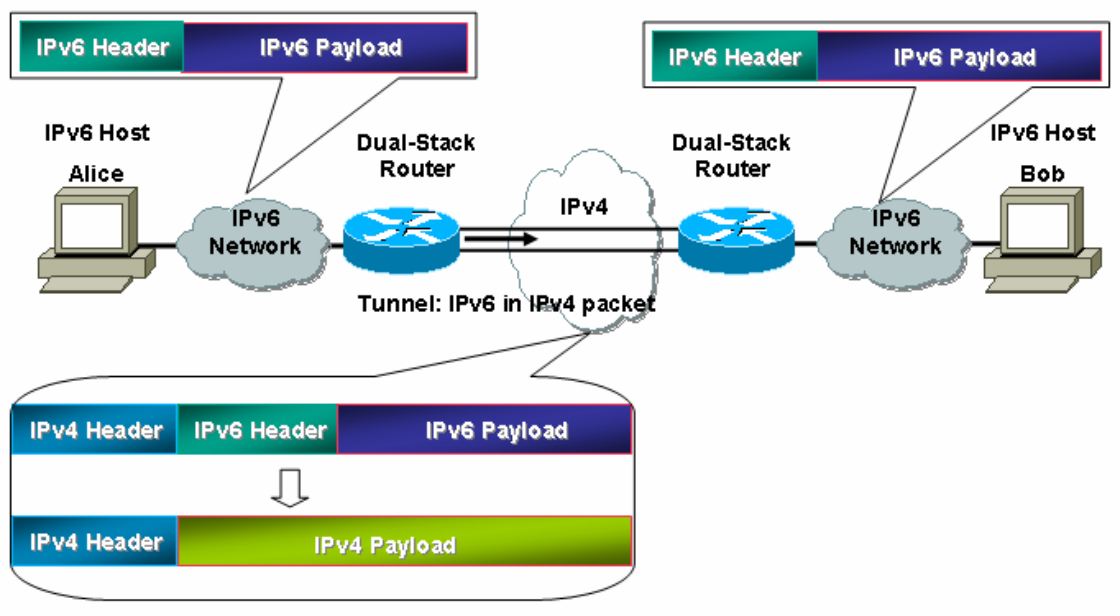
第三章　IPv6 為基礎之中介軟體通訊管理模式

3.1 IPv6 Tunnel

由前面文獻探討可以了解，目前正逢 IPv4 與 IPv6 轉換期，在 IPv4 轉換為 IPv6 的過程中，會有一段時間是 IPv4 與 IPv6 共存的環境。目前若要在沒有完整的 IPv6 網路環境下使用 IPv6 的通訊協定與其他 IPv6 主機溝通時，狀況如【圖十二】的運作模式，通訊雙方使用 IPv6 網路協定，而連結網路只提供 IPv4，因此透過中間 Dual-Stack Router 來做類似代理的工作。在左邊的 IPv6 Network 中，左邊 Alice 欲使用 IPv6 封包與右邊 Bob 溝通時，Alice 只需照原本 IPv6 的方式來發送封包，並不用管中間有一段網路環境為 IPv4，左邊的 Dual-Stack Router 會幫 Alice 將封包加上 IPv4 的 Header，將原本 IPv6 的封包當成新 IPv4 封包的 Payload，透過 IPv4 網路送往右邊的 Dual-Stack Router，右邊 Dual-Stack Router 收到後，再將封包中 IPv4 的表頭去掉，復原為原本 Alice 發出的封包。因此 Alice 與 Bob 可以照原本 IPv6 網路通訊協定作溝通，完全不用管中間的網路環境，Dual-Stack Router 會幫他們處理好。



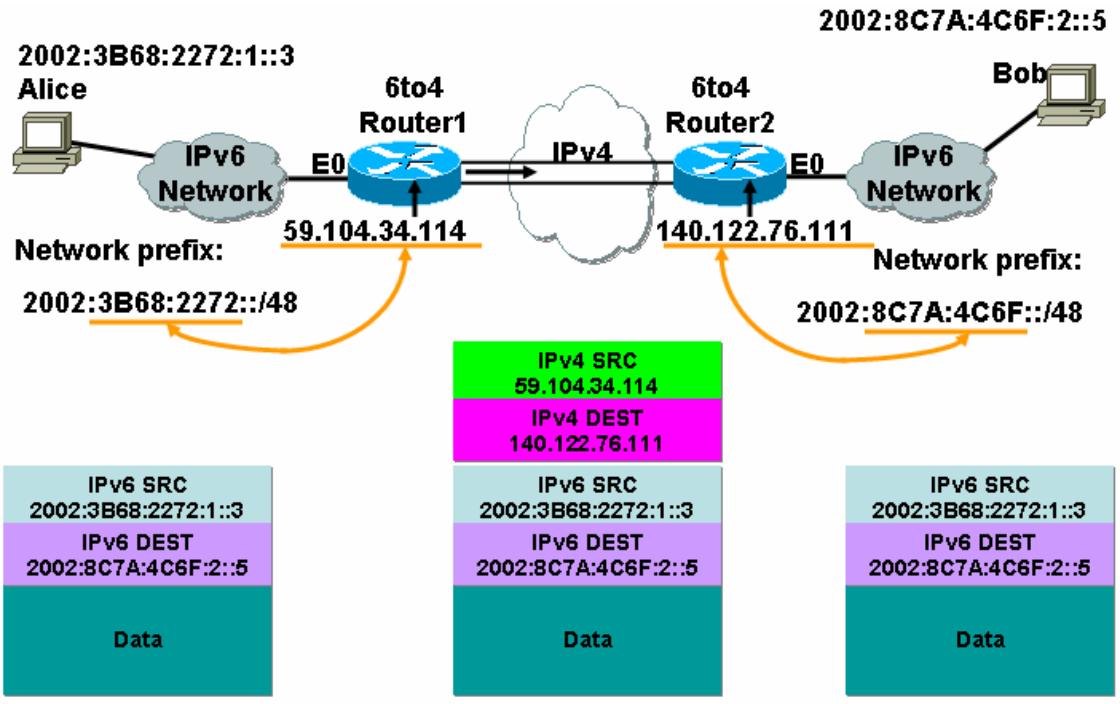
【圖十二】 IPv6 Tunnel 示意圖

IPv6 tunneling 目前最常用的為 6to4 Tunnel 與手動設定之 IPv6 Tunnel 這兩種，以下就這兩種一一說明，最後是筆者提出安全性較高，可做存取控制的 Tunnel 設立方式。

6to4 Tunnel

6to4 Tunnel[24]運作的方式大致如下【圖十三】，其網路位址為 2002::/16 開頭，接著的 32bit(第 17bit~48bit)則將該主機 IPv4 位址使用 16 進位方式表示，例如左邊 Router1 之 IPv4 位址為 59.104.34.114，59 與 104 以 2 進位表示為 00111011 01101000，此 16bit 改為 16 進位則可以 3B68 表示，另外，34 與 114 以 2 進位表示為 00100010 01110010，此 16bit 以 16 進位表示為 2272，因此 Network prefix 即為 2002:3B68:2272::/48。6to4 Tunnel 為一種自動建立 Tunnel 的機制，假設 Alice 的 IPv6 address 為 2002:3B68:2272:1::3，想與 Bob(2002:8C7A:4C6F:2::5)溝通時，Alice 送出 IPv6 封包，目的位址填寫 Bob 的 IPv6 address，Router1 收到封包後，

發現是要傳給 2002:8C7A:4C6F::/48 的封包，則由目的 IP 位址的 8C7A 與 4C6F 可以算出 v4 位址為 140.122.76.111。接著使用 Tunnel 技術，將封包自動加上 IPv4 的來源與對方目的，因此封包可以經由 IPv4 網路送到 Router2，Router2 收到後將 IPv4 的表頭拿掉，變為 IPv6 的封包送給 Bob，因此 Bob 收到的封包與 Alice 發出的完全一樣。Bob 回覆時，一樣使用 IPv6 的通訊協定來傳送封包，接著 Router2 使用相同的方式，使封包到達 Alice 那邊。使用這個方法可以達到自動建立 Tunnel，不過因 Router1 與 Router2 使用 IPv4 網路傳遞資料，Router1 與 Router2 都必須使用公開的 IPv4 位址才可以正常使用這種方式。

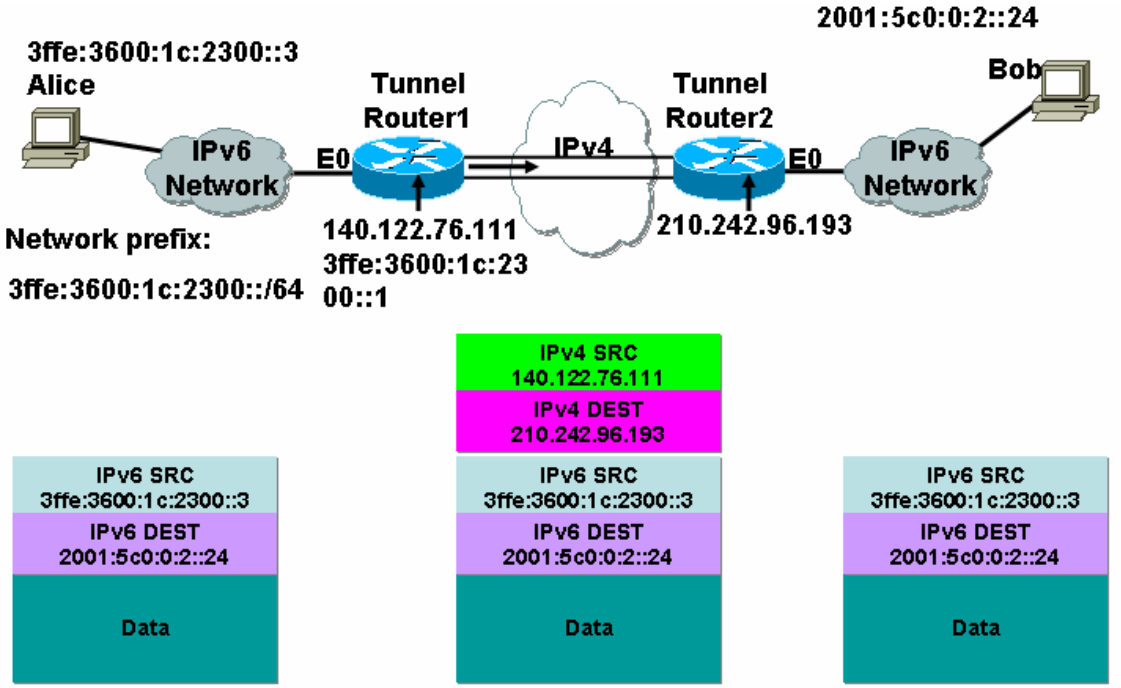


【圖十三】 6to4 Tunnel 示意圖

手動設定之 IPv6 Tunnel

另外一種常用的方式為到 Tunnel Broker 註冊申請建立 Tunnel 的方式，這種做法到 Tunnel Broker 註冊帳號後，建立一個新的 Tunnel 連線，對方會配發 IPv6

位址給我們，兩端設定建置 Tunnel。示意圖如下【圖十四】所示，Alice 必須設定為所在的 IPv6 網路位址，當 Alice 要傳送資料給 Bob 時，左邊的 Tunnel Server1 收到 IPv6 的封包時，直接加上自己的 IPv4 位址與 Tunnel Server2 的 IPv4 位址後，把封包透過 IPv4 網路傳送到對方，對方收到後將 IPv4 的表頭去掉後，透過 Tunnel Server2 的 IPv6 網路，將封包傳送給 Bob。



【圖十四】 手動設定之 IPv6 Tunnel 示意圖

由以上說明的 Tunnel 方式可以知道，電腦之間使用 IPv6 網路溝通時，不管網路的環境中是否有使用 Tunnel 的方式來設定 IPv6 網路，只要使用 IPv6 的標準互相溝通即可，中間的部份自動會做處理。因此增加了將網路更改為 IPv6 的方便性，並且可以在網路未全面升級成 IPv6 之前，測試 IPv6 的電腦是否可以正常使用 IPv6 的通訊協定互相溝通。

6to4 With VPN Tunnel

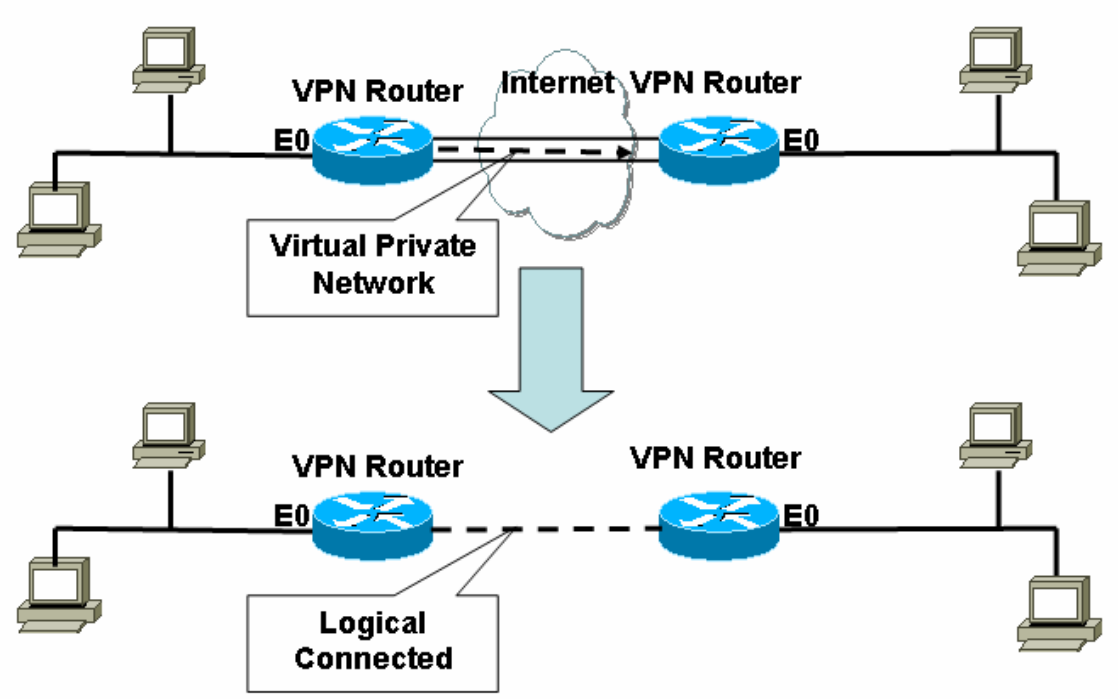
由於 6to4 Tunnel 與手動設定之 IPv6 Tunnel 都有一些小缺點，6to4 的方式需要設定哪些 IP 可以使用 6to4 Tunnel，只能使用連上來的主機之 IP 範圍做認證，無法提供帳號密碼的認證方式；手動設定之 IPv6 Tunnel 只能幫同一個子網路的電腦連上 IPv6 網路。為了解決這個問題，筆者使用 VPN 來做 6to4 使用者的認證，可先透過 VPN 的加密技術與通過認證後，接著再透過 6to4 Router 連上 IPv6 網路。

VPN 網絡能夠利用 Internet 為用戶創建隧道，並提供與專用網絡一樣的安全和功能保障，雖然 VPN 通訊建立在公開的網際網路上，但是用戶在使用 VPN 時感覺如同在使用專用網絡進行通訊。詳細示意圖如下【圖十五】所示，圖中雖然左邊的電腦與右邊的電腦實體上是在兩個不同的地方透過 Internet 連線，但若使用 VPN 技術後，兩邊變為等同於在同一個網路中的電腦，除此之外，VPN 具有以下幾個功能：

1. 封包加密：加密的技術依照金鑰的長度分為 DES 與 3DES…等。

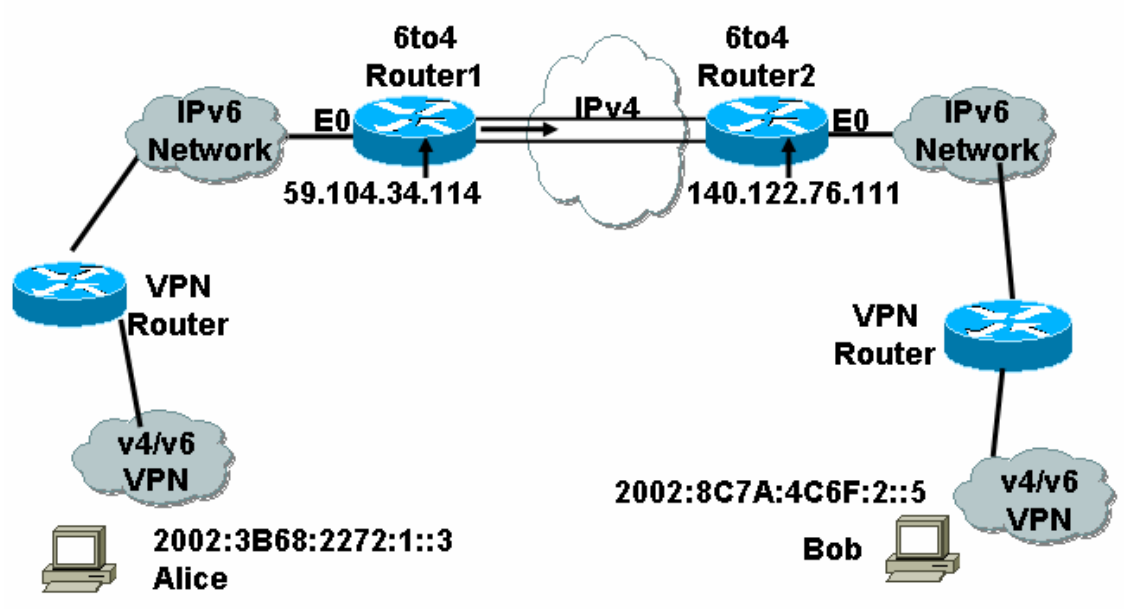
2. 存取控制：兩邊建立 VPN 連線時，會驗證使用者帳號與密碼，通過驗證後才可以建立 VPN 連線。

3. 封包驗證：兩邊建立 VPN 連線後，兩邊傳遞的封包有 AH[16]或 ESP[17]封包驗證的技術，確保封包在網際網路中傳遞時沒有遭到第三者修改傳遞的封包。



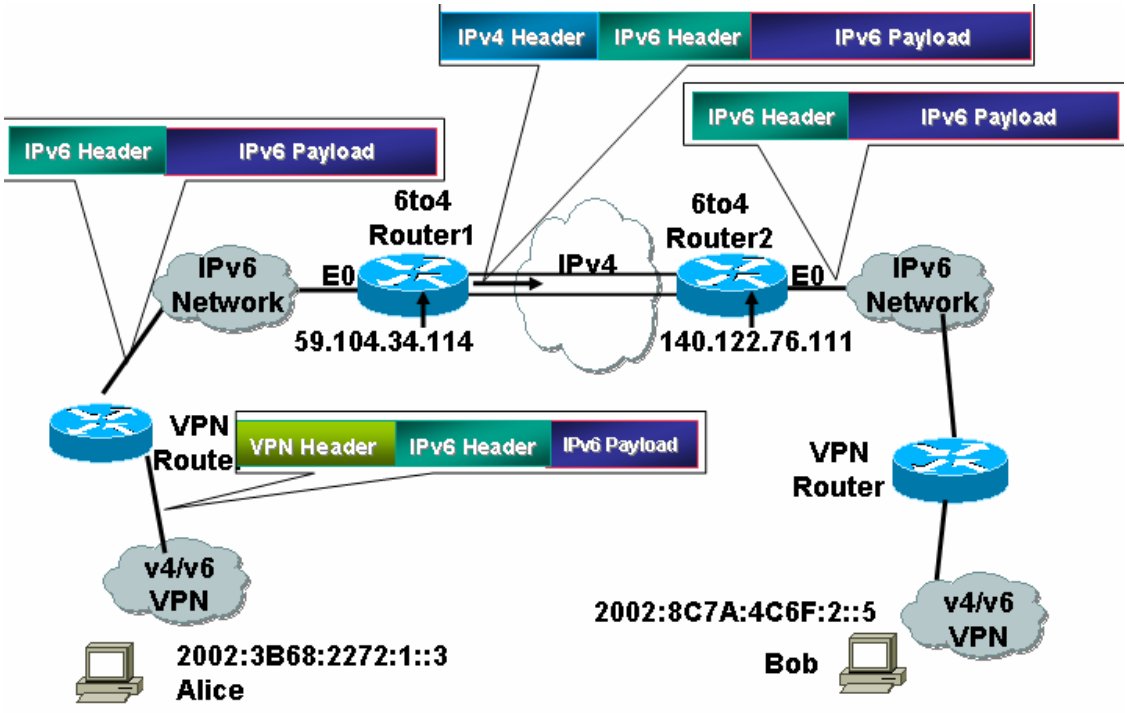
【圖十五】 VPN 連線示意圖

由於 VPN 具有強大的安全性功能，並且可以將 VPN 之間連線的電腦變為等同於同一個子網路，因此筆者將 6to4 與 VPN 同時使用後，達到安全性與連上 IPv6 網路的功能。此 6to4 with VPN 的架構圖如下【圖十六】所示，Alice 與 Bob 都透過 VPN 認證使用遠端的 6to4 Router 連上 IPv6 網路，因此 Alice 與 Bob 可使用 IPv6 訊息成功互相溝通。



【圖十六】 6to4 with VPN 架構示意圖

依照此架構示意圖，Alice 與 Bob 溝通時，傳遞的封包包裝如下【圖十七】所示，Alice 發送的 IPv6 封包先經 VPN Tunnel 包裝，到了 6to4 Router 時，包裝成 IPv4 格式封包在 IPv4 網路中傳遞，最後將封包訊息傳達到 Bob 端電腦。



【圖十七】 6to4 with VPN 網路封包圖

由此方法可以發現，不但可以提高安全性、驗證連上 IPv6 網路的使用者，並且不管在哪邊的 IP 只要可以通過 VPN 的驗證，就可以透過 6to4 Router 連上 IPv6 網路，解決了 6to4 只能設定某些 IP 可以連上的問題。這個方法不像手動設定的 Tunnel，只能幫同一個網路的主機連上 IPv6 網路。由於 VPN 在電腦端與 VPN Server 間使用 VPN Tunnel 的方式，可以通過 NAT 環境，因此 6to4 with VPN 也可以解決需要使用 Publice IP 的缺點。

目前常使用的 Tunnel 有其優點與缺點，這邊將 6to4 with VPN 的 Tunnel 方式與目前常用的兩種 Tunnel 方式做個比較如下【表六】所示：

【表六】6to4 with VPN Tunnel 與另兩種常用 Tunnel 比較

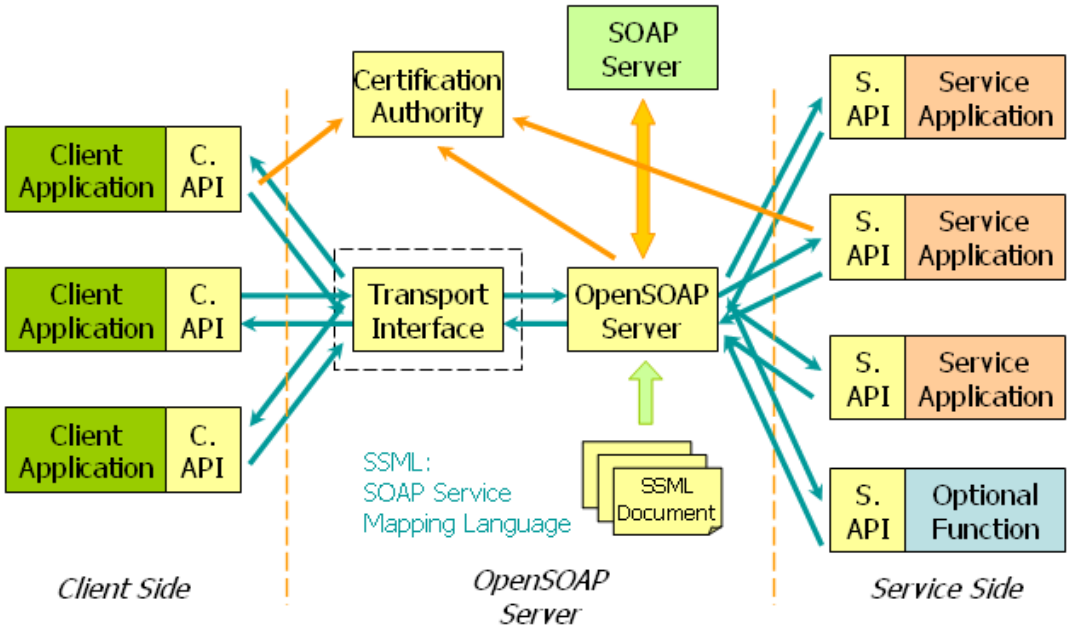
	手動設定之 IPv6 Tunnel	6to4 Tunnel	6to4 with VPN Tunnel
是否需要 Public IP	需要	需要	不需要
是否需手動設定	需要	不需要	不需要
是否可用於 NAT 環境	不可以	不可以	可以
使用時機	終端主機或是網路；Tunnel Server 端 IPv6 網路頻寬大時	設定簡單，需要快速導入 IPv6 測試時，指定 6to4 Router 即可連上 IPv6 網路	兼具 6to4 優點與 VPN 安全性優勢，需要安全性與方便性時採用。

目前 IPv6 正逢推廣期間，因此公開的 Tunnel Broker 可以輕易找到與申請。公開的 6to4 Router 也可以輕易找到，並且開放全世界電腦做 6to4 連線。不過這兩種方式各有其缺點，手動設定之 IPv6 Tunnel 方式設定後可以讓整個子網路的

電腦連上 IPv6 網路，但若是到另外一個子網路時就無法使用，必須另外設定一個 Tunnel；6to4 的缺點是目前公開的 6to4 Router 公開給全世界的 IP 有連線的權限，但這種方式只能使用 IP 區段做存取限制，存取控制安全性與方便性只能兩者選一。6to4 with VPN 的方式，不但可以有存取控制的功能，還擁有原本 VPN 封包加密的優點，是個較好的解決方式。

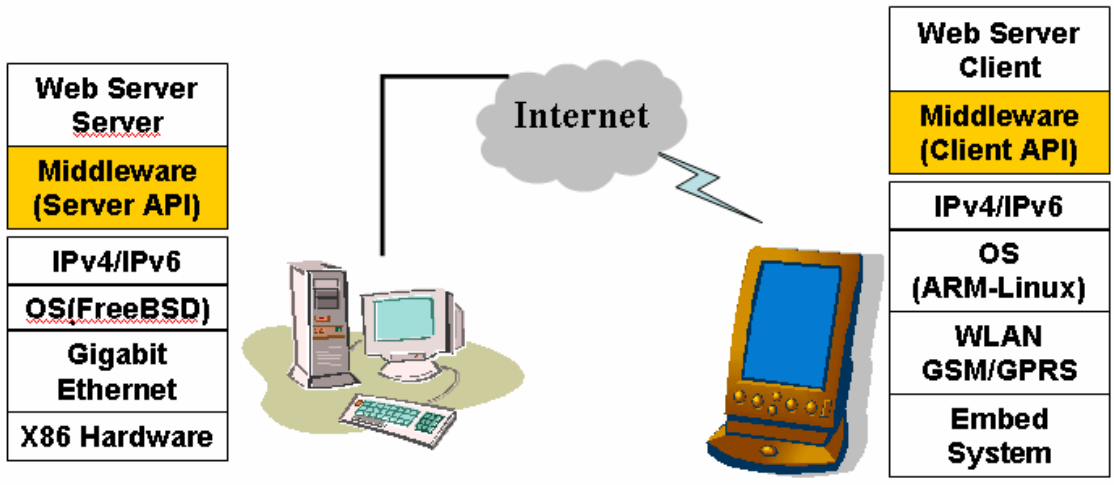
3.2 OpenSOAP

OpenSOAP 是日本 NEDO(New Energy and Industrial Technology Development Organization)發展的一個 Open Source 計畫，發展基於 SOAP 的 Middleware，本計畫成員大多為學校或是企業所組成，目前 Technoface Corporation 為此計畫之主要領導者，發展的理念是為了使用目前的軟體資源簡化 Web Service 的發展。OpenSOAP 主要包含兩個部份：OpenSOAP 伺服器與 OpenSOAP API。OpenSOAP 伺服器可以支援 Linux、Solaris8、Windows、FreeBSD…等平台，OpenSOAP API 可以支援 Linux、Solaris、HP-UX、IRIX、FreeBSD、NetBSD、Windows 2000…等平台。OpenSOAP 的架構圖如下【圖十八】所示，OpenSOAP API 有 Client API 與 Service API，Client Application 使用 Client API，Service Application 使用 Service API，中間的 Transport Interface 可以使用 Apache 代替。



【圖十八】 OpenSOAP 軟體架構圖
（資料來源：OpenSOAP 架構圖 http://www.opensoap.jp/doc/arch.png）

在本架構之下，只需要將 Transport Interface 改為支援 IPv6 的話，SOAP 即可以透過 IPv6 傳遞訊息，不需要修改 Service 與 Client 的程式。Service 與 Client 也可以輕易的分開，只需要分別安裝需要的 API 程式即可。OpenSOAP 於本研究之運作模式示意圖如下【圖十九】所示：



【圖十九】 OpenSOAP 於本研究之運作模式示意圖

OpenSOAP 的環境需求為系統中必須先有 libxml2 與 openssl 這兩個 package，

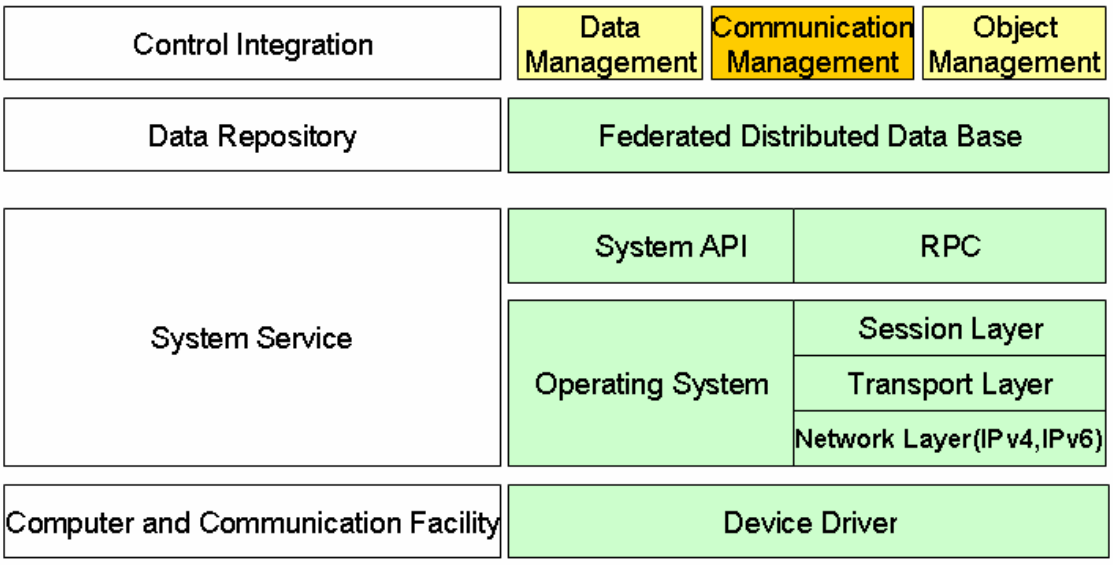
系統中若沒有的話，必須先依照系統安裝這兩個 package 的方法安裝後，接著才可以安裝 OpenSOAP 程式，若要將 OpenSOAP 跑在 Apache 上的話，則只需要安裝 API 即可。

OpenSOAP 為開放原始碼軟體，可自行修改架構以符合需求，筆者將其導入於 IPv6 網路環境中使用。軟體的架構可以將 Service 與 Client 的部份分開使用，因此筆者調整架構，將 Service 部份安裝於 x86 伺服器中，Client 部份只留下需要用到的幾個函式檔案，植入到嵌入式環境中，移植的過程中將 OpenSOAP 需要的 libxml2 與 openssl 這兩個套件也一起移植成嵌入式環境的格式，接著將 OpenSOAP 的 API 移植為嵌入式系統的格式。為了符合目前網路使用著重移動性的趨勢，將移植後的嵌入式環境加入無線連網功能，使成為 IPv6 為基礎之嵌入式環境 SOAP 無線連網架構。

3.3 IPv6 Middleware Framework

根據 Middleware[25]的定義，Middleware 是個應用程式與底層作業系統溝通的介面，應用程式可以透過 Middleware 的 API 程式與作業系統做溝通，而不需應用程式直接呼叫作業系統的函式。由於每個作業系統的函式可能不同，因此若應用程式直接與作業系統溝通的話，換到另外一個作業系統開發應用程式的時候，有關於作業系統函式部份必須全部修改；相反的，應用程式如果透過 Middleware 與作業系統溝通的話，應用程式在不同的作業系統開發時，可以只要更換 Middleware 的部份，應用程式一樣透過 Middleware 的介面與作業系統溝通，可以不用修改應用程式的內容。

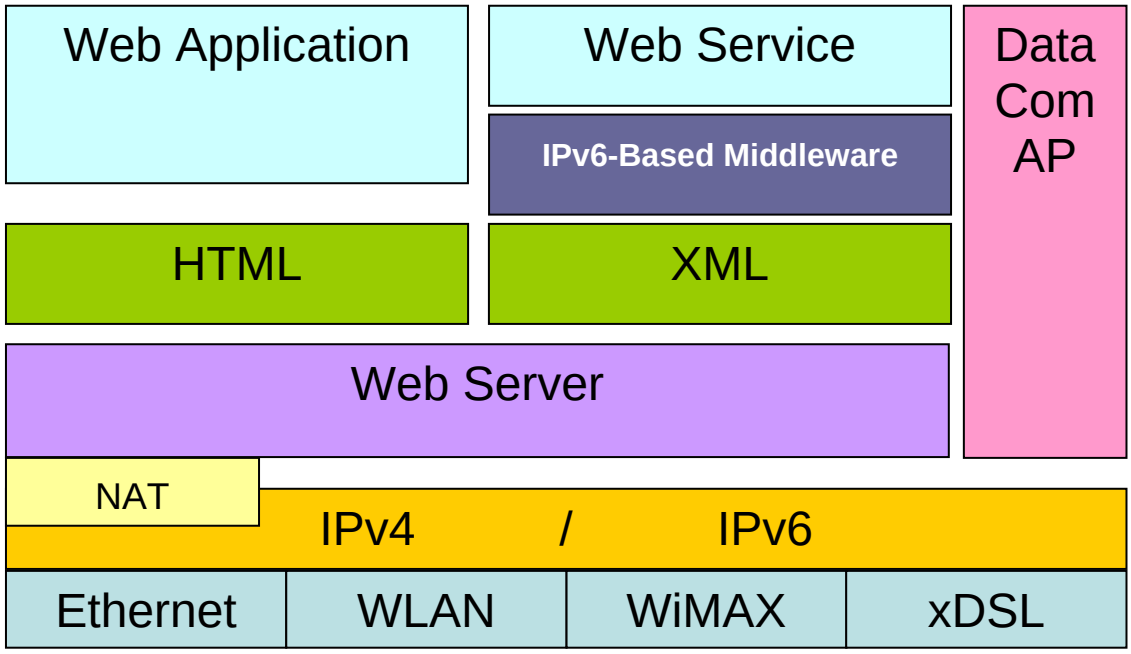
根據如下【圖二十】的架構圖[25]，Middleware 的 Control Integration 功能主要包含三個部份：Data Management、Communication Management 與 Object Management。近年來 XML 的使用，可以視為 Data Management 的例子，資料與訊息的傳遞，使用 XML 來管理，大家使用共同的標準來溝通，上層的應用程式不管在什麼平台中，都使用 XML 作為資料存取的統一格式。程式之間透過 XML 做溝通橋樑，程式只要可以存取 XML 格式的檔案即可，應用程式不需要因為使用不同的程式語言而修改儲存的資料格式，無須考慮使用哪種程式語言，而有 Data Management 的概念。之後 Web Service 的出現，讓電腦與電腦間溝通，使用服務的概念來完成，每個 Web Service 都是 Object，因此 Web Service 變為以服務為導向的 Object Management 概念。而 Communication Management 目前主要為管理 WAN 和 LAN 之間的溝通或者整合無線與有線的連網功能。



【圖二十】 Middleware 元件架構圖

本章第一節提到的 IPv6 Tunneling 的方式使兩個電腦間通訊時可使用 IPv4 或者使用 IPv6，中間的 Router 會自動做處理，將 IPv6 封包包成 IPv4 的格式傳遞，

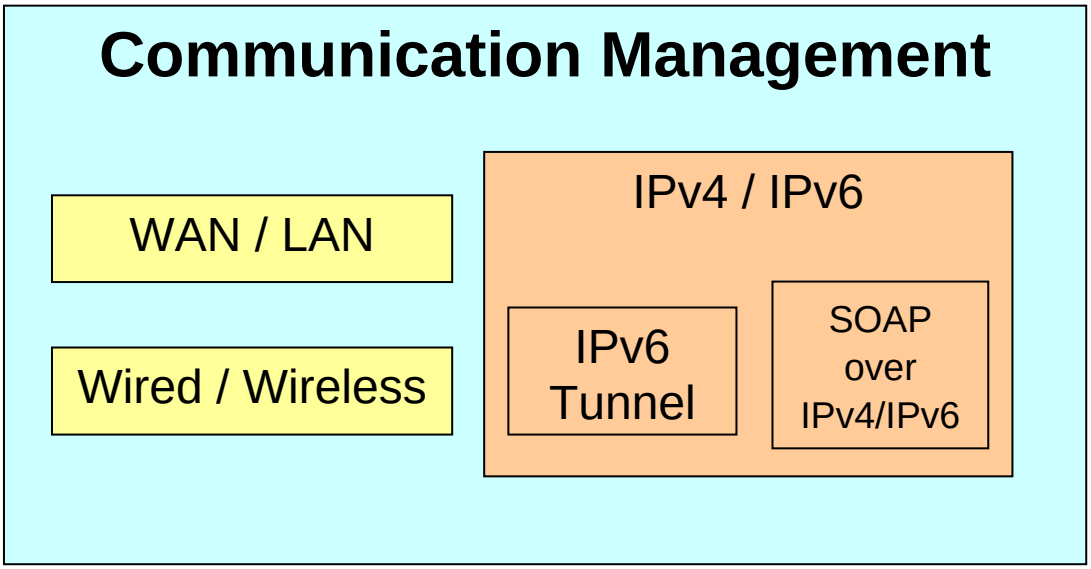
電腦不用因為要使用 IPv4 或者 IPv6 而做其他設定;本章第二節中提到的整合 IPv6 功能之 OpenSOAP 在如下【圖二十一】的架構圖中，可視為扮演 IPv6-Based Middleware 的角色。即使在不同的作業系統中，只要安裝上 Middleware 程式，在實作 Web Service 程式時只需要呼叫 Middleware 的 API 程式就可產生 Web Service 功能，並且可同時支援 IPv4 與 IPv6，不必因在不同的作業系統，或因不同的網路協定，而必須製作多個不同版本的程式。



【圖二十一】 IPv6-Based Middleware 架構

資訊網路技術是由底層漸漸發展的，由於底層的完備，才能繼續向上發展，轉移為 IPv6 的技術必須成熟，且安全性足夠，大家才會快速的導入，Middleware 的發展整合完善，大家才會繼續向上發展網路應用程式。本章第一節與第二節的概念，將現有的 Communication Management 範圍更加擴大為如下【圖二十二】。既有的 Communication Management 只探討左邊 WAN/LAN、Wired/Wireless，本研究將範圍擴大到 IPv4 與 IPv6 的通訊整合，IPv6 Tunnel 讓使用者不必因為使用

IPv4 或者使用 IPv6 而修改上網的方式。IPv6-Based Middleware 讓程式開發者不必因為使用 IPv4 或者 IPv6 網路而需要實作不同版本的程式。



【圖二十二】 本研究延伸之 Communication Management 架構