

第二章 相關文獻探討

2.1 問題研究

家庭網路服務社群的建立是以分散式網路環境為基礎，利用家用開道服務進行整合以提供服務，在以網路為基礎的智慧型環境中欲解決智慧型節點的裝置異質性問題、裝置互相察覺性問題、裝置資源不足性問題；以及智慧型空間的隨到即用問題、複合式服務整合問題，詳述如下：

(1)在智慧型節點方面，居家中的一項裝置、一項軟體應用程式都是居家服務，服務來自眾多的廠商，以合作、協商等方式直接或間接提供給使用者，因此必須滿足以下三項性質：

- 服務異質性：家庭網路在整合居家中所存在的眾多服務廠商所提供的軟體及硬體服務，不同的協定與標準成為裝置互通有無的障礙，家庭網路必須解決居家中基本的裝置異質性問題以貫串居家裝置，進而聯合軟體服務，對異質服務進行整合性的管理。
- 服務的互相察覺性：普及運算的進步促使服務提供者提供服務的形式可以是整合自其他多個不同的服務提供廠商或服務提供者，形成服務使用服務以提供服務的形式，家庭網路必須提供服務互相察覺機制以滿足家庭網路中服務互相操作的需要。
- 裝置資源不足性：鑒於行動裝置愈趨小型化與資訊家電中嵌入式系統設計，裝置的資源不足性愈趨明顯，家庭網路服務的設計必須考慮執行服務裝置的資源問題，以動態且微型服務來滿足行動裝置與資訊家電的裝置特性。

(2)在智慧型空間方面，智慧型空間提供服務的註冊與取用機制，作為服務執行的環境，因此必須滿足下列兩項特性：

- 隨到即用性：為滿足以網路為基礎的智慧型空間中居家服務新增與行動裝置移動性問題，家庭網路必須提供新增裝置與行動裝置在進入居家環境中立即可以操作居家服務的需要，增加使用上的便利性。
- 複合式服務整合管理：家庭網路統整各種服務後再經由整合管理機制以提供使用者整合性的服務，服務之間可以透過管理機制自動的互相取用或進行排程，以達到自動化服務的功用。

2.2 服務搜尋機制

在這一節裡，我們主要針對家庭網路內，不同服務搜尋協定進行探討。這些協定所要解決的共通問題是如何讓網路上的設備之間互動更加自動化，包括找到彼此的方法，了解對方能力的方法以及使用對方資源的方法等。首先，先討論所有服務搜尋機制的共通特性，再個別介紹 Jini、UPnP、SLP 和 Bluetooth SDP 搜尋機制的運作方式。

2.2.1 服務搜尋機制介紹

服務搜尋機制是為了發展動態 Client/Server 應用程式所產生的。在服務搜尋機制下，使用者可以動態存取周遭的服務，要如何尋找可用的服務以及要去存取這些服務是很重要的問題。在服務搜尋網路環境內，每個軟體、硬體資源都會被抽象成服務的型態。

在服務搜尋網路裡的服務可以根據服務屬性分類，例如：依照位置、服務種類、製造商、性質等分類。因此，使用者可依照服務屬性來搜尋需要或有興趣的服務。每個設備當進入網路環境時，會通知其它使用者它所提供的服務屬性；當設備離開時，亦會知會網路上的其它使用者。因此，其它網路的使用者可得知目前在網路上還有那些可供使用的服務。在網路內也會有個 Service Catalogs 追蹤著目前網路上尚可使用的服務。Service Catalogs 扮演著集中式的目錄管理服務。

服務搜尋機制的資源回收機制(Garbage collection)可避免系統仍保存著過時的資訊，例如，某個客戶端已不存在，但伺服器端仍以為其仍在存取服務，藉由資源回收機制會在一起時間後確認回收資源、更新資訊。

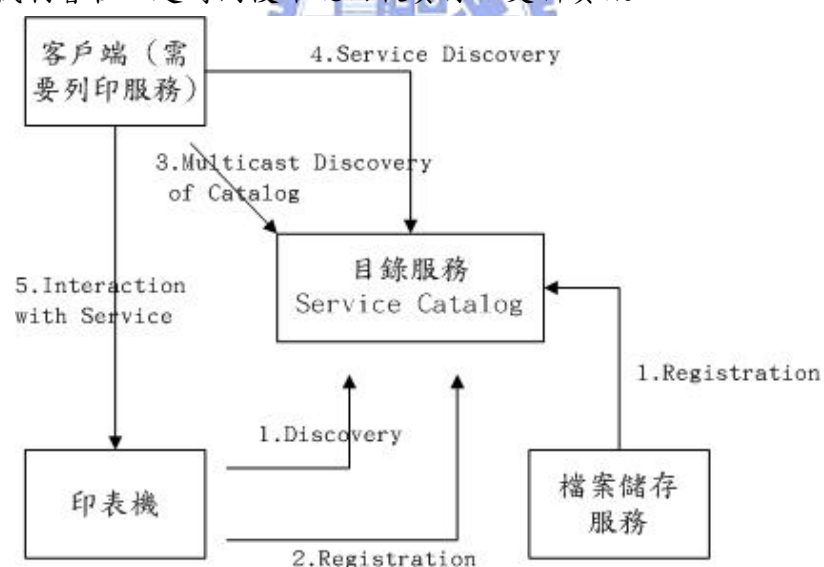


圖 2.1 服務搜尋流程。

在服務搜尋網路內具備二種角色，分別為客戶端(Client)和服務端(Service)。客戶端執行搜尋(Discovery)去取得所需要的服務。不過，客戶端也可以直接存取(Directly Seek)所需要的服務。但是一般的情況，客戶端會詢問 Service Catalogs 是否有所需使用的服務存在於網路。搜尋的結果會依據服務的型態、製造商、序號或其它服務屬性加以分類回覆給客戶端。

不論是直接存取或是經由 Service Catalogs 進行搜尋，客戶端都可以很動態的即時查詢和存取資源，不需要事先靜態組態環境。搜尋的方法主要是使用

multicast 的方式。當一個服務進入網路，它會先執行服務廣播(Service Advertisement)，針對網路上的客戶端或是 Service Catalogs。這個 Advertisement 包括和服務聯絡的相關資訊、服務的屬性描述。圖 2.1 描述一個搜尋的過程。

首先(1)檔案儲存服務知道線上有個 Service Catalog，並且註冊它的服務。同時，有個印表機設備進入網路，由於它事先並未設定 Service Catalog 的位置資訊，因此使用 Multicast 動態查詢 Service Catalog 的位置，(2)並且註冊印表機的資訊。之後，(3)客戶端需要列印服務，因此在找到 Service Catalog 後，向 Catalog 查詢可使用的印表機資訊，(4)最後，跟印表機交談，請求列印服務。

藉由服務搜尋機制，應用程式將更容易的支援動態的行為，像動態進入網路、位置的改變或是服務移除等。使用者也不需面對麻煩的管理問題。

2.2.2 服務搜尋機制的特性

服務搜尋機制提供標準的 Framework 解決了許多 Client/Server 系統的問題。在設計 Client/Server 系統時主要的問題包括：

- 目前提供什麼類型的服務？(無法提供關於服務的屬性描述)
- 服務的位置在那？(需要事先知道服務的位置)
- 客戶端使用這個服務，需要些什麼要求？(額外的需求需事先符合)
- 客戶端和服務端間所使用的協定為何？(協定需事先約定)

服務搜尋機制解決上述問題，提供建立動態、分散式通訊系統，並幫助開發者處理部署 Client/Server 應用程式設計和管理的問題。服務搜尋機制准許服務動態搜尋、自我組態，並使用最少的人工手動設定。

2.2.3 共同特性

由於目前服務搜尋機制具有很多種，共同的主要功能包括：

- Discovery of services：當需要服務時，可動態依需求搜尋，毋需事先對網路有所認知。使用者可使用服務型態或是描述屬性進行查詢；這種多樣性的查尋方式是在不同的服務搜尋機制裡，提供不同的方式。例如，Service Location Protocol(SLP)提供非常強大的屬性查詢機制，而在 Bluetooth SDP 則只提供整數數值的屬性查詢。
- Service “subtyping”：使用者可能偶爾只對特定型態的服務有興趣。例如，需針對高解析、彩色雷射印表機進行列印，而其它的單色印表機則不需要；使用者可依據他需要的特色找到需要的服務。
- Service insertion and advertisement：服務進入和離開網路時會進行廣播告知其它在網路的成員。讓其它服務可以動態更新資訊，而非依賴靜態的設定。

- Service browsing：使用者可以瀏覽目前網路上所提供的服務，再由使用者決定要使用什麼服務。
- Catalogs of available services：一些服務搜尋機制，像 UPnP 主要是採用 peer-to-peer 的方式，因此 advertisement 和 discovery 都是直接彼此存取。但像 Jini 則具備個集中的 Catalogs，主要負責追蹤服務的狀態。而 SLP 網路裡，則是 Service Catalogs 可有可無。在具備 Service Catalogs 的服務搜尋環境裡，服務會註冊它們的資訊於 Service Catalogs，而使用者則直接向 Service Catalogs 取得相關服務資訊。藉由 Service Catalogs 可以減少 Multicast 所造成的網路流量，且可藉由多個 Service Catalogs 的建置，加大服務搜尋的範圍。
- Eventing：事件機制准許使用者訂閱有興趣的事件通知，採用非同步的通知方式。去除需不斷的詢問(polling)的方式。
- Garbage collection：藉由資源回收機制可終止過時的服務資訊，並終止相關的客戶端訂閱的事件通知服務等相關資訊。租約(Lease)是最主要的資源回收機制。Lease 的做法是，要取得服務，需先經由服務提供者同意，會指定某個時間給使用者。時間到時，如果使用者沒有 renewal 的話，那服務提供者便會終止 Lease，將相關資源清除。

2.3 Jini 網路架構

Jini 是由昇陽所提出以 Java 為基礎的協定。Jini 的架構中有三個主要的元素，服務(Service)、查詢伺服器(Lookup Server)和使用者(Client)。服務提供資源給使用者，而查詢伺服器則扮演媒介服務和使用者的中間人。服務必需向查詢伺服器登錄自己的資料，並上傳一個遠端控制物件(Remote Control Object)，而使用者透過查詢伺服器找尋有用的資源，下載其遠端控制物件，並透過 Java 的遠端呼叫(Remote Method Invocation 或稱 RMI)方式與服務互動。由於 Jini 完全是以 Java 為基礎而開發的，所以繼承了所有 Java 的優點，可以跨平台、安全性高、高度整合的應用程式界面等，但同時也代表了，所以支援 Jini 的設備，都必需有一個 Java 需擬機器(JVM)以執行 Java 程式。

2.3.1 架構簡介

Jini 技術是由基本架構(Infrastructure)、程式模組(Programming Model)、服務(Service)三部份所組成，是由各個元件所組成之分散式網路系統，可隨時隨地簡單存取網路資源。

Jini 是由昇陽運用 Java 平台所發展出來，在1999年一月正式推出的新技術，它是一個很小的 Java 程式碼，具備 Java 支援跨平台、可在任何地方執行的特性。Jini 訴求之功能為：免安裝、免設定、隨插即用自動連接之功能，並且能達到即興式網路動態顯示目前連線的各種設備與服務，此外透過 Jini 轉接器，傳統周邊設備也能有 Jini 功能，使得設備之間可以透過下圖的架構，達到服務互相分享的功能。

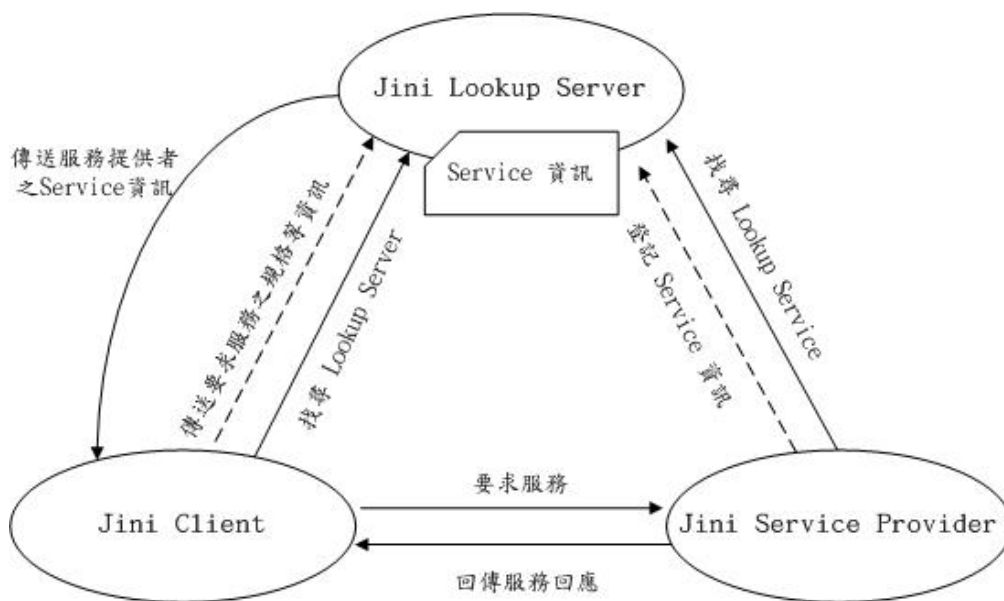


圖 2.2 Jini 架構

如圖2-2 所示，Jini 架構中包含三個角色分別是：

- Jini Lookup Server：提供Jini Client 搜尋服務與Jini Service Provider 註冊服務的功能；
- Jini Service Provider：提供Jini Client 所需的服務；
- Jini Client：為Client 端，向服務提供者要求所需的服務Jini Service Provider 能提供Jini Client 各種類型特定的服務。

Jini Service Provider 與Jini Client 兩者之間的供需關係是藉由Jini Service Provider 事先向Jini Lookup Server 註冊，隨後Jini Client 可以向Jini Lookup Server 提出搜尋服務要求，在Jini Lookup Server 的協調下完成供需雙方的需求，透過這三者的相互配合，可以建立無所不在的連結環境。

2.3.2 協定堆疊

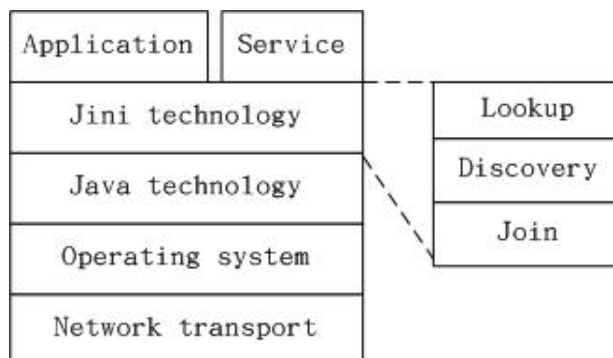


圖 2.3 Jini 協定堆疊

Jini 3 個主要核心協定：Discovery、Join 和 Lookup。Discovery 和 Join 這兩個協定是一對的，當使用者將 Jini 設備加入網路環境時使用的。Discovery 是用於當服務需尋找 lookup service，並註冊服務時。當找到 lookup service 後，便會將自己的服務加入這個社群，此時便使用 Join 協定。Lookup 主要用於當使用者或客戶端藉由服務描述進行尋找，並呼叫服務時。

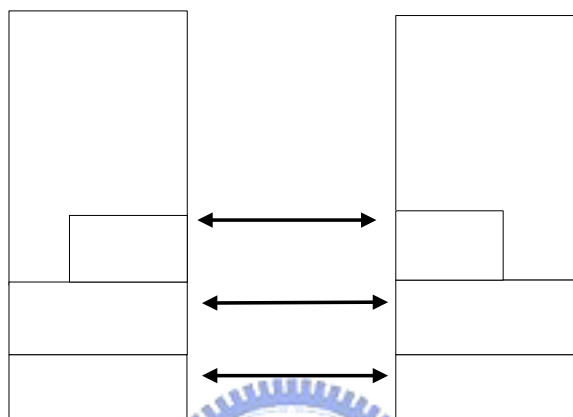


圖 2.4 RMI 機制

Jini 技術包含 Jini 動態網路基礎架構和程式設計模型，讓每個設備可以動態的形成一個社群(community)，彼此互相存取。Jini 使用 Java remote method invocation(RMI)協定，來動態載入或移動程式碼。

Jini lookup service 可視為一個目錄伺服器或是一個仲裁服務。Jini 使用三種相關聯的 discovery 協定。當一個應用程式或服務進入網路，變成 active 時，首先使用 multicast request 協定尋找網路上的 lookup service。Lookup service 使用 multicast announcement 協定來通知目前已在網路上的服務，Lookup service 本身相關的訊息。最後是 unicast discovery 協定是用如果先前已知 lookup service 的位置，則直接使用 unicast 建立和 lookup service 的連線。

然而，Jini lookup service 並不只是簡單的名稱伺服器。它處理著客戶端使用的 interface 對映到 service proxy 物件。它也負責維護服務的屬性和客戶端提出的服務查詢。客戶端由 Lookup service 下載所需服務的 service proxy 物件，這個 service proxy 相當於 RMI 內客戶端的 stub，藉由客戶端的 stub 和伺服器端服務的 skeleton 彼此溝通，呼叫服務。藉由這個 service proxy 物件，客戶端毋需事先了解 service proxy 的實作細節。因此，當服務提供者是一個設備時（例如印表機），客戶端不需要事先安裝該服務的驅動程式，而改由動態下載 service proxy 物件來處理和服務的溝通。客戶端下載的 service proxy 物件，本身可能提供所有的實作功能，或者是扮演 stub 角色，藉由遠端呼叫完成工作。

2.3.3 運作流程

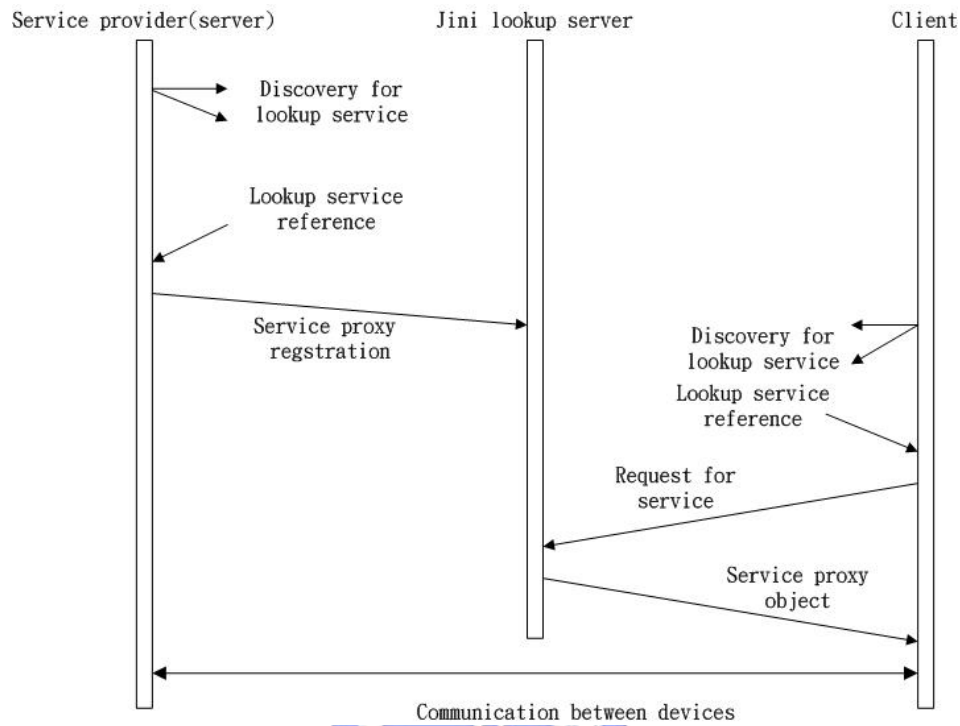


圖 2.5 客戶端和服務提供者間 Jini 的運作流程

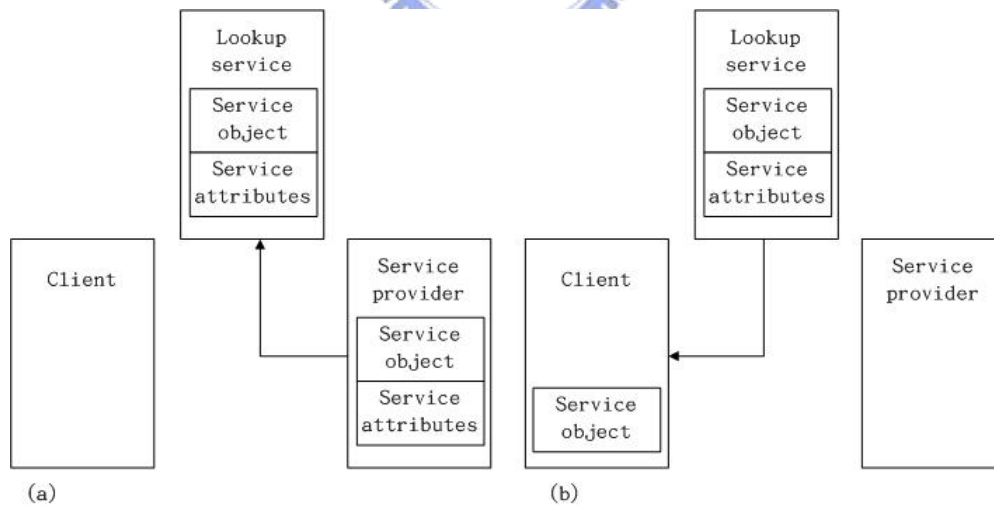


圖 2.6 客戶端使用服務提供者服務的流程

如圖 2.5、圖 2.6 描述。當一個 Jini 社群內的客戶端要使用一個服務，需經下列步驟：

- 服務提供者必需先使用 multicast 來尋找 lookup service，或是服務提供者依據先前資訊已知道 lookup service 的位置。(見圖 2.6 (a))

- 服務提供者必需向 lookup service 註冊它的服務物件和它的服務屬性。這個服務物件包含 Java 程式 interface，用於描述服務提供給客戶端呼叫的 method 資訊。(見圖 2.6 (a))
- 客戶端可以藉由 Java 型態或是服務屬性來要求需要的服務。Lookup server 找到符合的服務，便會複製服務物件，經由網路下載至客戶端。(見圖 2.6(b))
- 客戶端取得服務物件後，直件使用服務物件和服務交談，呼叫所需要的服務。(見圖 2.6 (b))

2.3.4 Jini 應用範例

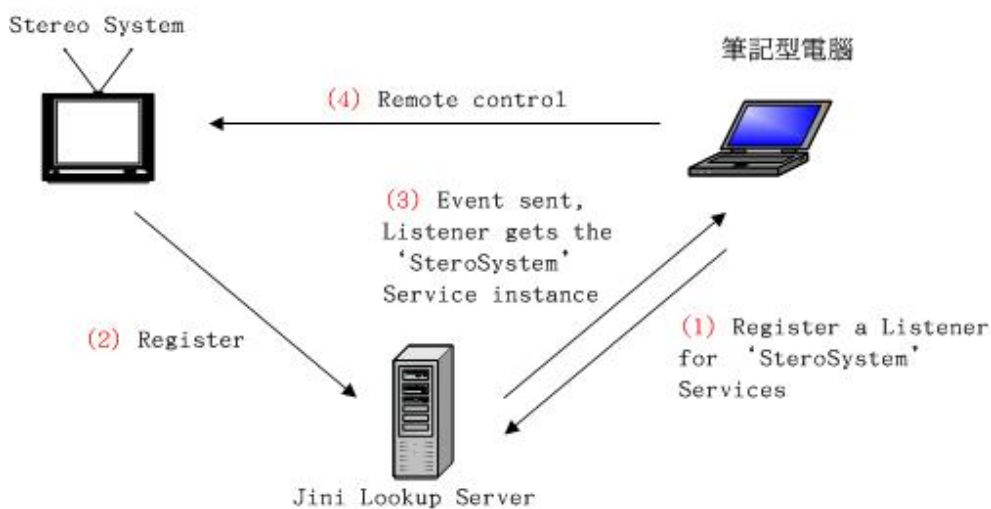


圖 2.7 Jini 搜尋範例

首先筆記型電腦由於需要影音相關服務，因此先向 Lookup Server 註冊一個 Listener。當有符合的服務時，由 Lookup Server 通知筆記型電腦。接著，一個影音系統進入網路，並向 Lookup Server 註冊，提供它的服務物件。Lookup Server 收到後，會將服務物件傳給筆記型電腦，筆記型電腦則使用這個服務物件進行遠端呼叫，執行載入歌曲清單、播放歌曲等功能。

2.4 UPnP 網路架構

UPnP 是由微軟所主導 UPnP 論壇所提出的標準，和 Jini 完全倚靠不同成員間傳遞 Java 程式碼為主的架構不同，UPnP 的訴求在將設備之間通訊的協定以 XML 格式訂出統一的標準，因此 UPnP 實際上是由許多協定所組成的。UPnP 架構主要的組成為主控者(Control Point)和設備(Device)。主控者相當於 Jini 中的使用者，設備則提供服務讓主控者使用。主控者和設備間的通訊並不通過第三者，先以 SSDP(Simple Service Discovery Protocol)找到對方，再以 SOAP(Simple Object Access Protocol)達到遠端呼叫的目的，而事件的傳遞則藉由 GENA(General Event Notification Architecture)達成。

2.4.1 架構簡介

UPnP 以分散式架構為主軸，以點對點 (Peer to Peer) 連接的方式，將所有的設備連接起來，是一種使用 TCP/IP 和 Web 技術的開放式網路架構。所謂的點對點連接方式，就是不經由中央伺服器，每個設備都能主動或被動地和其它設備溝通。

UPnP 中的 U 表示 Universal，其意義為：

- 以通用的協定代替廠商自行發展的驅動程式
- 獨立於底層的實體線路以及通訊協定之外
- 可以用任何的程式語言在任何的平台上實作
- 結合了開放式的網路技術，能夠輕易地處理不同平台間相容易的問題
- 設備可以經由 Web 界面操控，使用者無需安裝操控程式
- 設備製造商視需要可以自行擴充協定的功能

而 PnP (Plug and Play) 代表了在這樣的架構下，可以輕易地建立一個無需使用者設定，就能自行連上網路，和其它設備互動的網路環境。因此，無論在家中、在辦公室甚至任何一個地方，都能夠方便地使用網路上的資源。

UPnP 網路中有幾個重要的元件：

- 主控者 (Control Point)：能夠使用其它設備所提供的資源，可以看作是客戶端 (Client)，例如 PDA。
- 設備 (UPnP Device)：通常是硬體裝置，允許主控者使用它的資源，例如印表機。
- 服務 (Service)：每個設備都會提供服務，以供客戶端使用，例如印表機提供列印服務。
- DHCP Server：能夠動態指定 IP 及網路設定給設備，一個 UPnP 網路環境不一定要 DHCP 伺服器。有 DHCP 伺服器的網路環境稱作受到管理的網路 (Managed Network)，反之，稱作未受管理的網路 (Unmanaged Network)。

一個設備可以提供數個服務，也可能會有子設備在其中，一個設備可能同時也是一個主控者。

2.4.2 協定堆疊

UPnP 是一個實際上是由許多不同的通訊協定組成的一個協定組，以 XML 為資料交換格式。包含的通訊協定有 GENA (General Event Notification Architecture Base) 負責資訊傳遞；SOAP (Simple Object Access Protocol) 用來取用網路上其它裝置提供的資源；SSDP (Simple Service Discovery Protocol) 負責

找尋網路上可用的裝置。AutoIP 能夠在沒有 DHCP 的網路環境中讓設備得以連上網路。另外，使用 HTTP 作為溝通的媒介。

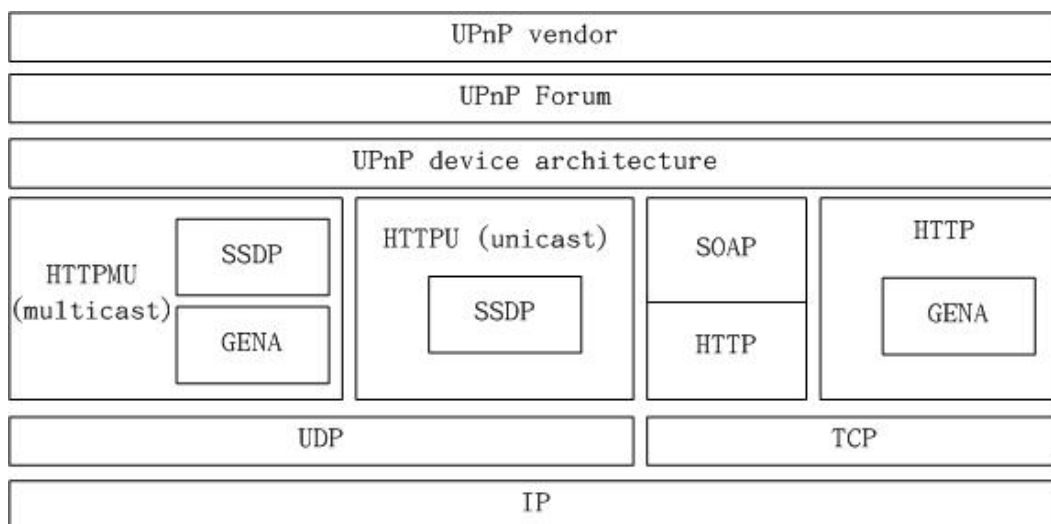


圖 2.8 UPNP 協定堆疊

2.4.3 運作流程

整個 UPNP 網路運作流程可以分成六個層次來說明

- Addressing
- Discovery
- Description
- Control
- Event
- Presentation

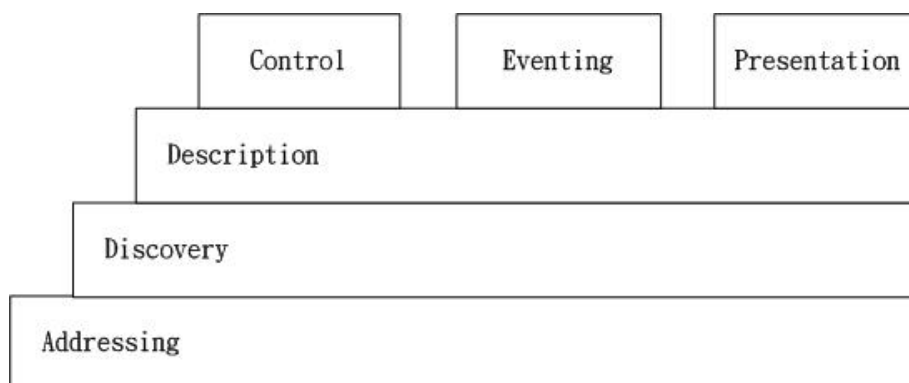


圖 2.9 UPNP 運作流程

Addressing：UPnP 網路是架構在 IP 之上，每一個設備都必需具備 DHCP 客戶端 (DHCP client) 的功能。當設備連接上網路時，首先要找尋 DHCP 伺服器，若在

網路環境中存在一個 DHCP 伺服器，設備就會從 DHCP 伺服器取得 IP 位址以及基本的網路設定；若找不到 DHCP，設備必需使用 AutoIP，自行設定一個私有 IP。

Discovery：當一個設備加入網路後，會使用網路廣播，告知主控者有一個新的設備加入；同樣的，當一個主控者加入網路後，會搜尋網路上是否有合於需求的設備。在這一階段，實際交換的資訊非常的少，只有設備的類別 (device type)、識別碼 (device ID) 和描述設備完整資訊的檔案位置。所使用的通訊協定是 SSDP。

Description：主控者找到有興趣的設備後，對設備的了解仍然很少，因此根據上一階段的檔案路徑，主控者使用 HTTP 協定下載關於設備完整的資訊。設備的描述檔使用 XML 格式，包含了製造商所提供的資料如設備的類型、型號、製造商的網址等等，同時也列出這個設備包含了那些子設備、它們所提供的服務為何以及註冊的路徑。對於每一個服務，描述檔列出了能接受的指令或稱作動作 (Action)、參數為何以及服務的狀態變數 (State Variable)。

Control：當主控者取得設備的註冊資訊後，就可以利用設備所提供的指令操控設備，使用設備所提供的服務。主控者根據描述檔裡的資訊，對設備所提供的服務送出一個控制訊息 (Control Message)，控制訊息也是以 XML 語法所寫成，底層的通訊協定為 SOAP。

Eventing：每一個服務都會有一個或多個狀態變數，當這些變數改變時，提供服務的設備必需公告變動的結果，與這些服務有關的主控者必需向服務註冊，以便收到變更的資訊。公告變更的方式是送出事件訊息 (Event Messages)，事件訊息的內容為狀態變數的名稱和最新的變數值，同樣是以 XML 語法描述，所遵循的規範為 GENA。

Presentation：若一個設備提供了網頁控制的功能，主控者能夠根據設備提供的位置，使用瀏覽器連上網頁，並且利用瀏覽器操作設備，但網頁控制並不需要實作設備提供的所有功能，可根據需求有所調整。

2.4.4 UPnP 應用範例

這裡以一個簡單的範例講解 UPnP 網路運作情形，在範例中有一個設備電視提供了控制電視的服務，另外有一台個人電腦，以及 PDA。個人電腦是一個主控者，同時也是設備，提供了行事曆服務。一開始，個人電腦是開著的，電視以及 PDA 則尚未連上網路。如圖 2.10 至圖 2.16。

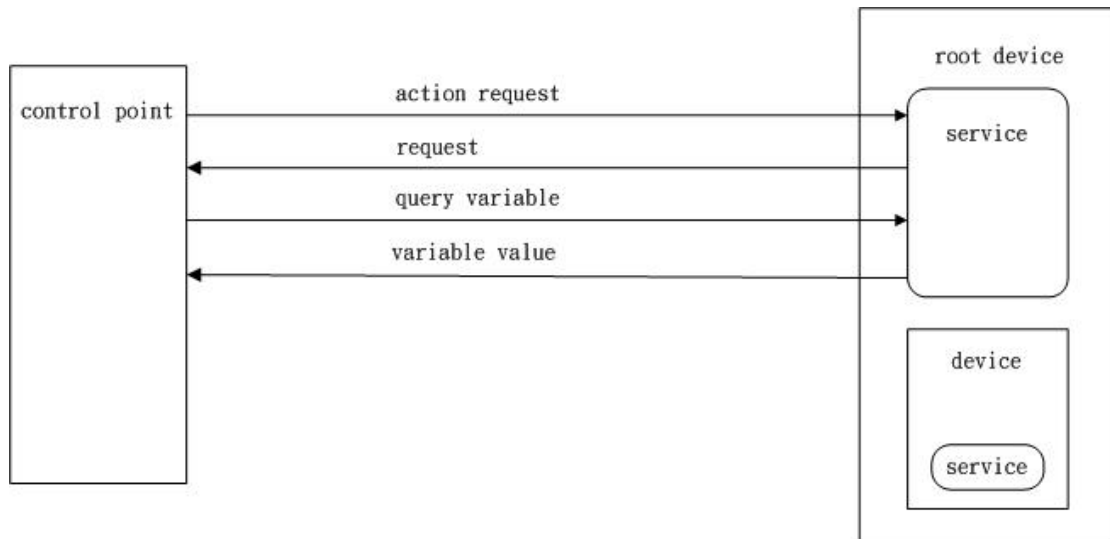


圖 2.10 UPnP 控制流程示意圖

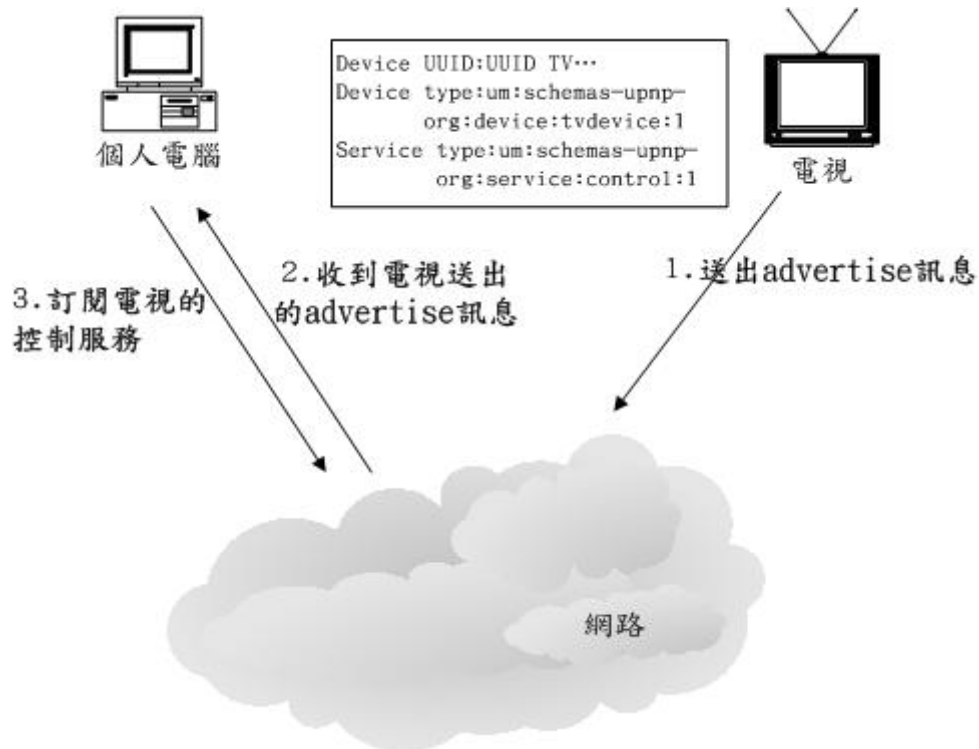


圖 2.11 UPnP 應用範例一

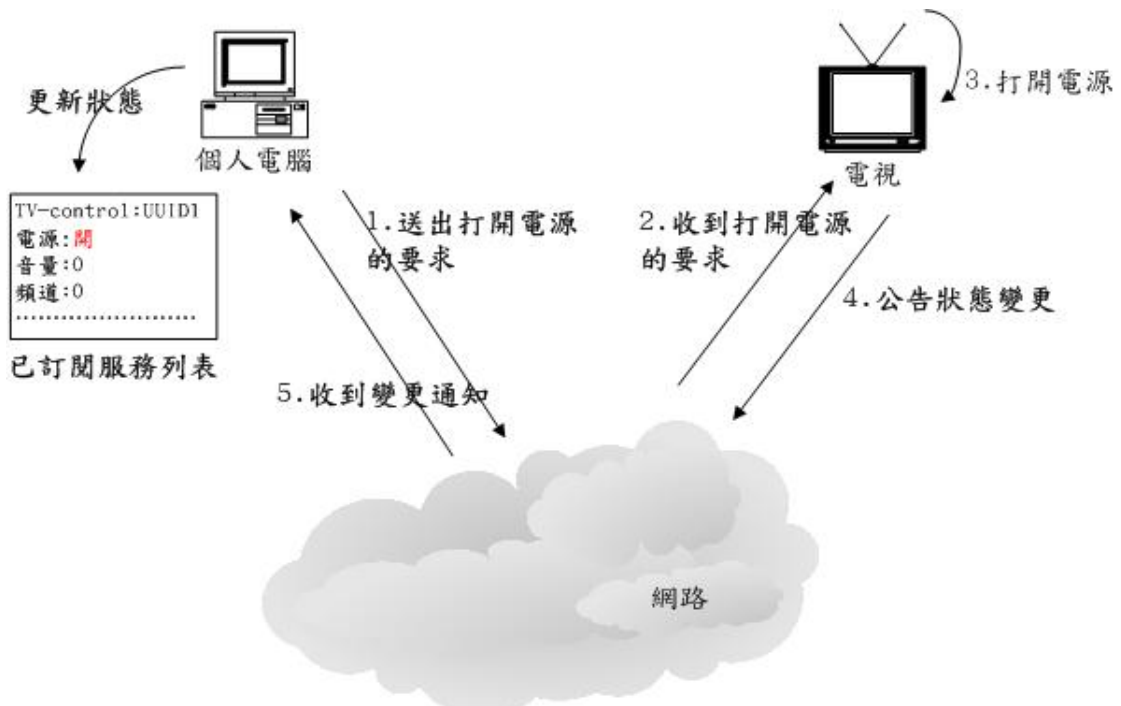


圖 2.12 UPnP 應用範例二

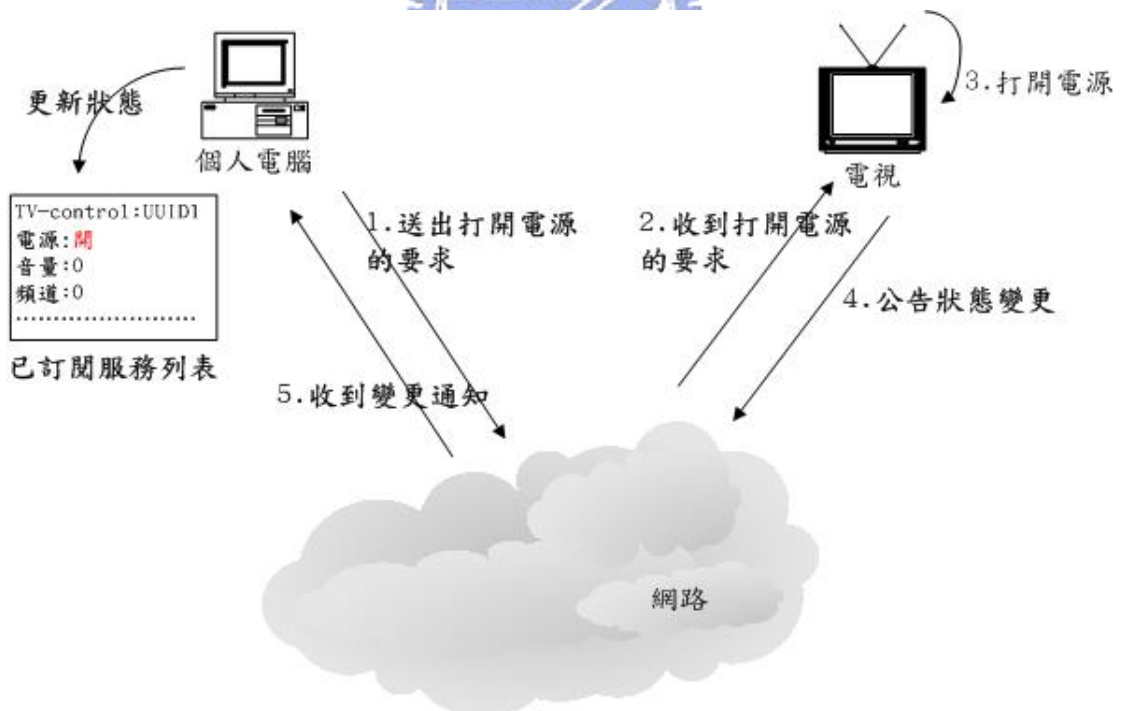


圖 2.13 UPnP 應用範例三



圖 2.14 UPnP 應用範例四

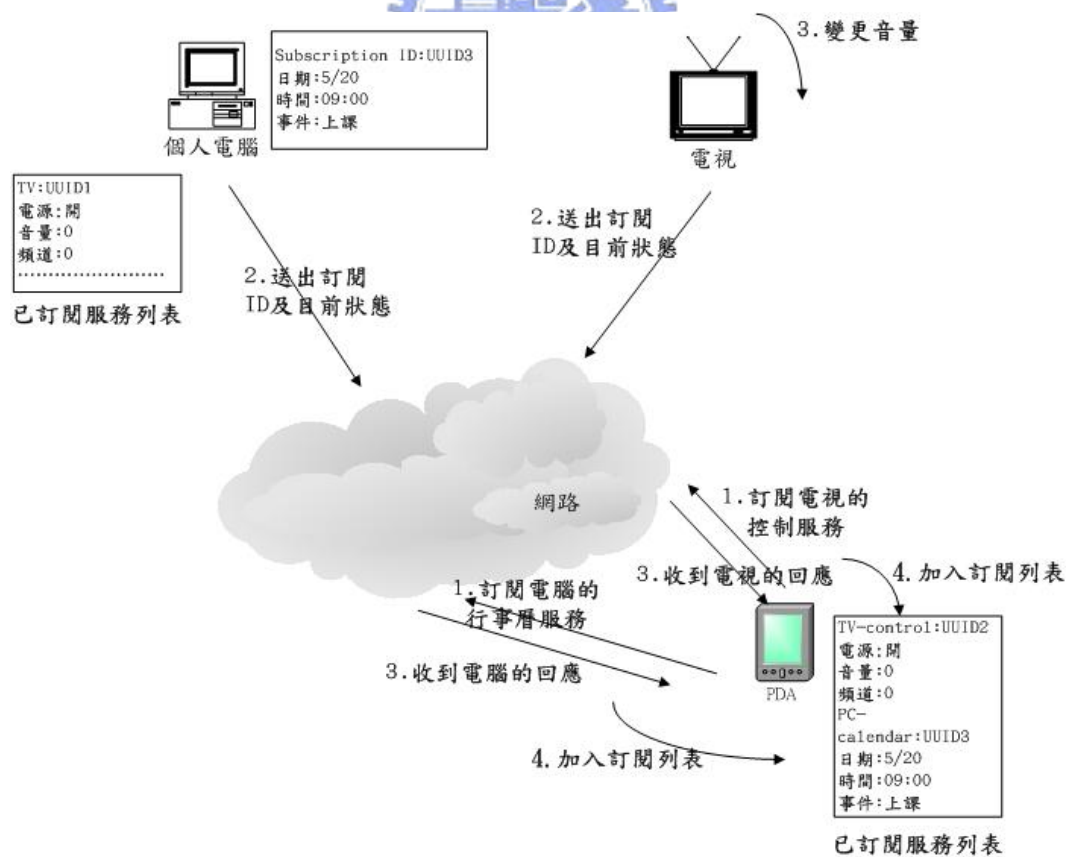
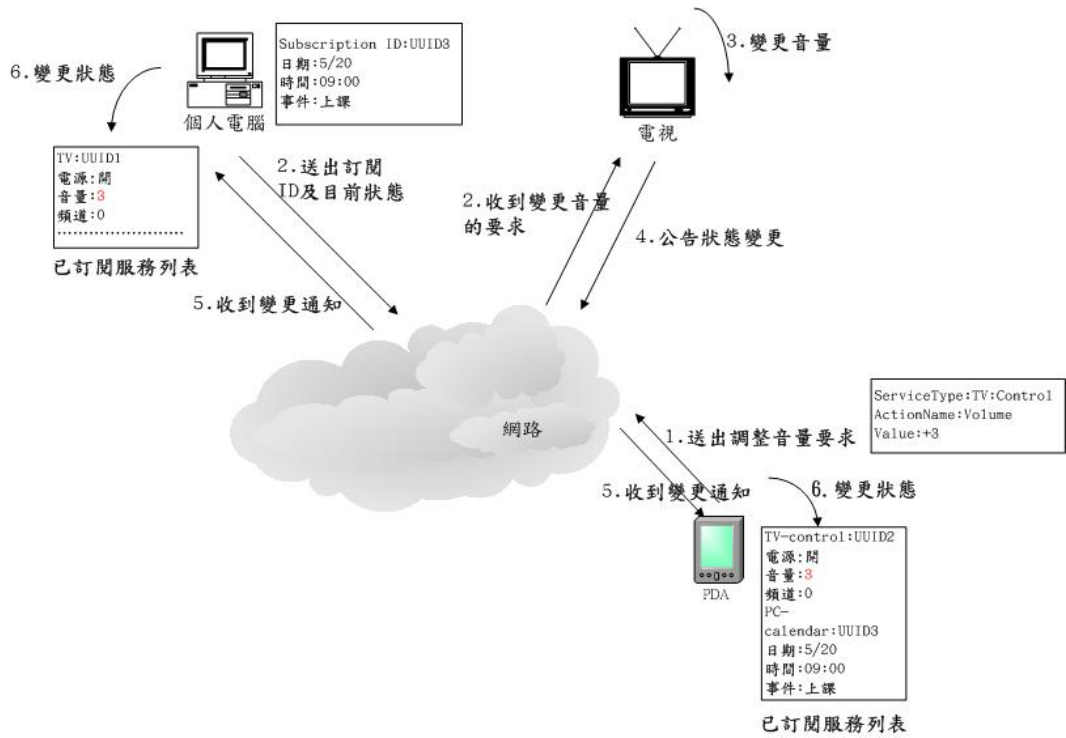


圖 2.15 UPnP 應用範例五



圖

2.16 UPnP 應用範例六

2.5 SLP 網路架構

IETF(Internet Engineering Task Force)所制定的 SLP(Service Location Protocol)只定義了如何找到對方的方法，並未定義遠端呼叫的機制，在提供客戶端使用強大的服務屬性查詢機制。

2.5.1 架構簡介

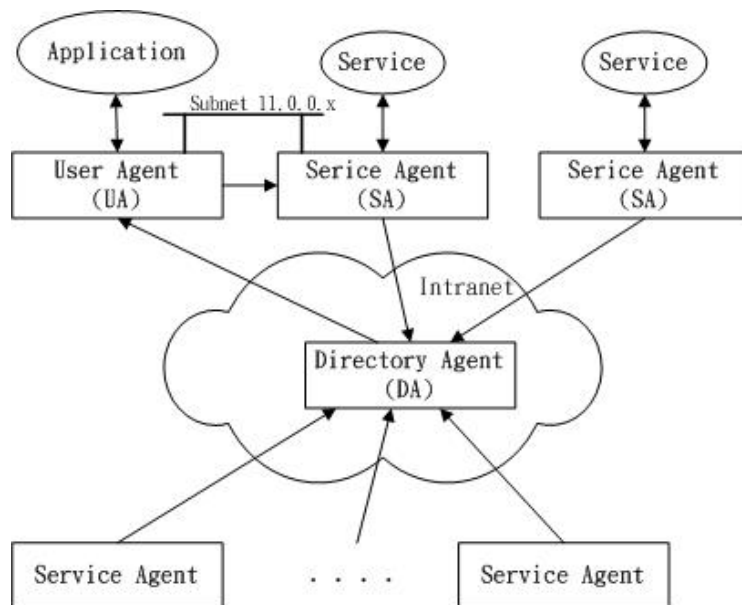


圖 2.17 SLP 網路架構圖

SLP 是 IETF 所製訂的服務搜尋標準，具有非集中式、輕型和延伸性強的特性。SLP 使用 service URL 來存取特定的服務，URL 裡描述了服務的型態、屬性和位置。例如，”service:printer:lpr://hostname”用以表示目前位於 hostname 所提供印表機的列印服務。利用 service URL，使用者可以查詢在網路內有那些可使用的服務，並利用屬性描述，選擇他們所需要的服務。

2.5.2 運作流程

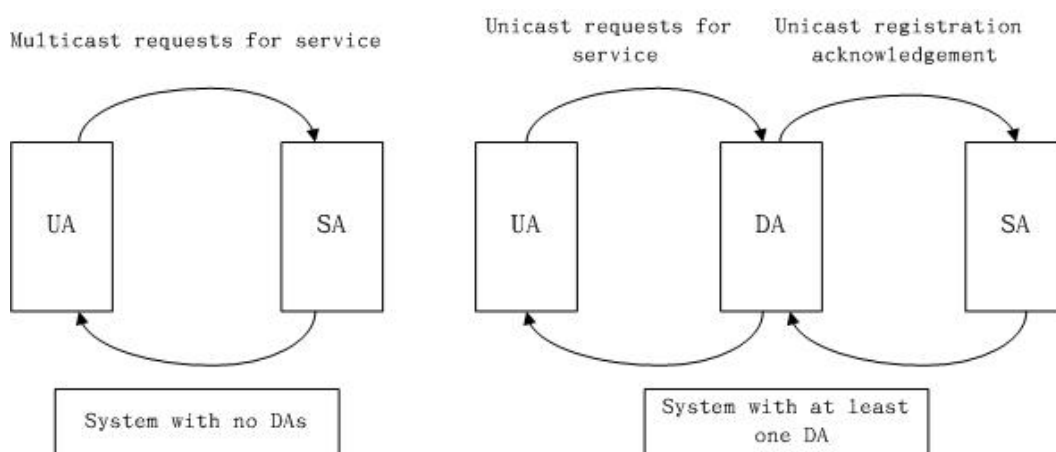


圖 2.18 SLP 網路搜尋流程

在 SLP 網路內主要有三種類型的 Agent：包括 User、Service 和 Directory Agent。UA 是一個軟體實體，主要工作是依據使用者需求，送出服務搜尋的需求到網路。而 SA 則會廣播它所提供的服務。在 SLP 網路下若是採行集中式的架構則會具備 DA，由 DA 記錄 SA 廣播出來的服務，並接受來至 SA 的服務搜尋需求。SA 在廣播它所提供的服務時，若 DA 存在，則會向 DA 註冊 SA 所提供服務的資訊。註冊的訊息也是採行 URL 的格式，包括 SA 的存活時間和服務屬性描述。SA 會週期性向 DA 更新它的註冊資訊；DA 更新完成後會再傳送 ACK 訊息給 SA，通知其更新完成。

當 UA 需要存取服務時，會向 DA 詢問服務的位置。DA 會至回傳符合 UA 需求的服務 URL 給 SA。UA 存得服務 URL 後，UA 可利用服務 URL 進行存取 SA 所提供的服務。在 SLP 的網路環境內，DA 並非必需的，如圖 2.17。在小型的網路環境裡，DA 可能不存在，此時，UA 則利用 multicast，直接傳送服務詢求的訊息至各個 SA。

SLP 支援服務的瀏覽和使用字串屬性表示方式來進行服務查詢；UA 可利用指定屬性來選擇最適當的服務。SLP 准使 UA 屬性使用包括 AND、OR、比較表示和部份字串比較來進行查詢服務。SLP 在服務屬性比對上的功能較 Jini 和 UPnP 來的強大。

2.5.3 SLP 應用範例

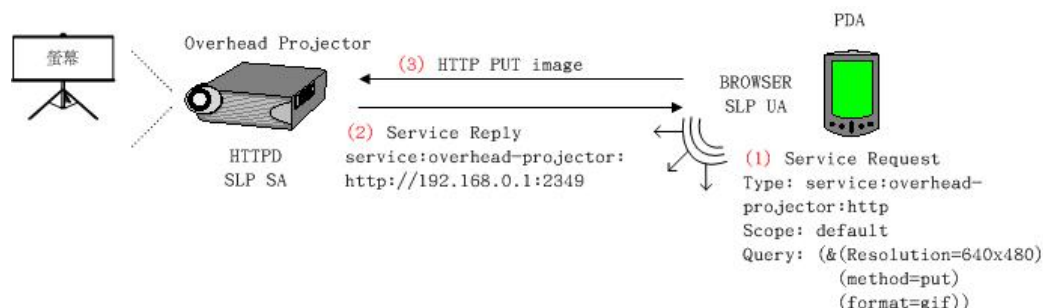


圖 2.19 SLP 搜尋範例

這裡以會議室中使用 PDA 來搜尋投影機為例，說明 SLP 的服務搜尋機制。在投影機本身內含一個 SLP Service Agent(SA)。SA 主要的工作在於聆聽網路上是否有針對投影機的服務型態查尋，並提出需求的使用者；如果有，SA 會回應使用者的需求。

假設使用者要將 PDA 上的一個圖檔經由投影機上呈現，使用者需要解析度 640x480，並且支援 gif 格式的投影機。接著，由 PDA 內含的 SLP User Agent(UA)藉由 multicast 它的需求來尋找符合的投影機。這個需求會被聆聽中的 SA 接到，而 SA 會直接回應訊息給 UA。UA 所需的服務屬性會利用服務型態和字串為主的查詢方式來表達。UA 找到投影機的位置後，再利用 HTTP 上傳圖檔。

2.6 Bluetooth SDP 網路架構

不像 Jini、UPnP、Salutation 或是 SLP，Bluetooth SDP 僅可使用於 Bluetooth 裝置上，SDP 主要是用來解決服務搜尋問題。它並不提供服務的存取、服務仲裁、服務廣播，並且當服務已移除時，也不提供事件機制通知其它網路成員。SDP 支援藉由服務類別、服務屬性進行搜尋和服務瀏覽的功能。當一個 Bluetooth Client 不知道附近其它有什麼 Bluetooth 服務提供時，可利用 SDP 來進行搜尋。SDP 是 Bluetooth profile 定義的一個規格，主要是執行於連接導向轉輸協定 L2CAP 之上。

2.6.1 架構簡介

Bluetooth SDP 提供了一個標準機制，讓 Bluetooth 設備間(peer)可以進行詢問和搜尋機制，Bluetooth 在服務搜尋上只能針對無線環境設備提供。Bluetooth 在搜尋詢問的方式是採行 Peer-to-Peer 的方式，但關於 SDP 是一個 Client/Server 協定。在 SDP 環境內包含客戶端和伺服器端，伺服器端維護一串的服務記錄(Service Records)，服務記錄描述著主機的服務特性。藉由使用 SDP 查詢，客戶端可以瀏覽全部主機上可使用的服務記錄，或由服務記錄取出特定的屬性資料。

在查詢和回覆的訊息格式上，SDP 規格書也定義了標準的方法來描述服務屬性(Service Attribute)。服務屬性只採用<identifier, value>一組的方式來表示。在 1.1 的規格書內定義了一些共同使用的屬性，為了避免碰撞衝突的發生，每個服務都有一個唯一的識別值(UUID)。如果，客戶端已知道某一個正在尋找服務的 UUID，則可直接查尋 SDP Server 取得特定的服務屬性資料。

SDP 提供了一個由別的設備取回服務資訊的機制，但有關於如何呼叫遠端服務則不屬於 SDP 定義的範疇。

2.6.2 協定堆疊

Bluetooth 的協定堆疊的功能簡單描述於表 2.1。由於在本研究僅使用 Bluetooth SDP 來進行服務搜尋的機制。

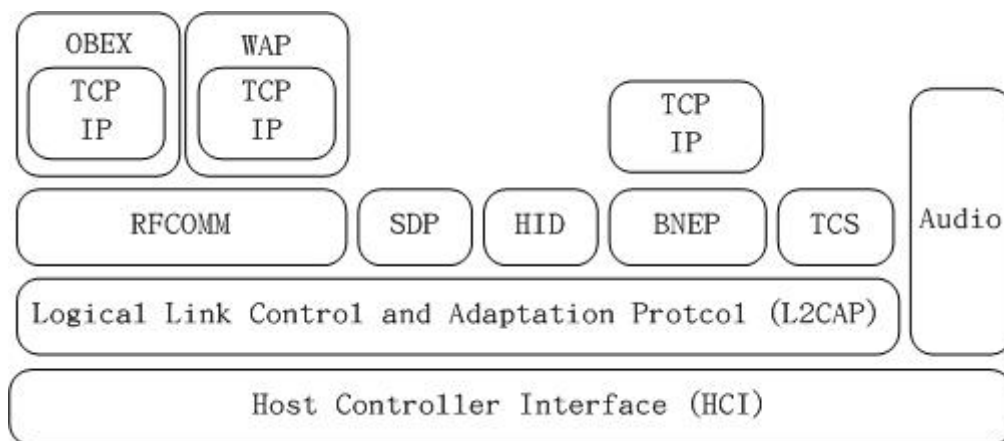


圖 2-20 Bluetooth 協定堆疊

表 2-1 Bluetooth 協定堆疊說明

縮寫	完整名稱	描述
HCI	Host Controller Interface	主要是介於主機 (例如 PC) 和 Bluetooth 控制模組間。
L2CAP	Logical Link Control and Adaptation Protocol	為上層處理所有相關的資料傳輸工作。
SDP	Service Discovery Protocol	處理區域內 Bluetooth 設備間的服務搜尋機制。
RFCOMM	RFCOMM	准許建立虛擬的序列埠和傳送串流資料。
TCS-BIN	Telephony Control Protocol Specification	准許產生音效應用方面的控制訊號。
WAP	Wireless Access Protocol	准許使用 WML 來呈現資料內容。
OBEX	Object Exchange	准許進行物件資料的傳送和接收。
BENP	Bluetooth Network Encapsulation Protocol	封裝來至不同協定的封包資料，轉成 L2CAP 層的封包。
HID	Human Interface Device Protocol	處理輸入設備 (像鍵盤、滑鼠) 的控制訊號和資料。

2.6.3 運作流程

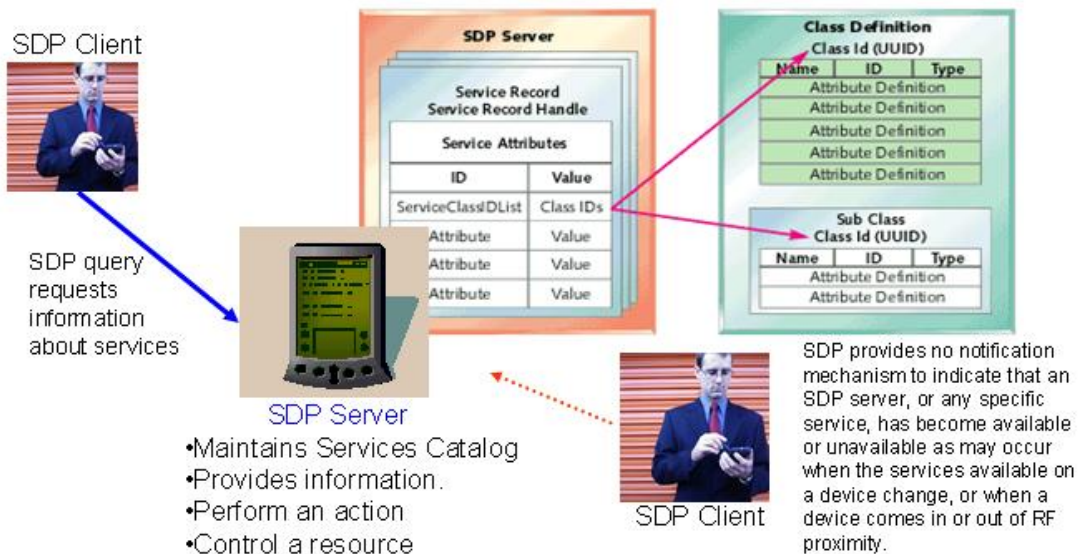


圖 2.21 Bluetooth SDP 服務搜尋流程

SDP Server 記錄關於主機服務的屬性、服務資訊等，SDP Client 可以藉由查詢來瀏覽所有的服務記錄，或是取出特定服務記錄的屬性資料。藉由取得屬性資料可得知關於主機所提供的服務資訊或是主機的特性。如圖 2.21。

2.6.4 Bluetooth SDP 應用範例

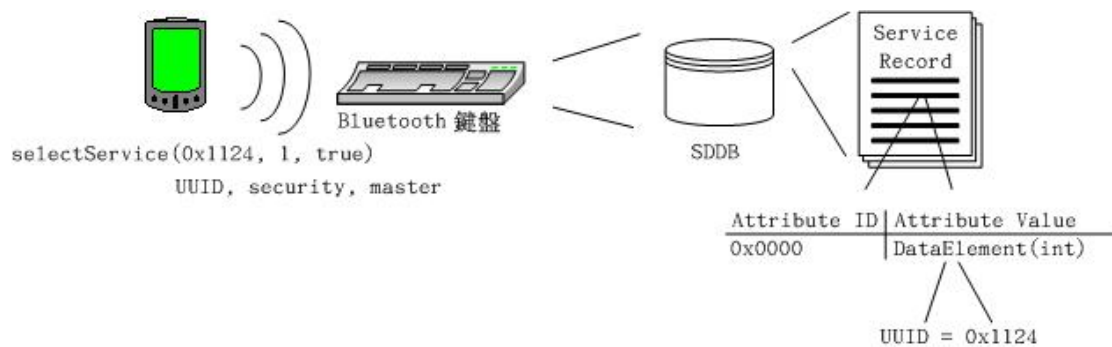


圖 2.22 PDA 使用 SDP 進行搜尋 Bluetooth 鍵盤服務流程

這裡以 PDA 查詢 Bluetooth 鍵盤服務為例，PDA 已知鍵盤服務的 UUID，則可利用 UUID 至 Bluetooth 設備的 SDP Server 查詢，是否有該服務記錄，如果有則表示找到所需的鍵盤服務。

2.7 服務搜尋整合技術分析

在這整理 3 個有關服務搜尋機制整合的研究的，並簡略說明其研究概要，如表 2-2 所示。並於下面各節加以說明其架構。

表 2-2 服務搜尋整合技術研究

Jini Meets UPnP: An Architecture for Jini/UPnP Interoperability(G. G. RichardIII 2003)	提出一個 Framework 用以整合 Jini 和 UPnP 網路，並藉由 virtual service 來相互存取，並減少新的服務加入時程式修改的時間。
SLP-Jini Bridge(Erik Guttman 1999)	提出由 SLP-Jini Bridge 扮演 SLP UA 的角色，當做 SLP/Jini 網路的中介橋樑，使用 HTTP 來下載 client-driver 和 SLP 網路溝通。
Jini Discovers Bluetooth(Vincent Lenders 2002)	使用 Bridge 來轉換 Jini 服務進行 Bluetooth 網路，反之亦同，利用 Bridge 來連接 Bluetooth piconet 和 Jini TCP/IP 網路，並進行服務整合。

2.7.1 Jini 和 UPnP 相互整合架構

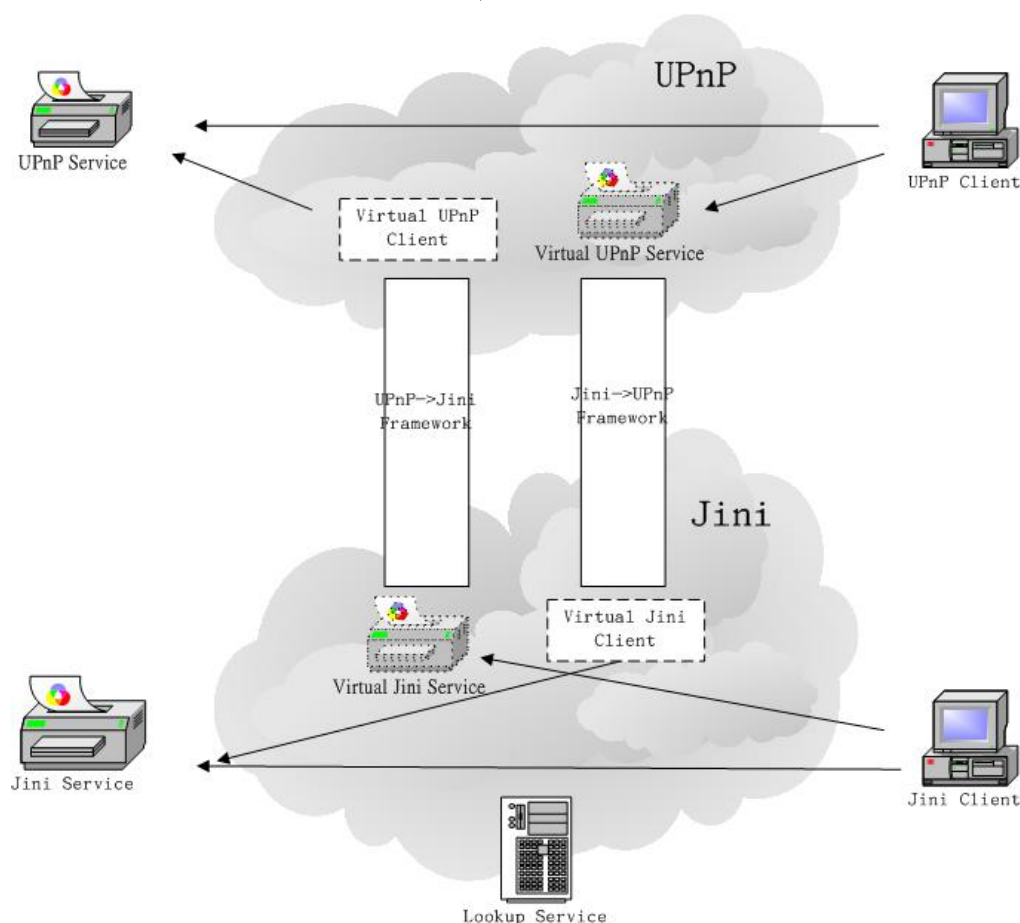


圖 2.23 Jini/UPnP Interoperability Framework 架構圖

這個架構主要是藉由服務的 proxy 整合 Jini 和 UPnP 網路。讓 UPnP 客戶端可以存取 Jini 的服務，Jini 的客戶端亦可以存取 UPnP 服務。在這個 Framework 下，新的服務提供時，需做適度的程式碼撰寫，但 Jini/UPnP 客戶端毋需修改程式。這個 Framework 主要的目的在於儘可能減少因 Jini/UPnP 互相存取而需修改的程式。

要達成這個目標，需藉由欺騙 Jini Client 達成；當它在和 UPnP 服務存取時，讓它以為它是在跟 Jini 服務溝通；而 UPnP Client 的情況也是相同。這個 Framework 會為每個實體 UPnP 服務產生 virtual Jini service，這個觀念在相反方向的亦同(UPnP client 使用 Jini 服務)。這個過程描述於圖 2.23。每個 UPnP 的服務都會由 Framework 自動產生一個 virtual Jini service，並且向 Jini lookup service 註冊這個 virtual Jini service。對每個 Jini 服務，亦會由 Framework 產生一個 virtual UPnP service，並執行 UPnP advertisement，用於通知 UPnP 網路內其它服務或客戶端。這個 virtual service 在執行不同網路的橋接、訊息轉換工作。當 UPnP、Jini 服務離開網路時，這個架構也會自動移除相對應的 virtual service。

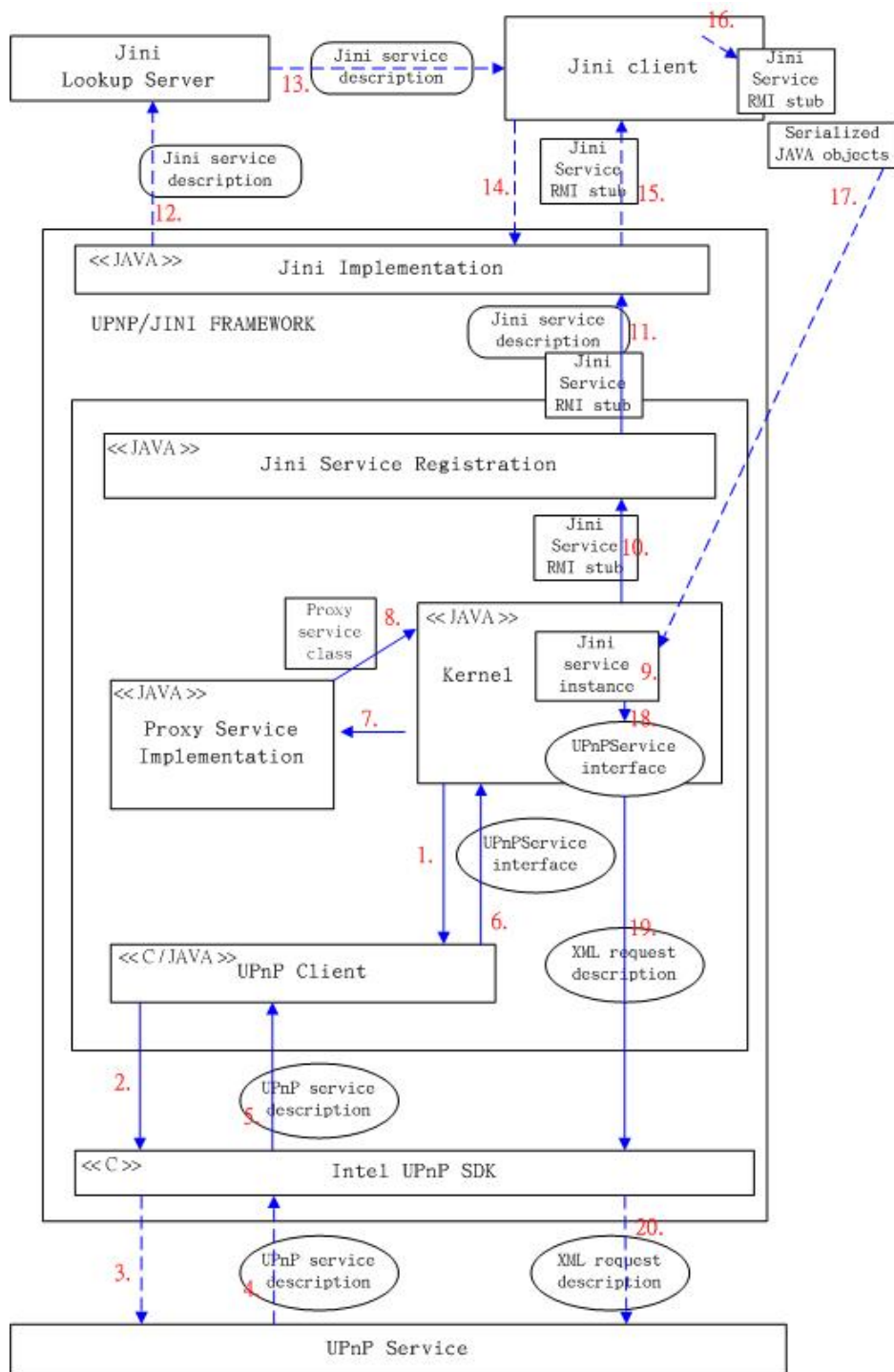


圖 2.24 Jini/UPnP Interoperability Framework 執行流程

在這裡說明 Jini/UPnP Framework 運作流程，如圖 2.24 內所示編號步驟。

1. 首先，由 Kernel 使用 UPnP Client 元件去搜尋可用的服務。在這僅搜尋符合 Proxy 支援的 Service Type。
2. 將 request 轉向給 UPnP SDK，向 UPnP SDK 提出請求。

3. 服務請求訊息由 UPnP SDK 轉給實際的 UPnP 服務。
4. UPnP 服務將自身所提供的服務描述文件回轉給 UPnP SDK。
5. UPnP Client 元件會解析回傳的描述文件，取得 UPnP 服務提供服務的資訊。
6. UPnP Client 元件建立一個 UPnPService interface，用於呼叫 UPnP 服務。並將 interface 傳送給 Kernel。
7. Kernel 配置適當的 Proxy Service 類別，以供 Jini 搜尋服務時使用。
8. 產生出 Proxy Service 類別後，傳回給 Kernel。
9. Kernel 建立 Jini 服務的 proxy instance。
11. 建立 Jini 服務 registration instance。
12. 將 Jini 服務的描述向 lookup service 註冊。
13. 客戶端提出服務需求，並由 lookup service 下載服務的 stub。
- 17-20. 客戶端藉由服務的 stub 進行呼叫，這個呼叫會經由 virtual service 傳送給實體的 UPnP 服務。

2.7.2 SLP、Jini 橋接器設計

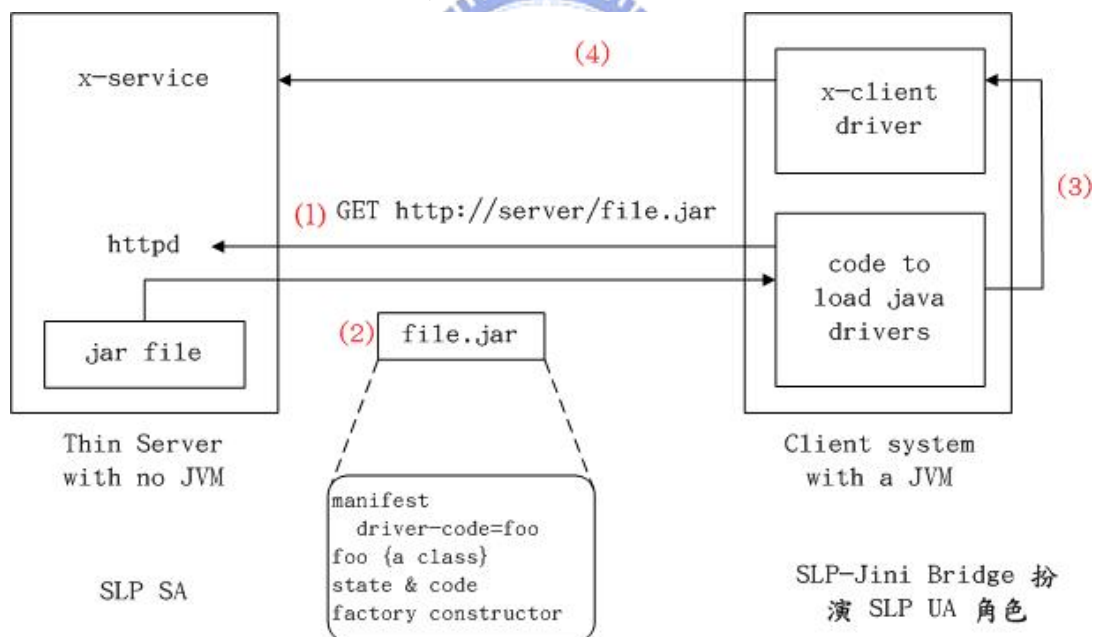


圖 2.25 SLP/Jini 橋接器運作流程

客戶端的 Jini 服務的服務和屬性描述物件的取得，將藉由使用 driver 載入的方式。這種機制可以使不具備 JVM 的服務，藉由提供物件具備 JVM 的客戶端呼叫。圖 2.25 描述一個 driver 載入的動作如何完成。

1. 客戶端首先向 thin server 提出 driver factory 的要求。
2. 客戶端下載這個 jar 檔。

3. 這個 jar 檔包含一個 manifest 檔案，由 manifest 描述 driver factory 類別的名稱。藉由 manifest 的描述，客戶端可啟始這個 driver 物件。
4. 客戶端藉由使用這個 client driver 來和由 thin server 提供的服務溝通。

2.7.3 Jini Discovery Bluetooth

使用 Jini-Bluetooth Bridge 來連接 Jini TCP/IP 網路和 Bluetooth Piconet。這個 Bridge 設備需同時支援 Jini 和 Bluetooth。在 Jini 網路裡，Jini 服務藉由詢問和搜尋 Lookup Server 取得，而在 Bluetooth 設備則提供它本身的 Service Record 資料庫(SDP Server)供查詢。藉由 Bridge 來委任雙邊網路提出來的需求，例如，Bluetooth 客戶端先使用 SDP 向 Bridge 詢問服務，而 Bridge 需同時向 Jini Lookup Server 詢問，再回應給 Bluetooth 客戶端，當 Bluetooth 客戶端要進行存取時，也經 Bridge 代為執行。反方向的 Jini 客戶端要求服務時，也是先詢問 Bridge 來進行。如圖 2.28。

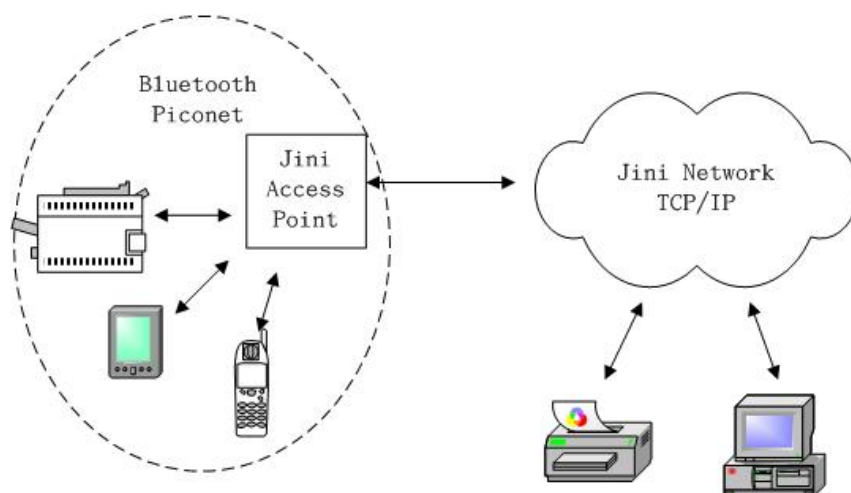


圖 2.26 Jini/Bluetooth 閘道器架構圖

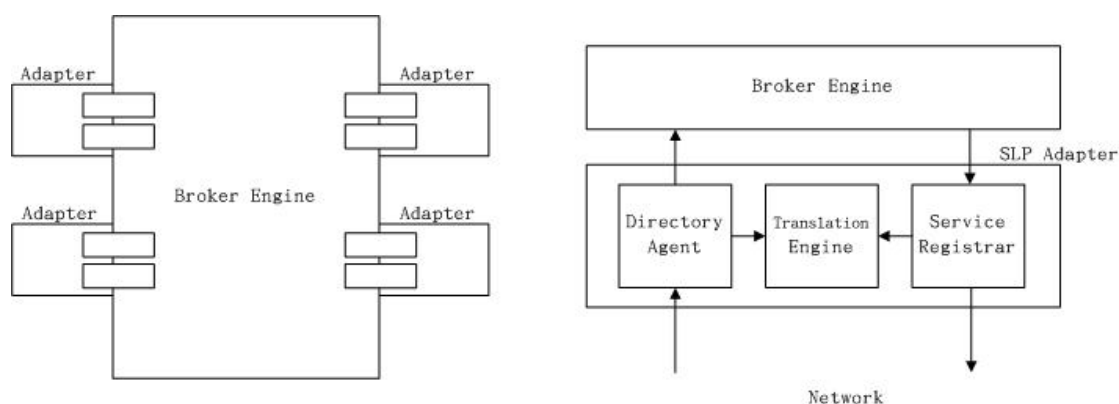


圖 2.27 Bluetooth 協定堆疊

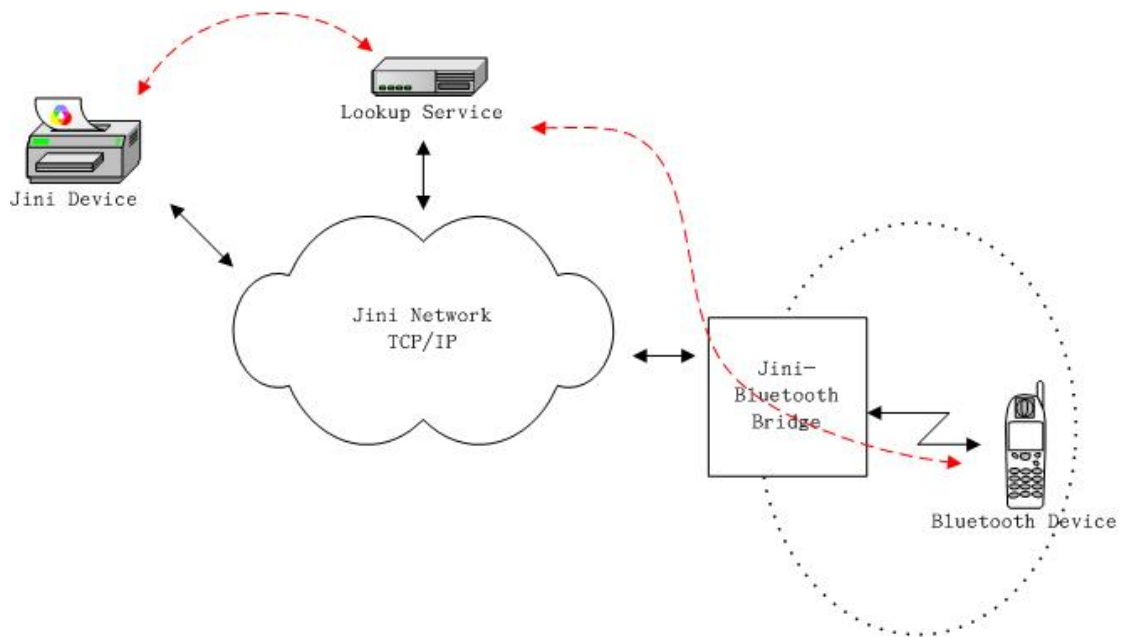


圖 2.28 Jini/Bluetooth 服務存取流程圖

