

目錄

中文摘要.....	II
英文摘要.....	III
誌謝.....	IV
目錄.....	V
表目錄.....	VII
圖目錄.....	VIII
第一章、緒論.....	1
1.1 研究動機與背景.....	1
1.2 研究目標.....	3
1.3 論文架構.....	4
第二章、相關文獻探討.....	5
2.1 問題研究.....	5
2.2 服務搜尋機制.....	5
2.2.1 服務搜尋機制介紹.....	6
2.2.2 服務搜尋機制的特色.....	7
2.2.3 共同特色.....	7
2.3 Jini 網路架構.....	8
2.3.1 架構簡介.....	8
2.3.2 協定堆疊.....	9
2.3.3 運作流程.....	11
2.3.4 Jini 應用範例.....	12
2.4 UPnP 網路架構.....	12
2.4.1 架構簡介.....	13
2.4.2 協定堆疊.....	13
2.4.3 運作流程.....	14
2.4.4 UPnP 應用範例.....	15
2.5 SLP 網路架構.....	19
2.5.1 架構簡介.....	19
2.5.2 運作流程.....	20
2.5.3 SLP 應用範例.....	21
2.6 Bluetooth SDP 網路架構.....	21
2.6.1 架構簡介.....	21
2.6.2 協定堆疊.....	22

2.6.3 運作流程.....	23
2.6.4 Bluetooth SDP 應用範例.....	24
2.7 服務搜尋整合技術分析	24
2.7.1 Jini 和 UPnP 相互整合架構.....	25
2.7.2 SLP、Jini 橋接器設計.....	27
2.7.3 Jini Discovery Bluetooth.....	28
第三章潛在的問題與改進構想.....	30
3.1 服務搜尋協定整合潛在問題.....	30
3.2 網路閘道架構概觀.....	36
第四章、系統設計與實作.....	38
4.1 系統功能分析	38
4.2 系統架構與相關模組.....	39
4.3 Scenario.....	42
4.3.1 註冊流程.....	42
4.3.2 查詢呼叫流程.....	42
4.3.3 事件通知機制.....	43
4.3.4 例外狀況處理.....	44
4.4 系統實作說明.....	45
第五章、系統實驗與效能評估.....	48
5.1 實驗環境介紹	48
5.2 實驗方法.....	53
5.3 效能評估.....	73
第六章、結論與未來展望	76
6.1 研究成果.....	76
6.2 未來展望.....	76
參考文獻	77

表目錄

2.1 Bluetooth 協定堆疊說明.....	23
2.2 服務搜尋整合技術研究.....	24
3.1 SLP、UPnP、SLP、Bluetooth SDP 功能比較表.....	30
3.2 不同服務搜尋機制下，閘道整合解決方案.....	35
4.1 家用閘道器元件功能描述.....	45
4.2 Repository Table schema.....	46
5.1 不同 Domain 下，客戶端進行遠端呼叫的平均回應時間.....	74



圖目錄

1.1 一般家庭網路.....	1
1.2 連接不同 Domain 的 Service Broker.....	2
1.3 連接不同服務搜尋機制的家用網路閘道.....	2
2.1 服務搜尋流程.....	6
2.2 Jini 架構.....	9
2.3 Jini 協定堆疊.....	9
2.4 RMI 機制.....	10
2.5 客戶端和服務提供間 Jini 的運作流程.....	11
2.6 客戶端使用服務提供者服務的流程.....	11
2.7 Jini 搜尋範例.....	12
2.8 UPnP 協定堆疊.....	14
2.9 UPnP 運作流程.....	14
2.10 UPnP 控制流程示意圖.....	16
2.11 UPnP 應用範例一.....	16
2.12 UPnP 應用範例二.....	17
2.13 UPnP 應用範例三.....	17
2.14 UPnP 應用範例四.....	18
2.15 UPnP 應用範例五.....	18
2.16 UPnP 應用範例六.....	19
2.17 SLP 網路架構圖.....	19
2.18 SLP 網路搜尋流程.....	20
2.19 SLP 搜尋範例.....	21
2.20 Bluetooth 協定堆疊.....	22
2.21 Bluetooth SDP 服務搜尋流程.....	23
2.22 PDA 使用 SDP 進行搜尋 Bluetooth 鍵盤服務流程.....	24
2.23 Jini/UPnP Interoperability Framework 架構圖.....	25
2.24 Jini/UPnP Interoperability Framework 執行流程.....	26
2.25 SLP/Jini 橋接器運作流程.....	27
2.26 Jini/Bluetooth 閘道器架構圖.....	28
2.27 Bluetooth 協定堆疊.....	28
2.28 Jini/Bluetooth 服務存取流程圖.....	29
3.1 服務搜尋機制架構.....	31
3.2 閘道服務在不同架構上的處理方式.....	31
3.3 閘道服務進行服務屬性轉換.....	32

3.4 一般分散式元件運作架構和 SOAP 遠端呼叫流程.....	32
3.5 服務搜尋機制 Stub/Skeleton 動態產生情況.....	32
3.6 線上動態產生 Stub/Skeleton.....	33
3.7 使用 bytecode 產生器動態產生 Stub/Skeleton.....	33
3.8 使用閘道服務傳送服務資訊.....	34
3.9 閘道器內部的 Framework 設計.....	34
3.10 閘道器處理型別轉換機制.....	35
3.11 網路閘道設計方法.....	36
3.12 Generic Domain 和不具 Generic Domain 的網路架構.....	37
3.13 使用 Generic Domain 和閘道服務連接不同 Domain.....	37
4.1 以 Jini 做為 Generic Domain，連接不同服務搜尋 Domain.....	39
4.2 Domain Adapter/Translation 功能說明.....	39
4.3 家用網路閘道器主要元件.....	40
4.4 Jini RMI 機制.....	41
4.5 家用閘道器組成元件.....	41
4.6 顯示 Jini 服務至 SLP Adapter 閘道服務註冊/取消流程.....	42
4.7 SLP 客戶端呼叫 UPnP 服務流程.....	43
4.8 家用閘道器主要元件.....	45
5.1 家庭網路實驗環境.....	48
5.2 Jini Lookup Service 啟動程式.....	49
5.3 啟動 SLP Daemon.....	49
5.4 啟動 MySQL.....	50
5.5 家用閘道器啟動程式.....	50
5.6 啟動 Jini Service Agent.....	51
5.7 啟動 SLP Service Agent.....	51
5.8 啟動 UPnP 閘道服務.....	52
5.9 啟動 SLP 閘道服務.....	52
5.10 支援 Jini 網路的果汁機資訊家電模擬程式.....	54
5.11 支援 UPnP 網路的電燈泡資訊家電模擬程式.....	54
5.12 支援 SLP 網路的印表機資訊家電模擬程式.....	55
5.13 UPnP 客戶端模擬程式.....	56
5.14 UPnP 客戶端模擬程式搜尋方式.....	56
5.15 UPnP 客戶端模擬程式顯示目前網路上提供的服務.....	57
5.16 UPnP 客戶端模擬程式的服務資訊.....	58
5.17 UPnP 客戶端呼叫 UPnP 服務流程說明.....	59
5.18 UPnP 電燈泡模擬程式初始狀態.....	59

5.19 UPnP 客戶端模擬程式查詢到電燈泡服務.....	60
5.20 UPnP 電燈泡所提供的 method 資訊.....	60
5.21 UPnP 電燈泡 SSDP 提供的服務資訊.....	61
5.22 呼叫 SetTarget method 設定電源狀態.....	61
5.23 將 UPnP 電燈泡電源開啟.....	62
5.24 進行遠端呼叫後，UPnP 電燈泡設備會改變自己的狀態為開啟.....	62
5.25 UPnP 客戶端呼叫 Jini 服務流程說明.....	62
5.26 Jini 果汁機模擬程式初始狀態.....	63
5.27 UPnP 客戶端模擬程式查詢到 UPnP 閘道服務下轉換的 Jini 果汁機服務....	64
5.28 呼叫 setSpeed method 設定果汁機搖動速度.....	64
5.29 將 Jini 果汁機搖動速度設為 2.....	65
5.30 進行遠端呼叫後，Jini 果汁機設備搖動速度會改變為 2.....	65
5.31 呼叫 getSpeed method 取得果汁機目的搖動速度.....	65
5.32 進行遠端呼叫後，傳回目前 Jini 果汁機設備搖動速度為 2.....	66
5.33 UPnP 客戶端呼叫 SLP 服務流程說明.....	66
5.34 SLP 印表機模擬程式初始狀態.....	67
5.35 UPnP 客戶端模擬程式查詢到 UPnP 閘道服務下轉換的 SLP 印表機服務...	67
5.36 呼叫 setPage method 設定印表機列印紙張數目.....	68
5.37 將 SLP 印表機列印紙張設為 2.....	68
5.38 進行遠端呼叫後，SLP 印表機設備列印紙張數目會改變為 2.....	69
5.39 Jini 客戶端呼叫 UPnP 服務流程說明.....	69
5.40 UPnP 電燈泡模擬程式初始狀態.....	70
5.41 遠端呼叫，設定電燈泡電源為開啟.....	70
5.42 進行遠端呼叫後，SLP 電燈泡設備電源會改變開啟.....	72
5.43 Jini 客戶端呼叫 SLP 服務流程說明.....	72
5.44 SLP SA1 服務啟動.....	73
5.45 Jini 客戶端經由 SLP 閘道服務呼叫 SLP SA1 回傳結果.....	73
5.46 不同 Domain 下，客戶端進行遠端呼叫的平均回應時間.....	74

第一章、緒論

1.1 研究動機與背景

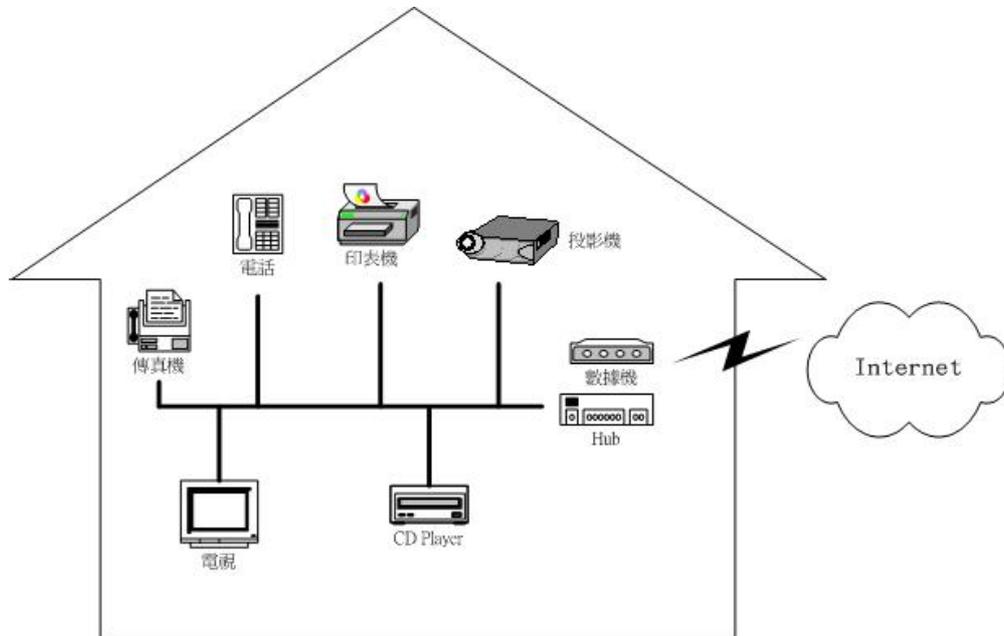


圖 1.1 一般家庭網路

普及運算近年來蓬勃發展。普及運算的目的：讓使用者可以簡單的使用周遭的日常家電產品，例如：冰箱、微波爐、喇叭等具有上網能力的日常用品。在這樣的環境裡，要如何尋找可用的物品以及要如何去存取這些日常用品是很重要的問題。在普及運算環境裡的物品及服務可以根據某些屬性分類，例如：依照位置、服務種類、性質等分類存取。

行動計算和普及運算的出現，將家用網路帶入了一個更加動態的環境。由於在動態的環境下，當使用者要存取網路上所提供的服務時，依據服務的位置資訊來存取，便成了一個問題，因此，該服務可能已移除於網路，或是新的服務剛加入網路內。因此，使用，依據所需使用的服務特性來查詢服務變成目前的趨勢，而不在使用固定的網路位址來靜態存取。因此，發展出許多服務搜尋機制(Service Discovery Solution)，包括 Bluetooth、Jini、Universal Plug and Play、Salutation 和 Service Location Protocol 等，但是不同的服務搜尋機制間並無法相互溝通、協同合作。但是，在行動和普及運算的環境裡，可相互整合的 Federating Service Discovery Domain 是迫切需要的。

在家用網路的動態網路環境裡，往往不能假設每個網路服務存在於固定的位置。如果提供服務的設備並非移動裝置，但是使用者仍可能將它移往別處；即使提供服務的裝置不會移動，但仍可能有新的服務在不同的網路環境加入，並較原有的設備提供更適切的服務。因此，需要一種依據使用者的需求來取得

服務的架構。這種架構稱作服務搜尋機制(Service Discovery Solutions)。這些也是家庭網路中資訊家電使用的存取協定。

目前的服務搜尋機制包括 Bluetooth、Jini、Universal Plug and Play、Salutation 和 Service Location Protocol，但彼此間並無法相互溝通。因此，如果在一個網路環境內同時具備多種服務搜尋機制的設備，它們所提供的服務便無法彼此互相存取，自然也不知道對方的存在(invisible)。尤其在行動和普及運算的情況時，ad-hoc 網路裡通常是存在著多種不同的 Service Discovery Domain。

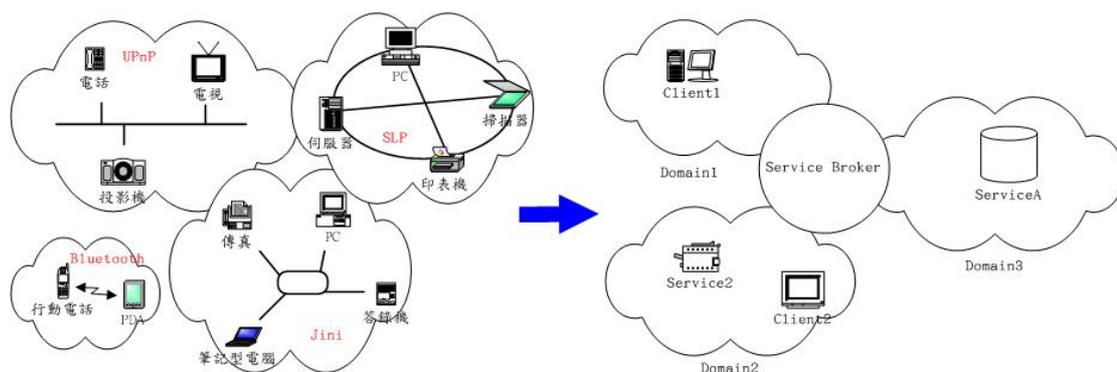


圖 1.2 連接不同 Domain 的 Service Broker

最近的研究已提出 Service Discovery Domain Federation 的需要性，希望藉由跨不同的 Service Discovery Domain，藉由 Federation 來連接不同的 Domain 的服務和使用者；使用者可以跨越不同的 Domain 來存取服務，藉由中介軟體元件 Service Broker 來連接不同的 Domain 成為單一的 Logical Domain。圖 1.2 說明了 Service Broker 和各 Service Discovery Domain 間的關係。

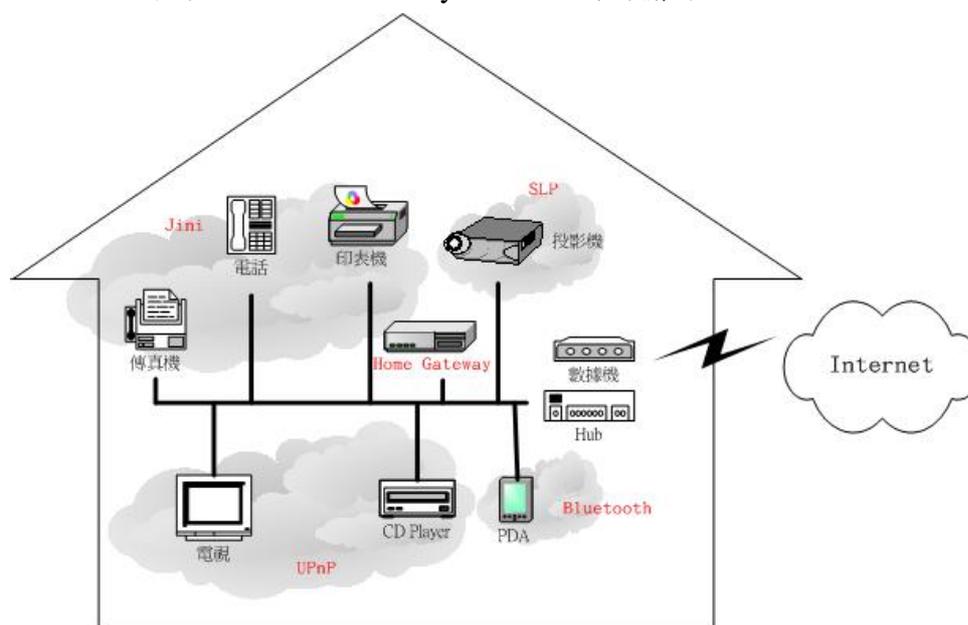


圖 1.3 連接不同服務搜尋機制的家用網路開道

在本篇研究裡，將針對家庭網路環境，設計與實作一個智慧型家用網路開道器，來連接不同的 Service Discovery Domain，以解決目前各 Domain 間無法互相存取彼此服務的問題。

1.2 研究目標

在這個步入後PC時代的時期，家庭網路的程式設計是未來不可或缺的技术，而如何能克服設備之間不同協定之互通性問題，眾多設備的管理與整合問題，如何能符合嵌入式系統之動態載入執行條件，如何達到設備間彼此互相發覺的問題等，都需要加以整合不同的技術，發展整合式家庭網路服務。

根據家庭網路的需求，本研究目的在建立一個智慧型的家用開道服務環境，在此一環境下，每一個資訊家電都能具有以下能力：

- (1)服務搜尋機制與註冊的能力
- (2)提供服務的能力
- (3)自動加入任何一個社群的能力
- (4)使用服務的能力

這些在家庭網路內的不同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，包括了找到彼此的方法，了解對方的能力以及使用對方資源的方法等等。在這篇研究裡，以目前最常用的四種服務搜尋機制(Jini、UPnP、SLP、Bluetooth SDP)做探討。

藉由這些資訊家電服務搜尋機制，加以整合並設計智慧型家用開道服務，這個家用開道服務可達成下列目的：

一、提出功能性整合之資訊家電開道服務

藉由家用開道服務，可以整合多樣化之資訊家電功能服務，協助使用者可跨越不同的服務網路，存取該設備的服務，而不會因協定不同，而無法使用。

二、無縫式服務整合環境

藉由家用開道服務，原有的設備無需做任何更新，便可在開道服務的協助下存取不同網路的服務，而新加入的設備也不需要做任何更改。

三、彈性、可延伸的家用開道服務

家用開道服務提供 Plug-In 的機制，未來需要加入新的服務搜尋機制時，可依照開道 interface 設計，加以延伸，並不影響其它原有功能，提供可擴充的機制。

1.3 論文架構

本論文分為六個章節，各章節內容說明如下：

第一章、緒論：述說本論文研究背景、動機與目地、並闡述本研究的問題與研究上的限制及定義和本論文的架構。

第二章、相關文獻研究探討：包含有家庭網路的介紹與本研究中的核心技術。了解服務搜尋機制Jini、UPnP、SLP、Bluetooth SDP基本架構、運作原理，並以範例說明。

第三章、潛在的問題與改進構想：說明目前服務搜尋機制在整合時面臨的問題，如何解決，並提出改進的構想和模型，以求解決各中介模型間的互相通訊問題。

第四章、系統設計與實作：實作第三章所提出的模型，並實作智慧型式家用開道器。

第五章、效能評估與分析：針對實作所得的系統進行使用性分析。

第六章、結論與未來展望：跟據本論文的研究與結果進行總結，並提出未來的研究方向



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。

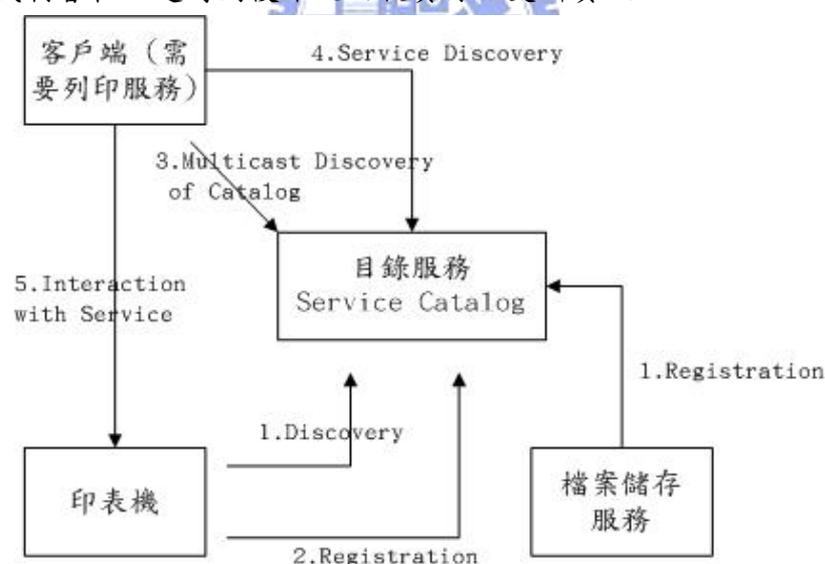


圖 2.1 服務搜尋流程。

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。

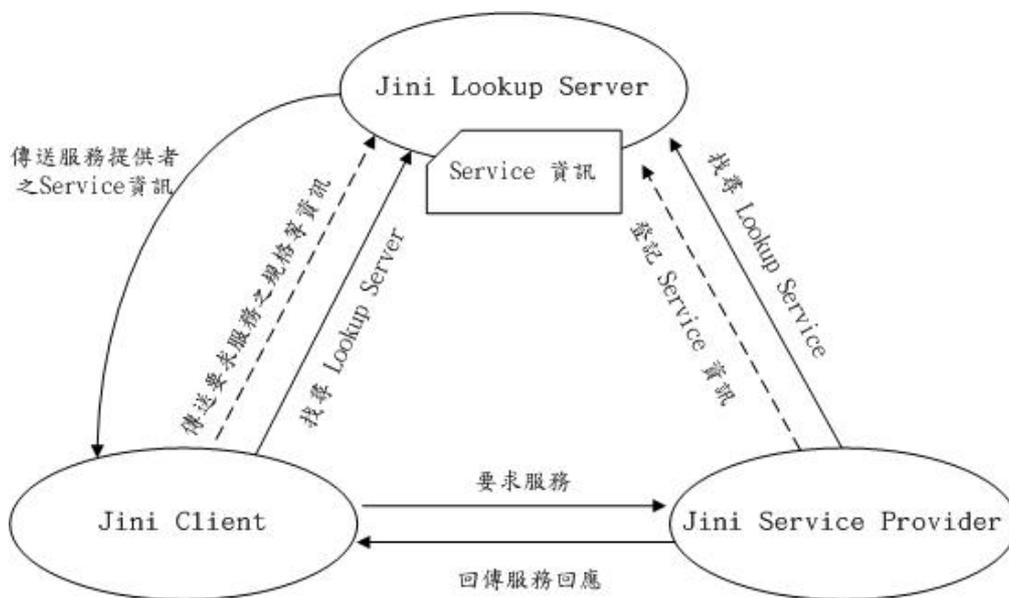


圖 2.2 Jini 架構

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

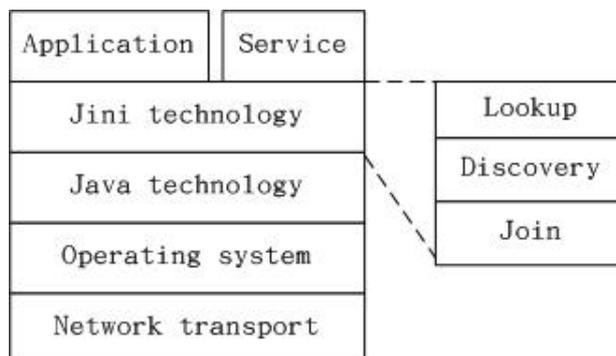


圖 2.3 Jini 協定堆疊

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。

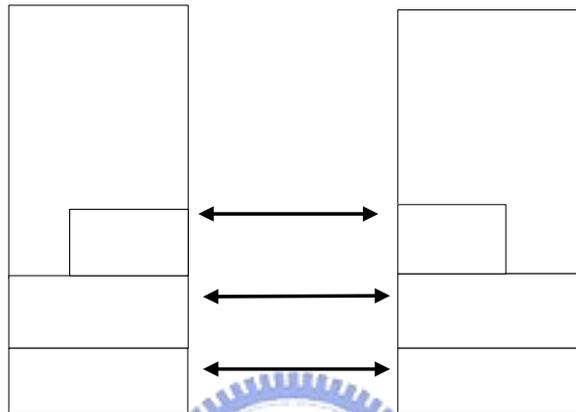


圖 2.4 RMI 機制

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。