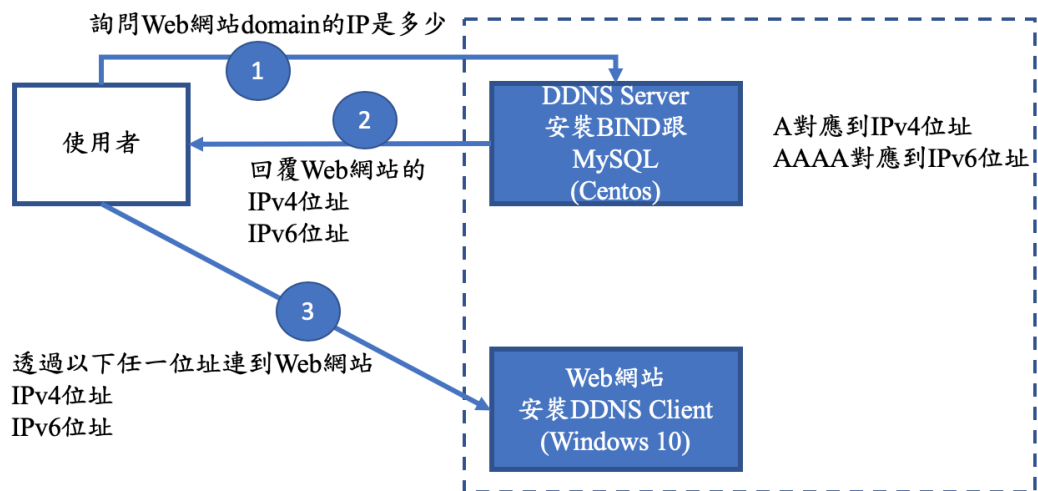


圖 6 DDNS IP 位址取得流程

依據上圖，當訪客要連到一台使用動態 IP 架設的網站時，訪客需要先連到 DDNS Server，取得網站對應的動態 IP，之後再透過 IP 與網站連結，取得網站資料。

二、 DDNS Client 連接 DDNS Server 規劃

(一)、 DDNS Client 端與 DDNS Server 透過 API 驗證流程

圖可以看出 DDNS Client 是如何更新 DDNS Server 上的資料。如果在更新過程中遇到錯誤，DDNS Client 會顯示錯誤訊息給使用者，讓使用者知道發生錯誤，並尋求解決問題的方法。

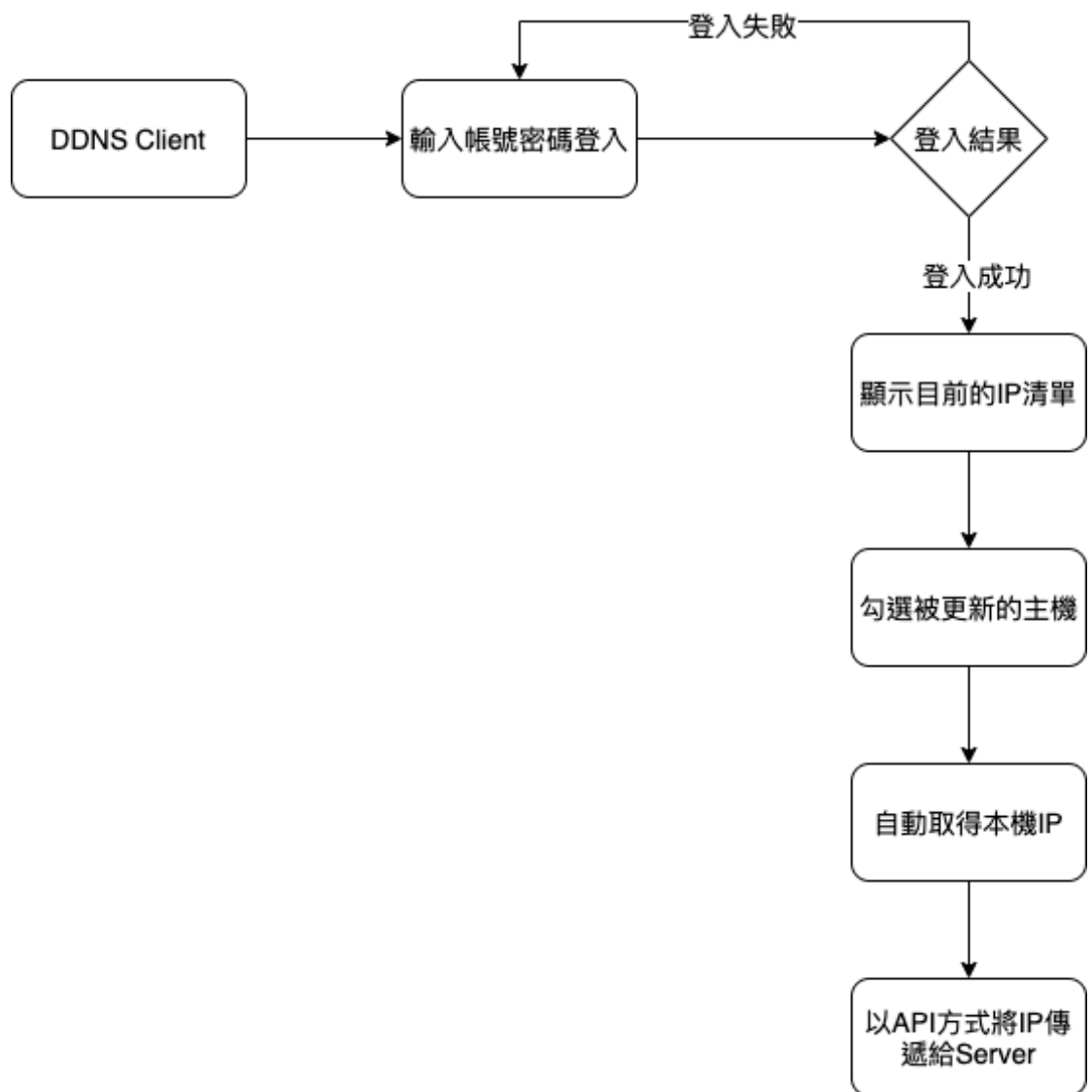


圖 7 DDNS CLIENT 更新資料的流程

三、 DDNS Client 開發使用程式語言的選擇

根據今(109)年 10 月 7 日對 Github 上 DDNS Client 相關專案
進行分析得到以下開發工具的分析：

表 10 Client 開發使用程式語言分析

程式語言	專案數	跨平台
Shell	28	X
Python	21	O
Go	5	△(部分)
PHP	5	O
Javascript	4	O
C	2	X
C#	2	△(部分)
C++	2	X
Java	2	O
Perl	1	O

由上表可以估算出支援跨平台(程式碼寫一次，在不同平台上
編譯後就可以運作)約 50%，因此跨平台開發本身就是一個趨
勢。而本次開發使用的 Electron 也具備跨平台的優勢，以
Electron 開發出的知名軟體包括 Microsoft Studio Code、
Atom(知名編輯器)、Basecamp(知名專案管理軟體)、Github
Desktop(知名 Git 管理工具)、Skype(知名通訊軟體)、Unity(知
名 3D 開發軟體)、WhatsApp(知名通訊軟體)等。

四、 開發 Windows 10 DDNS Client 軟體

(一)、 Windows 10 應用軟體開發注意事項

在 Windows 上開發應用程式時，需要注意 Windows 有應用程式有兩種模式，開發者可以挑自己擅長的方式進行應用程式的開發。目前在 Windows 上的介面框架有兩種，Universal Windows Platform (UWP) 和 Windows Presentation Foundation (WPF)。UWP 對於現有的 Windows 軟體開發者會是較為陌生的框架，主要是因為 UWP 是 Windows 新推出的 UI(介面)框架，以下整理一個表格簡單說明使用 UWP 跟 WPF 的差別。

表 11 UWP 跟 WPF 的差別

	UWP	WPF
架構	Xaml 的 UI 搭配 C#或 VB 的後端 Xaml 的 UI 搭配 C++ 寫的後端 DirectX 的 UI 搭配 C++ 寫的後端 Javascript 和 HTML	Xaml 的 UI 搭配 C#寫的後端 Xaml 的 UI 搭配 VB 寫的後 端 Xaml 的 UI 搭配 F#寫的後端 Xaml 的 UI 搭配 C++寫的後 端
效能	較快	較慢
編譯結果	Native	基於.NET Framework 的代碼
發布時間	2015 年	2006 年
支援 XP	無	有 搭配.NET Framework 4.0 版 本
支援設備	PC 平板 智慧手機 XBOX IoT	PC Windows 平板

	Surface Hub	
觸碰操作	支援	部分支援
畫面顯示 速度	快	慢
Windows 7 支援	無	有
安裝方式	應用商店或者開發者 模式	只要有 .NET Framework 即可
文件	較少	較多

五、 測試 DDNS Client 在 IPv4 環境下運作

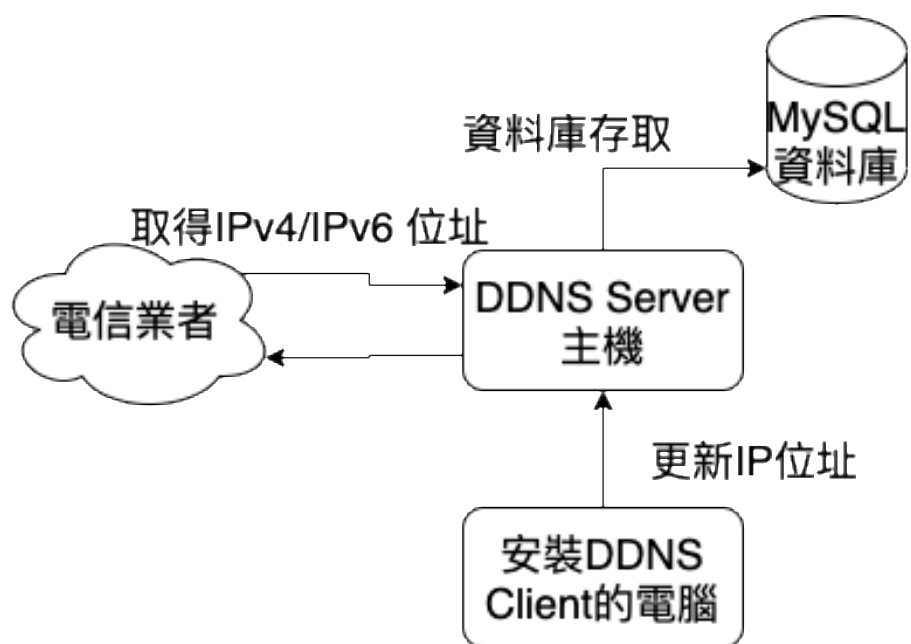


圖 8 測試環境

(一)、 IPv4 環境部署建議

在台灣申請光世代網路時，現在預設都會提供動態 IP，而且同時提供 IPv4 跟 IPv6 位址，因此在測試動態 DDNS 服務時，網路架構圖會是一樣的。

安裝 DDNS Client 的主機或者架設 DDNS Server 的主機都不要透過 NAT(英語: Network Address Translation，縮寫: NAT；又稱網路掩蔽、IP 掩蔽)處理，因為如果透過 NAT 主機，則 IPv4 的 IP 會是 Private IP(私有 IP)無法連到外部網路。

(二)、 IPv4 環境下，DDNS Client 連線測試

要測試 IPv4 的 DDNS Client 是否可以正常運作時，只要在主機上啟動 DDNS Client，透過 Windows 作業系統的命令 ipconfig 取得網路卡的 IP 位址，再比對 DDNS Client 取得的 IP 位址是否相同。

在啟動 DDNS Client 之後，由於 DDNS Client 需要將取得的 IP 位址傳給 DDNS Server，以確保 BIND 9 可以將網域跟 IP 做正確的對應，因此測試時，可以連到 DDNS Server，查看在 DDNS Server 上記錄的網域與 IP 位址的對應表是否跟 DDNS Client 傳過來的 IPv4 位址一樣。

如果不一樣，則需要檢查網路是否有通，或者是否有防火牆設定，或者 DDNS Server 沒有啟動等。以下將檢查項目整理成一張表供使用者評估使用。

表 12 IPv4 環境測試檢查表

編號	項目	檢查
		結果
1	網路連線是否正常？	<input type="checkbox"/> 是 <input type="checkbox"/> 否
2	檢查安裝 DDNS Client 的主機是否是取得 IPv4 位址？	<input type="checkbox"/> 是 <input type="checkbox"/> 否
3	DDNS Client 顯示的網域的 IPv4 位址是否正確？	<input type="checkbox"/> 是 <input type="checkbox"/> 否
4	DDNS Server 是否有設定防火牆，如果有，是否允許 DDNS Client 的 IPv4 位址可以連線(Port 80 跟 443)	<input type="checkbox"/> 是 <input type="checkbox"/> 否
5	DDNS Server 的 Port 53 UDP 是否有在防火牆對外開啟？(Port 53 UDP 是 DNS 查詢服務)	<input type="checkbox"/> 是 <input type="checkbox"/> 否

六、 測試 DDNS Client 在 IPv6 環境下運作

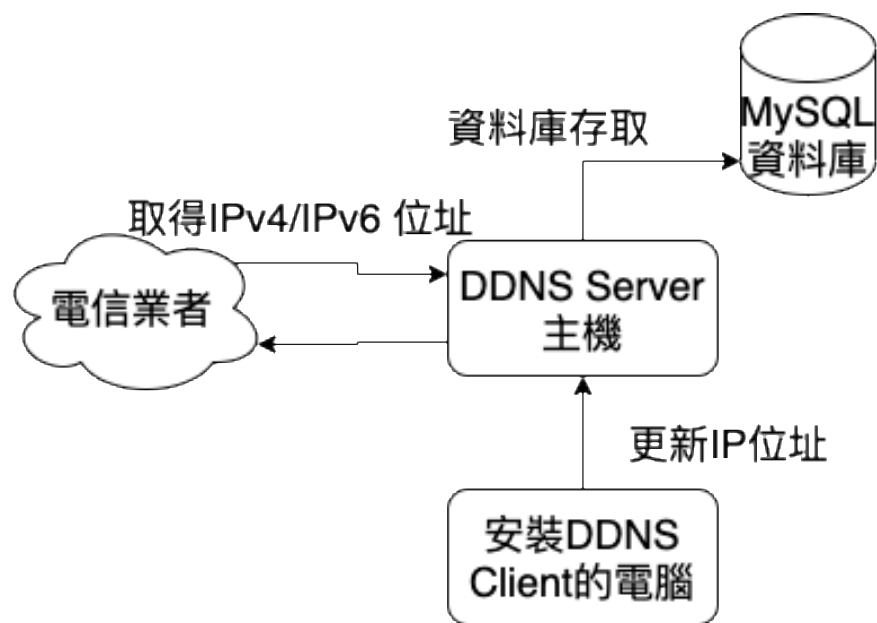


圖 9 測試環境(與 IPv4 測試方式相同)

在正常情況下，如果是直接透過光世代，應該會拿到 2 個 IPv6 位址，在測試 IPv6 DDNS 服務是否正常時，需要安裝的網路環境架構與 IPv4 是相同的，都需要直接接在光世代盒子的後面，不可以接在任何 IP 分享器後面。而主機獲得 IPv6 的方式都是透過 PPPOE，而不要透過 DHCP 方式取得，因為如果是透過 DHCP 取的 IP 位址，表示在該主機的上游有使用 IP 分享器，這種情況下 IPv6 可以測試，但 IPv4 就無法測試 DDNS 服務，因此為了確保測試的條件跟 IPv4 一致，我們建議主機透過 PPPOE 取得 IPv6 位址，這樣也可以避免在找問題時，多了 IP 分享器在上游干擾的問題。讓網路測試環境越單純，可以讓測試方式更精確，也可以減少干擾。

(一)、 IPv6 環境下，DDNS Client 連線測試

如果是申請中華電信的光世代，並使用 PPPOE 取得 IP 位址，則測試方法跟檢查項目可以參考 IPv4 跟 IPv6 的檢查表。

如果不是使用中華電信的光世代網路，如果遇到問題時，除了可以參考本文件做故障排除之外，還可以以電話詢問提供網路服務的業者，請他們的技術人員協助回答問題。

表 13 IPv6 環境測試檢查表

編號	項目	檢查
		結果
1	網路連線是否正常？	<input type="checkbox"/> 是 <input type="checkbox"/> 否
2	DDNS Client 的 IPv6 是否有拿到？	<input type="checkbox"/> 是 <input type="checkbox"/> 否
3	DDNS Server 顯示的網域 IPv6 位址是否正確？	<input type="checkbox"/> 是 <input type="checkbox"/> 否
4	DDNS Server 是否有設定防火牆，如果有，是否允許 DDNS Client 的 IPv6 位址可以連線(Port 80 跟 443)	<input type="checkbox"/> 是 <input type="checkbox"/> 否
5	DDNS Server 的 Port 53 UDP 是否有在防火牆對外開啟？(Port 53 UDP 是 DNS 查詢服務)	<input type="checkbox"/> 是 <input type="checkbox"/> 否

(二)、 IPv4 及 IPv6 並存環境建置

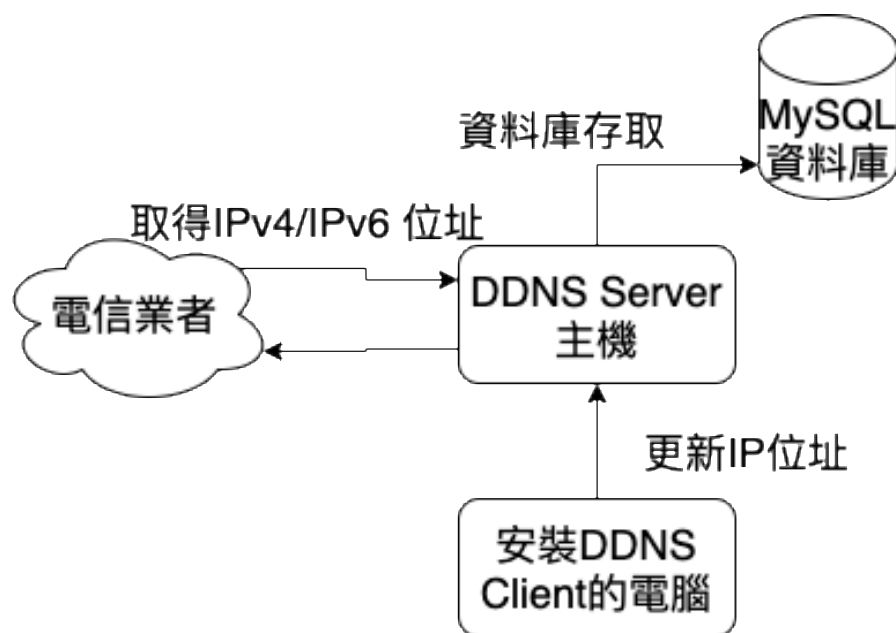


圖 10 測試的網路架構圖

由於本案架設的 DDNS Server 需要同時支援 IPv4 與 IPv6，因此在規劃之初，在測試環境的規劃上，就確保了測試方法的一致性，因此可以看到架構圖是一樣的。

一樣的測試架構優點是足夠清楚且易於尋找問題與進行故障排除，如果在測試環境中又多了 IP 分享器、防火牆、交換機、路由器等，就會讓整個測試的複雜度變高，當發現 DDNS 無法如預期正常運作時，第一個動作應該把不必要的硬體設備都先移除，等故障排除之後，再逐一將實務上要使用的設備一一加入。

(三)、 IPv4 及 IPv6 並存環境下，DDNS Client 連線測試

在台灣因為 IPv6 的普及度夠高，因此測試時，是可以讓 IPv4 跟 IPv6 一起同時進行測試，測試的架構圖已經在上面進行說明。

要測試是否 IPv4 與 IPv6 並存情況下的 DDNS 連線是否正常，可以用下表的檢查清單作為參考依據。

表 14 DDNS 重新連線的測試檢查

編號	項目	檢查 結果
1	DDNS Client 主機是否可以抓到新的動態 IPv4/IPv6 位址？	<input type="checkbox"/> 是 <input type="checkbox"/> 否
2	關閉測試電腦的 IPv6 功能，去連線 DDNS Server 時，是否可以透過 IPv4 正確連線到主機？	<input type="checkbox"/> 是 <input type="checkbox"/> 否
3	關閉測試電腦的 IPv4 功能，去連線 DDNS Server 時，是否可以透過 IPv6 正確連線到主機？	<input type="checkbox"/> 是 <input type="checkbox"/> 否
4	DDNS Server 的 Port 53 UDP 是否有在防火牆對外 開啟？(Port 53 UDP 是 DNS 查詢服務)	<input type="checkbox"/> 是 <input type="checkbox"/> 否

第三節 IPv6 DDNS Server 及 Client 軟體安裝及使用

本計畫完成 IPv6 DDNS Server 及 Client 軟體的開發。

IPv6 DDNS Server 軟體中，包含登入、新增主機、主機列表、修改主機、刪除主機、新增帳號、帳號列表、以及變更密碼等相關功能。IPv6 DDNS Client 軟體中，包含登入、取得 zone 及 host 清單、更新 zone 及 host、以及設定更新頻率等功能。

DDNS Client 負責將本機的 IP 位址更新給特定的主機，讓主機的在 DNS 內的紀錄可以更新。DDNS Client 程式是以 Electron 開發，因此 DDNS Client 同時支援多種作業系統，目前有提供的版本包括 Windows 跟 Mac 版本。如果要支援 Linux 版本，只要取得 DDNS Client 軟體的原始碼，在 Linux 環境下編譯，就可以取得可在 Linux 下執行的 DDNS Client。

DDNS Server 主要是由 PHP、MySQL 開發，並搭配 BIND9 共同運作的一套系統，藉由提供 Web 操作介面，使用者可輕鬆的管理 zone 跟 host。以 Web 介面管理，其優點是自動檢查錯誤，不易發生錯字。利用 BIND9 的 DLZ 跟 MySQL 資料庫，可以做到即時的更新，而且可以減輕管理人員的負擔。另外，使用者也可以基於這套 Web 系統上繼續開發額外的功能，例如監控、稽核、帳務等功能，讓擴展功能變得很容易。DDNS Server 登入系統之後可以新增 zone 跟 host，也可

以管理帳號跟密碼，讓 Client 軟體可以更新指定的 zone 跟 host 的對應 IP 位址(包括 IPv4 跟 IPv6)。詳細功能介紹請參閱：IPv6 動態 DNS Server 及 Client 軟體開發報告。

第四節 軟體推廣與下載統計

軟體目前可以在 <https://ipv6.twnic.tw/promotion-ipv6-ddns.html>

網站上下載，根據統計，目前累計下載次數已經超過 100 次。



The screenshot shows the TWNIC website's page for IPv6 Dynamic DNS software. The header includes the TWNIC logo and navigation links for IPv6, Public IPv6 Upgrade, and Corporate IPv6 Upgrade. The main banner features the text '民眾升級IPv6' (Public Upgrade IPv6) with a blue background and white text. Below the banner, the page title is 'TWNIC IPv6 動態 DNS 軟體' (TWNIC IPv6 Dynamic DNS Software). The content is organized into two main sections: 'IPv6 DDNS Server端軟體' (IPv6 DDNS Server Software) and 'IPv6 DDNS Client端軟體' (IPv6 DDNS Client Software). Each section provides instructions on which operating systems the software is compatible with (CentOS 7.5 for Server, Windows 10 for Client) and includes a '檔案下載' (Download File) button. The footer contains the TWNIC address in Taipei, Taiwan, contact information, and a copyright notice for 2019.

TWNIC 財團法人台灣網路資訊中心
TAIWAN NETWORK INFORMATION CENTER

關於IPv6 民眾升級IPv6 企業升級IPv6 首頁

民眾升級IPv6

首頁 / 民眾升級IPv6 / 一般民眾升級IPv6資訊

TWNIC IPv6 動態 DNS 軟體

IPv6 DDNS Server端軟體

可在CentOS 7.5上運作之IPv6動態DNS Server端軟體：

檔案下載

可在CentOS 7.5上運作之IPv6動態DNS Server端軟體手冊：

檔案下載

IPv6 DDNS Client端軟體

可在Windows10平台運作之IPv6動態DNS Client軟體：

下載連結

可在Windows10平台運作之IPv6動態DNS Client軟體手冊：

下載連結

105臺北市松山區八德路四段123號3樓 | 3F, No. 123, Sec. 4, Bade Rd., Songshan Dist., Taipei City 105, Taiwan (R.O.C.)
總機：+886-2-2528-9696
©copyright
2019財團法人台灣網路資訊中心 All Rights Reserved.

圖 11 IPv6 軟體下載推廣

一、IPv6 動態 DNS Server 軟體安裝及使用

本計畫 IPv6 動態 DNS Server 軟體之安裝，分為在 CentOS8 安裝 DDNS 平台以及在 AWS EC2 安裝 DDNS 平台。安裝的步驟分別為套件安裝，包括 BIND 及 MySQL 套件設定、檢查 BIND 設定、測試 BIND 及 MySQL、以及安裝 DDNS 管理平台。詳細安裝步驟請參閱：IPv6 動態 DNS Server 軟體安裝建置手冊。

[illegible]

圖 12 啟動 DNS 訊息畫面

DDNS Server 軟體安裝後，其功能包括登入、主機管理、密碼設定、及帳號管理等。使用操作包括系統登入、新增主機、主機列表、新增帳號、帳號列表、及變更密碼等。

表 15 DDNS Sever 功能說明

功能	說明
登入	<p>本套 DDNS 系統提供 Web 管理介面，方便使用者藉由 Web 管理介面進行動態 DNS 的相關設定。由於本套系統是開放給使用者下載之後自行安裝，因此只有提供單純的登入介面。當使用者把系統安裝好之後，系統提供一組預設的帳號跟密碼供使用者登入使用。預設的帳號為 admin，預設的密碼為 w123456</p>
主機	<p>當使用要新增主機到網域時，或者同時新增網域跟主機時，就可以在此操作。新增主機時，需要提供 zone(網域)、host(主機名稱)、type(類型)等資訊，這些資訊輸入之後，會直接儲存到與 BIND9 共用的 MySQL 資料庫內。</p>
密碼	<p>這裡的密碼並非是 Client 與 Server 連結的密碼，這裡是指登入 Web 管理介面時，需要使用的密碼。使用者可以視自己的需要進行變更。當然，密碼長度越長，就越安全。</p>
帳號	<p>只有管理者具備新增、修改及刪除帳號之功能。</p>

	一般使用者不能修改帳號。
金鑰	<p>提供除了用帳號跟密碼之外的另一種驗證方式。</p> <p>提供包括新增金鑰及刪除金鑰功能。此功能對於慣於使用命令列(CLI)來串接系統的開發者而言，是很方便的工具。</p>



登出

主機	<ul style="list-style-type: none"> 新增主機 主機列表 	PTR 反解，查詢 IP 所對應的主機名稱 NS 查詢管理領域名稱 (zone) 的伺服器主機名 SOA 查詢管理領域名稱的伺服器管理資訊 A IPv4的位址 AAAA IPv6的位址 CNAME CNAME記錄用於將一個域名（同名）映射到另一個域名（真實名稱），域名解析伺服器遇到CNAME記錄會以映射到的目標重新開始查詢。這對於需要在同一個IP位址上運行多個服務的情況來說非常方便。 TXT 文字記錄，通常存放一些驗證用的資訊。 MX 郵件交換記錄 (MX record)是域名系統（DNS）中的一種資源記錄類型，用於指定負責處理發往收件人域名的郵件伺服器。MX記錄允許設定一個優先次序，當多個郵件伺服器可用時，會根據該值決定投遞郵件的伺服器。簡單郵件傳輸協定（SMTP）會根據MX記錄的值來決定郵件的路由過程。
帳號		
金鑰		
其他		

*Zone

*Host

*Type

☐ A
 ☐ AAAA
 ☐ CNAME
 ☐ TXT
 ☐ MX
 ☐ NS
 ☐ SOA
 ☐ PTR

*Data

圖 13 新增主機畫面



登

主機	<ul style="list-style-type: none"> 新增主機 主機列表 	PTR 反解，查詢 IP 所對應的主機名稱 NS 查詢管理領域名稱 (zone) 的伺服器主機名 SOA 查詢管理領域名稱的伺服器管理資訊 A IPv4的位址 AAAA IPv6的位址 CNAME CNAME記錄用於將一個域名（同名）映射到另一個域名（真實名稱），域名解析伺服器遇到CNAME記錄會以映射到的目標重新開始查詢。這對於需要在同一個IP位址上運行多個服務的情況來說非常方便。 TXT 文字記錄，通常存放一些驗證用的資訊。 MX 郵件交換記錄 (MX record)是域名系統（DNS）中的一種資源記錄類型，用於指定負責處理發往收件人域名的郵件伺服器。MX記錄允許設定一個優先次序，當多個郵件伺服器可用時，會根據該值決定投遞郵件的伺服器。簡單郵件傳輸協定（SMTP）會根據MX記錄的值來決定郵件的路由過程。
帳號		
金鑰		
其他		

Zone	Host	Type	Data	修改	刪除
www	ddns.idv.tw	A	192.168.1.1	修改	刪除
test1.com1	www1	A	192.168.1.12	修改	刪除

圖 14 主機列表畫面



主機 >
帳號 >

新增帳號
帳號列表

金鑰 >
其他 >

*帳號

*密碼

密碼必須為 6 到 20 個數字或字元，至少要包含 1 個數字跟1個字元
*確認密碼

*身份
一般使用者 ▾

送出 取消

圖 15 新增帳號介面

</

圖 16 帳號管理介面



主機 >	<p>以下是您新增的API金鑰的列表，如果您有些金鑰已經不再使用，請記得務必利用右側的刪除功能。</p> <p>另外，為了確保您的平台安全，請不要將這些金鑰張貼在討論區、網站或者臉書跟社團內，避免有心人士濫用您的資源，造成您的損失。</p>						
帳號 >							
金鑰 >	<div>API Key</div> <table> <tr> <td>7273d2921759b87847288d97522e0f81624800b0ba874497db2e0827b62eb72f</td><td>刪除</td></tr> <tr> <td>16b550e0a33dc9665160c48928aea50d7cb689d4955c1994938925c19befe1bd</td><td>刪除</td></tr> <tr> <td>cc2a7f6615dab06c5584c9acea7b54543e82ed5afd5a50f0abdd4b9e79b29d09</td><td>刪除</td></tr> </table>	7273d2921759b87847288d97522e0f81624800b0ba874497db2e0827b62eb72f	刪除	16b550e0a33dc9665160c48928aea50d7cb689d4955c1994938925c19befe1bd	刪除	cc2a7f6615dab06c5584c9acea7b54543e82ed5afd5a50f0abdd4b9e79b29d09	刪除
7273d2921759b87847288d97522e0f81624800b0ba874497db2e0827b62eb72f	刪除						
16b550e0a33dc9665160c48928aea50d7cb689d4955c1994938925c19befe1bd	刪除						
cc2a7f6615dab06c5584c9acea7b54543e82ed5afd5a50f0abdd4b9e79b29d09	刪除						
新增API金鑰							
API金鑰列表							
其他 >							

圖 17 金鑰管理介面

詳細使用說明請參閱：IPv6 動態 DNS Server 軟體使用手冊。

二、 IPv6 動態 DNS Client 軟體安裝設定

本計畫 IPv6 動態 DNS Client 是以 Electron 開發，可以編譯出在 Windows、Mac OS 下運作的軟體。使用者可以依據原始碼自行編譯出在不同作業系統下運作的版本。本計畫已編譯完可支援的作業系統版本包括 Windows 10 以及 Mac OSx 10.14。如果需要了解程式開發細節，請參考 IPv6 動態 DNS Server 及 Client 軟體開發報告。

DDNS Client 軟體下載安裝完成後，由於 DDNS Client 要連到 DDNS Server 需要先變更 name server，將自己架設的 name server 加入到網路設定內，這樣 DDNS Client 在連線時，第一個會先查詢自己架設的 name server，查詢到 IP 之後就直接連線。另外，DDNS Client 軟體的使用操作包括指定 DNS、登入畫面、更新的主機、以及變更 IP 更新頻率等。

DDNS Client 負責將本機的 IP 位址更新給特定的主機，讓主機的在 DNS 內的紀錄可以更新。由於程式是以 Electron 開發，而 Electron 本身支援多種作業系統，因此使用者或開發者可以無需重新修改程式，直接在不同的作業系統上進行編譯，即可使用，此為使用 Electron 開發的最大優勢。詳細使用說明請參閱：IPv6 動態 DNS Client 軟體安裝設定手冊。

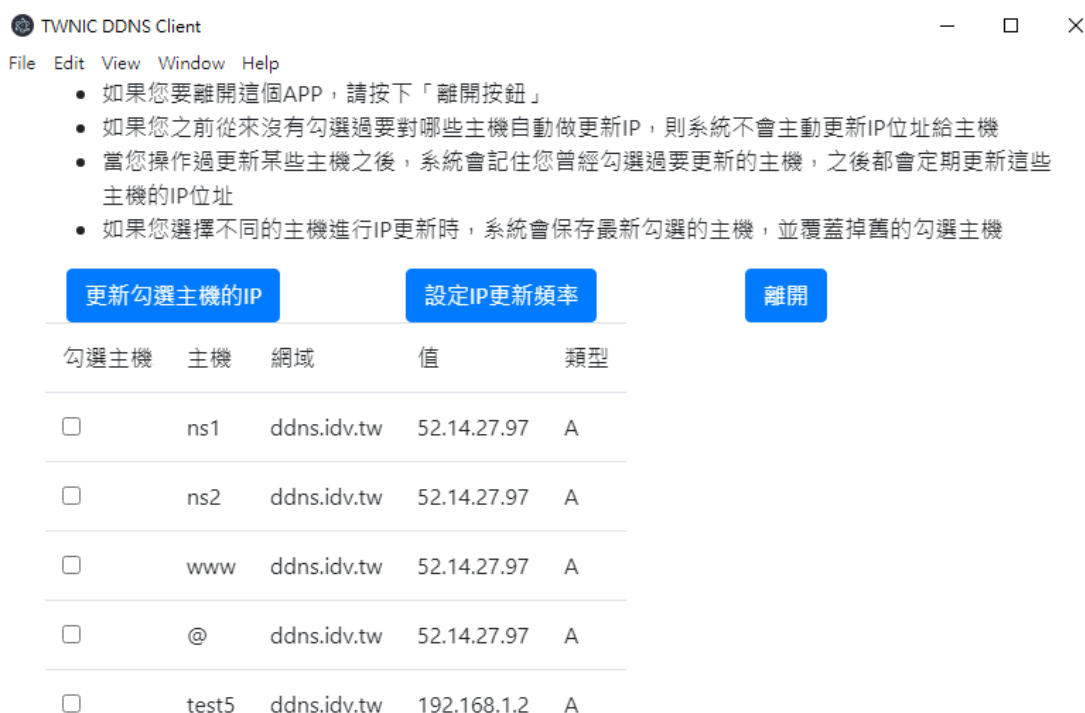


圖 18 DDNS Client 介面



圖 19 設定更新頻率

第三章 結論與建議

第一節 結論說明

一、 開發 IPv6 DDNS Server 端軟體的規劃說明。

在規劃 DDNS Server 端時，主要要注意的地方在於 DLZ 是否可以正常運作，當透過 DDNS Client 將網站的 IP 位址傳遞給 DDNS Server 後，DDNS Server 會將 IP 位址寫入到資料庫內，此時需要測試 DLZ 是否可以在本地端跟遠端進行查詢時，會透過 DLZ 向 MySQL 詢問資料，並回傳正確的 IP 位址。

二、 開發 IPv6 DDNS Client 軟體的規劃說明。

DDNS Client 是透過 API 將資料傳回給 DDNS Server，也是透過 API 向 DDNS Server 取得最新狀態。這部分的規劃因為是透過 API 跟 JSON 格式做交換，在使用上，不容易出問題，只要不是輸入錯誤，這部分都可以正常運作。

三、 DLZ 並不是一個新技術，但簡單好用且不容易犯錯，但不普及。

建議未來相關的專案可以加以推廣。

四、 今(109)年 10 月 7 日在 Github 上搜尋 DDNS 相關的專案共 2,918 個，支援 IPv6 有 79 個(占比約 2.7%)，這 79 個專案大都是依賴現存的 DDNS 服務商所開發出來的 DDNS Client，而到 2020 年

還有在維護且開放 Server 跟 Client 原始碼的專案是 0 個，因此
建議開放 Client 原始碼，以造福更多的使用者。

第二節 建議事項

- 一. 資料庫的安裝建議安裝最新版本，因為本專案使用的資料庫 MySQL 的維護一直都是很積極的狀態，使用最新的版本有可以避免後續有安全漏洞卻無人修復的疑慮。
- 二. 為了降低安裝跟設定的複雜度，DDNS Server 可以跟資料庫裝在同一台，因為 CPU、記憶體跟磁碟空間的使用量都很低，只要一台電腦就已經足夠一般使用者架設網站使用。如果擔心系統中斷服務，建議磁碟做備份，並將主機電源接在 UPS 上，以一般家用需求，這樣的規劃已經很足夠。
- 三. 要增加 DDNS 軟體的使用頻率，建議在 Github 上開放原始碼供全球開發者跟使用者自行下載跟修改，並建議將授權改為 MIT 或 BSD 授權條款。原因是使用 MIT 或 BSD 授權時，使用者可以在商業跟自由軟體中使用，這樣有助於使用者在下載之後進行修改，或者進行商業化銷售。此外，大部分開發者在搜尋軟體時，大部分都會先到 Github.com 網站上搜尋關鍵字，例如 DDNS 或者 IPv6，因此放在 Github 上曝光效果也會較佳，有助於軟體的推廣。
- 四. 軟體目前放在 TWNIC 推廣 IPv6 的官網 <https://ipv6.twnic.tw/promotion.html> 進行推廣，但我們使用

SimilarWeb 網站去分析 ipv6.twNIC.tw 網域時，並沒有得到關於這個網站的流量數據(全部數據為 N/A)，代表這個網站本身的流量是低的。建議可以編列預算對這個網站進行推廣。

五. DDNS 的 Server 端軟體是以 PHP 開發，因此可以架設在任何作業系統上，包括 Windows、Linux 跟 Mac OS。而 DDNS Client 軟體是以 Electron 開發，只要提供原始碼給開發者，開發者可以在完全不需要修改程式碼的情況下，編譯出可以在 Windows、Linux 跟 Mac OS 上運作的軟體。這樣的開發模式(程式碼寫一次，就在多個不同作業系統編譯之後運行)值得在其他軟體開發專案中推廣，這樣可以大幅度減少開發成本，也可以滿足不同作業系統使用者的需求。

參考資料

- [1]. BIND 9 各版本 EOL 時間表 <https://www.isc.org/bind/>
- [2]. BIND 9 與 BIND10 比較
<https://stackoverflow.com/questions/30414796/what-are-the-main-differences-between-BIND-9-and-bund>
- [3]. BIND 9 特殊功能說明 <https://BIND9.readthedocs.io/en/latest/advanced.html#ipv6-support-in-bind-9>
- [4]. 關於 A6 records 說明 <https://tools.ietf.org/html/rfc6563>
- [5]. IPv6 位址轉為 nibble 格式 <http://rdns6.com/nibble>
- [6]. IPv6 不支援 nibble 格式 <https://BIND9.readthedocs.io/en/latest/advanced.html#ipv6-support-in-bind-9>
- [7]. RFC 3363 宣告將 binary lable 設定為不再使用
<https://tools.ietf.org/html/rfc3363.html>
- [8]. Catalog zone 的說明 <https://kb.isc.org/docs/aa-01401>
- [9]. 關於 AXFR/IXFR 的說明 <https://kb.isc.org/docs/axfr-style-ixfr-explained>
- [10]. 關於 Dynamic Database 的支援 <https://BIND9.readthedocs.io/en/latest/advanced.html#dynamic-database-dyndb>
- [11]. 關於 DLZ 的說明 <https://BIND9.readthedocs.io/en/latest/advanced.html#dlz-info>
- [12]. BIND 9 編譯時支援 DLZ 的作法 <http://www.51niux.com/?id=125>
- [13]. BIND 9 支援硬體加密的方式 <https://BIND9.readthedocs.io/en/latest/advanced.html#pkcs-11-cryptoki-support>
- [14]. DNSSEC 的介紹

http://www.cc.ntu.edu.tw/chinese/epaper/0022/20120920_2206.html

[15]. Split DNS 的用法 <http://dns-learning.twnic.net.tw/dns/04FireDNS.html>

[16]. IXFR 通訊協定說明 <https://tools.ietf.org/html/rfc1995.html>

[17]. DNS 的通知機制 <https://tools.ietf.org/html/rfc1996.html>

[18]. Bundy 已經無人維護 <https://github.com/bundy-dns/bundy>

[19]. DLZ 官網介紹 <http://bind-dlz.sourceforge.net/>

[20]. BIND 9 支援 MySQL 設定

http://bind-dlz.sourceforge.net/mysql_driver.html

[21]. MySQL 新增 DLZ 使用的資料庫表格

http://bind-dlz.sourceforge.net/mysql_example.html

[22]. API 的範例說明 <https://developer.mozilla.org/zh-TW/docs/Glossary/API>

[23]. JSON 格式說明 <https://zh.wikipedia.org/wiki/JSON>

[24]. MySQL Replication 設定說明

<https://dev.mysql.com/doc/refman/8.0/en/replication.html>

[25]. GitHub <https://github.com/>

[26]. gitlab <http://gitlab.com/>

[27]. POEdit <https://poedit.net/>

[28]. Crowdin <https://crowdin.com/>

[29]. Google Cloud Translation <https://cloud.google.com/translate>

[30]. XML 與 JSON 比較 <https://hackr.io/blog/json-vs-xml>

中英專有名詞對照

A	
ACL 存取控制列表(Access Control List)	ICMPv6 網際網路控制訊息協定第六版(Internet Control Message Protocol Version 6)
D	
DNS 網域域名伺服器(Domain Name System)	ICP 網路內容供應商(Internet Content Provider)
DHCP 動態主機組態協定(Dynamic Host Configuration Protocol)	IETF 網際網路工程任務小組(Internet Engineering Task Force)
E	
Email 電子郵件(Electronic mail)	IPv4 網際網路通訊協定第四版(Internet Protocol version 4)
F	
Firewall 防火牆	IPv6 網際網路通訊協定第六版(Internet Protocol version 6)
FTP 檔案傳輸協定(File Transfer Protocol)	ISP 網際網路服務提供者(Internet Service Provider)
I	
	IASP 網際網路服務提供者(Internet Access Service Provider)
	IT 資訊技術(Information Technology)