

逢 甲 大 學

資訊工程學系專題報告

TAPI 應用 - 插斷式視訊電話

學生：方繼輝(四甲)

徐麒翔(四甲)

指導教授：李維斌 老師

中華民國 九十二年十二月

目 錄

圖表目錄.....	IV
摘要.....	VI
第一章 導論.....	1
1.1 製作動機與目的.....	2
1.2 系統簡介.....	2
1.3 環境需求.....	3
第二章 相關技術簡介.....	4
2.1 TAPI介紹.....	7
2.1.1 TAPI與TSP.....	7
2.1.2 TAPI3.0.....	8
2.1.3 TAPI的運作.....	13
2.1.4 TAPI 3.0 中的IP 多點傳送會議.....	15
2.1.5 H.323.....	20
2.2 COM的介紹.....	26
2.2.1 COM的基本認知.....	26
2.2.2 COM的運作方式.....	27
2.2.3 COM介面.....	29
2.2.4 COM基礎介面的介紹.....	32

2.3 與TAPI相容的語音MODEM	33
第三章 系統架構.....	38
3.1 系統需求.....	39
3.2 系統流程圖.....	40
第四章 結論.....	49
4.1 軟體功能介紹.....	50
4.2 個人心得.....	53
4.3 應用與展望.....	55
4.4 TAPI 本身限制及優缺點.....	55
4.5 曾經遭遇困難的解決方法及探討.....	56
4.6 工作分配.....	58
參考文獻.....	59
附錄 1 TAPI 架構的總覽.....	60
附錄 2 所需要的函式庫.....	61
附錄 3 錯誤代碼.....	62

圖表目錄

表 2.0-1 TAPI 版本的演進.....	6
圖 2.1.1-1 TAPI 與 TSPI 的系統架構圖.....	7
圖 2.1.2-1 TAPI 與傳統電話服務的架構圖.....	9
圖 2.1.2-2 TAPI 的架構圖	10
圖 2.1.2-3 TAPI 的物件關係	11
圖 2.1.2-4 呼叫與呼叫中心位址關係圖	13
圖 2.1.3-1 使用者和核心模式元件的DirectShow 篩選器圖表範例	15
圖 2.1.4-1 網路拓樸(以傳送者的視點).....	16
圖 2.1.4-2 實際的網路拓樸	17
圖 2.1.4-3 使用擴展樹來實行多點傳播	18
圖 2.1.4-4 IP 多點傳送的會議架構.....	19
圖 2.1.5-1 H.323 的架構	20
圖 2.1.5-2 H.323 的元件	22
圖 2.1.5-3 H.323 的 TSP 架構	23
圖 2.1.5-4 Alice 註冊並更新她的 IP 位址.....	24
圖 2.1.5-5 John 查詢 Alice 的位址.....	24
圖 2.1.5-6 協商完成會議開始	25
圖 2.2.3-1 COM 介面和 Vtable 的關係.....	29

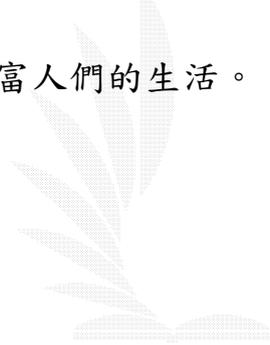
圖 2.2.3-2 COM 介面的取得方式.....	30
圖 2.2.3-3 新舊 COM 介面版本的處理方式.....	31
圖 2.3-1 Win9X 及 2K/XP 下的語音裝置位址	34
圖 2.3-2 如何更新你數據機的驅動程式 PartI	35
圖 2.3-3 如何更新你數據機的驅動程式 PartII	36
圖 3.2-1 系統應用流程圖	40
圖 3.2-2 TAPI Application 與電話架構示意圖	41
圖 3.2-3 Initial TAPI Object.....	42
圖 3.2-4 初始化 Event 及 Register Filter	43
圖 3.2-5 建立連線及撥號.....	44
圖 3.2-6 接聽電話.....	45
圖 3.2-7 斷線處理.....	46
圖 3.2-8 接聽電話 event 資料流程圖	47
圖 3.2.9 接聽電話 USE-CASE 圖型	48
圖 4.1-1 等待接聽電話的圖片	50
圖 4.1-2 電話訊息來電中	51
圖 4.1-3 通話中的圖片	51
圖 4.1-4 撥號中的圖片	52
圖 4.1-5 撥號部份的通話中畫面	52
附錄一 TAPI 架構總覽	60

摘要

TAPI3.0 是微軟用於整合電信與網路通訊的介面，之前的版本因為偏重於電信方面的功能，而導致整合效果不佳，不過在推出新的版本後(TAPI3.X)，就大幅改善了，它是以 COM 為基礎所設計的，避免了程式設計師直接去控制底端的東西，加速了程式的開發時間，同時也更完整的提供網路通訊方面的功能，這跟 IP PHONE 能夠在近期蓬勃發展，有著莫大的關係，就讓我們拭目以待 TAPI 的大放異彩。

我們的目的是利用 TAPI，來提供使用者更便利的環境，讓人類更貼近未來的生活，能夠讓使用者在聽著音樂時，有訊號進來時，插斷音樂的訊號，改由本程式發出聲音，提醒使用者接聽電話。這樣的話使用者不用急急忙忙去接電話，而可以舒舒服服躺在椅子或床上繼續做事情。

利用一點點的創意，來豐富人們的生活。



第一章 導論

- 1.1 製作動機與目的
- 1.2 系統簡介
- 1.3 環境需求

第一章

1.1 製作動機與目的

對於要升大三的我們，對著專題有著無限夢想但卻不知從何下手，從一開始的資料庫，到設計一個遊戲，經歷幾番討論之下，發現都有種不滿意的感覺，設計遊戲的美工、音效、和劇情佔一大部分，所能表現程式設計的地方太過於狹小。

在尋求李老師的意見後，他提議我們可以試試用 TAPI 的東西，來製作以電腦來模擬類似電話答錄機的軟體。在使用者正在用無線的耳機和麥克風聽 mp3 或是音樂時，可以暫時的離開工作區去作其他事例如躺在床上閱讀。如果有電話或訊號進來的時候，TAPI 的應用程式能暫停 mp3 或音樂的撥放，而自動的接聽電話，這樣就不用在忙著去接電話囉，而可以舒舒服服躺在床上，繼續做自己的事。

我們覺得這樣可行，便依此為藍本，來發展我們的專題。再稍微加上一些自己的想法。

1.2 系統簡介

我們是 Visual C++6.0 來撰寫此程式，以 TAPI3.0 的規格來達成所需要的傳輸介面，在我們的系統中，我們規劃下面幾種功能

- (1) 網路電話功能：
- (2) 視訊傳遞
- (3) 電話監聽及撥打功能
- (4) 當使用者在聽音樂或 MP3 的時候，當有訊號進來時，能夠把 MP3 關掉以便接受此次通話。

1.3 環境需求

軟體需求：

Windows2000/Windows XP 以上的OS
提供視訊功能的 Driver

硬體需求：

CPU：基本上可以正常執行Windows2000 & Windows XP 即可。

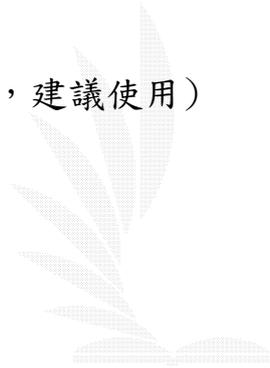
RAM：建議128MB 以上或256MB。

與TAPI相容且支援語音的MODEM

Speaker(喇叭)

MicroPhone(麥克風)

WebCamara(網眼)(非必要，建議使用)



第二章 相關技術簡介

2.1 TAPI介紹

2.1.1 TAPI與TSP

2.1.2 TAPI3.0

2.2 COM的介紹

2.2.1 COM的基本認知

2.2.2 COM的運作方式

2.2.3 COM介面

2.2.4 COM基礎介面的介紹

2.3 與TAPI相容的語音MODEM

第二章 相關技術簡介

前言

TAPI 3.0 是集合傳統式PSTN 電話服務和IP 電話服務的漸進式應用程式介面(API)。IP 電話服務可讓公司和個人降低現有服務成本(例如聲音和廣播電視)，同時增加傳輸方法，以納入現代視訊會議、應用程式共用和白板工具。

過去，公司部署個別的網路來處理傳統的聲音、資料和視訊傳輸。每個網路都有不同的傳輸要求，安裝、維護和重新組態這些網路的花費很大。而且，由於這些網路在實體上截然不同，即便可能，整合也很困難，以致限制其潛在的用途。

IP 電話服務藉由指定每個聲音、視訊和資料的公用傳輸、IP，將這三種網路混合成一個。結果使網路更易於管理，降低支援費用，形成新型合作工具並提高了生產力。

IP 電話服務的可能應用包括遠端交換、即時文件合作、遠距教學、人員訓練、視訊會議、視訊郵件和可視需要取得的視訊。

TAPI 的歷史演進及搭配的作業系統可由圖2-1所示。由表中可以看出TAPI 的發展可區為兩個階段，區分點介於TAPI 2.1 版及TAPI 3.0 版之間。主要的差別在從TAPI 3.0 版後開始提供網際網路電信服務。

TAPI 版本	要點及特性
TAPI 1.3	1. Microsoft 及 Intel 於 1994 年提出 2. 附加於 Windows 3.1
TAPI 1.4	1. 隨 Windows 95 提出(1995 第四季) 2. 提供 32 位元的應用程式發展環境
TAPI 2.0	1. 隨 Windows NT 4.0 提出(1996 第三季) 2. 更新部分功能，提供話務中心介面 3. 提供 UniCode 環境
TAPI 2.1	1. 可附加於 Windows 95 及 Windows NT 4.0 之，但 Windows 95 只可以執行客戶端(cline)，伺服器端(sever)必須執行於 Windows NT 4.0 Severler 2. 真正提供主從式架構 (Cline-Sever Architecture)
TAPI 3.0	1. 隨 Windows 2000 上市 2. 提供網際網路電信服務

表 2.0-1 TAPI 版本的演進

2.1 TAPI的介紹

2.1.1 TAPI與TSP

簡單的說TAPI 與TSPI 的架構關係是：TSPI 是介於TAPI與硬體間的一個標準溝通介面。對上能夠提供TAPI所要求的服務，而對下則是對硬體發出命令，以期能夠達成應用程式透過TAPI所要求的服務。如此一來，對於應用程式而言，就不需要擔心底層所用的硬體是何廠商所提供的，只商有提供符合TAPI版本的TSP即可。而對於硬體廠商而言，只要提供相對應的電話傳輸驅動程式版本(TSP)，即可符合所有的TAPI 應用程式，而且不需考慮到資源的管理工作，因為若是硬體廠商自行提供驅動程式時，必須考慮到多個應用程式都用到同一個硬體設備時應如何處理，何者具有優先權，但TSPI 中並不需要考慮這些問題，為TAPI 會統一管理這些硬體裝置，決定應式使用裝置的優先順序。

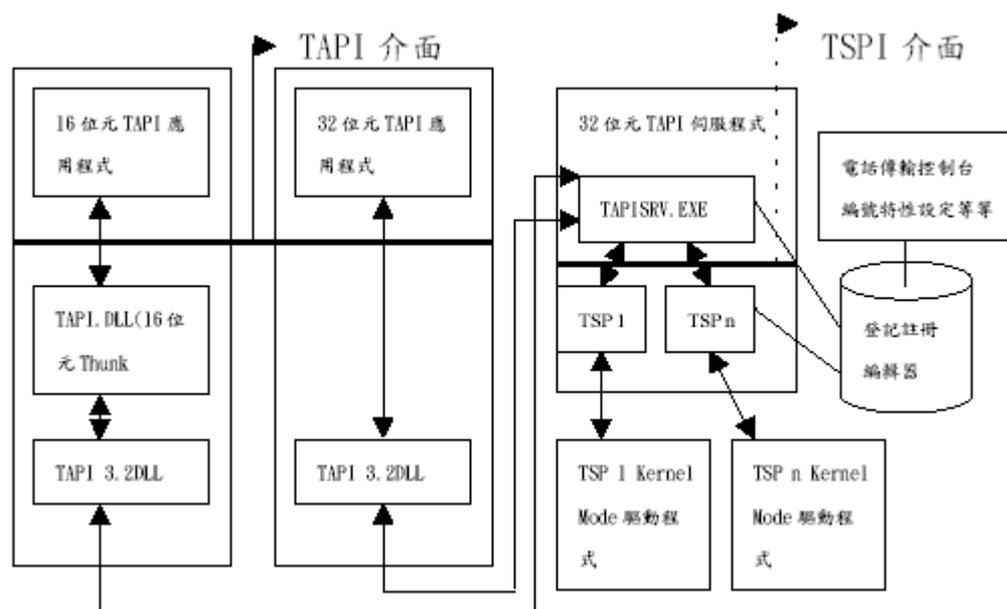


圖 2.1.1-1 TAPI與TSPI的系統架構圖

由圖中可看出當一個應用程式使用到TAPI時，不管是16位元32位元的應用程式均會呼叫到TAPI的重態連結程式庫—TAPI32.DLL(若為16位元的應用程式會先透過一中間轉換媒介”Thunk”，將16位元轉換成32位元再呼叫TAPI32.DLL)。而TAPI的動態連結程式庫會將應用程式所發出的服務要求傳到TAPI伺服器程式—TAPISRV.EXE，由TAPI伺服器程式解譯其中的要求，再決定往下向TSP要求服務或直接處理回應，在這裡TAPI即是扮演統一管理的角色。而在登記註冊編輯器(registry)中則記錄有各家硬體驅動圖式及其硬體的相關資訊，如硬體A卡提供有四條通訊線、硬體B卡提供有一備通訊線，則在登記註冊編輯器中就會有每一條通訊線的資訊，包含在系統中的硬體裝置編號、特性等相關訊息。

因為應用程式在進入TAPI使用環境後就會對所有的裝置搜尋過一次，瞭解所有的裝置的特性，並採用合適的裝置。這樣的機制對於硬體提供者而言是相當的方便，因為不需要額做資源管理的工作，TAPI伺服器程式便會做妥善的管理相對應裝置的使用權限及使用狀況。

2.1.2 TAPI 3.0

隨著電話服務和呼叫控制在桌上型電腦中的普及，需要用一般電話服務介面讓應用程式可以存取任意電腦上可用的所有電話服務選項。呼叫上的媒體或資料也必須以標準方式供應用程式使用。TAPI 3.0 提供了簡單而普通的方法，能夠結合兩部或多部電腦，並存取這種結合所涵蓋的任何媒體資料流。它摘錄了呼叫控制功能，讓不同而看似不相容的傳輸通訊協定提供應用程式使用的公用介面。

當公司開始從昂貴而缺乏彈性的電路交換公用電話網路轉移至有智慧、彈性而且便宜的IP 網路時，IP 電話服務能夠自若地面對爆炸性的成長。Microsoft 預料到這個趨勢，所以建立了強健的電腦電話服務基礎結構TAPI。目前TAPI 的第三個主要版本，適合快速、簡單地部署IP 電話服務應用程式。見下圖

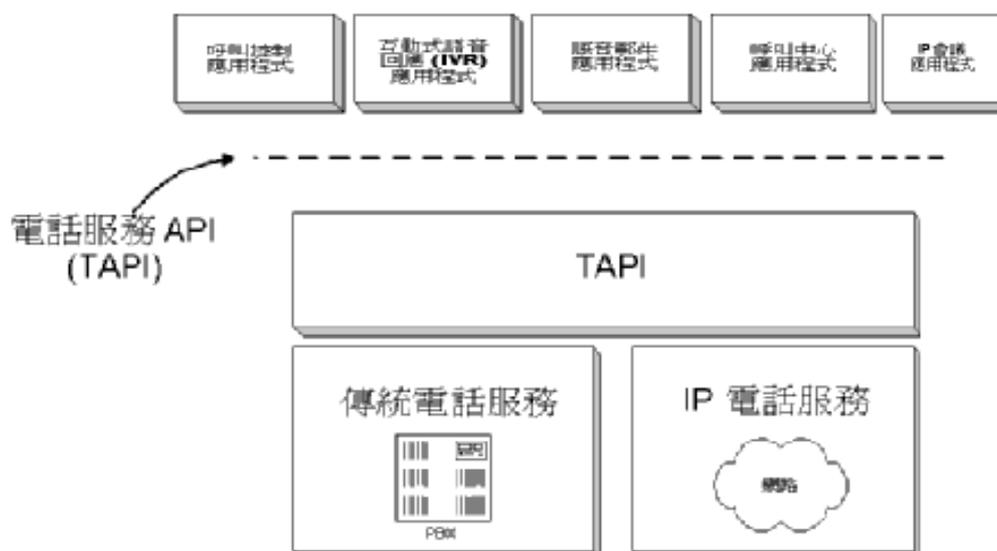


圖 2.1.2-1 TAPI 與傳統電話服務的架構圖

TAPI 3.0 整合多媒體資料流控制和傳統電話服務。此外，它是從TAPI 2.1 API 到COM 模式的改進，可用任何語言編寫TAPI 應用程

式，例如C/C++ 或MicrosoftR Visual BasicR 。

除了支援傳統的電話服務提供程式，TAPI 3.0 還支援標準H. 323 會議和IP 多點傳送會議。TAPI 3.0 使用WindowsR 2000 Active Directory 服務來簡化公司內的部署，支援服務品質(QoS) 功能，提高會議品質，使網路易於管理。

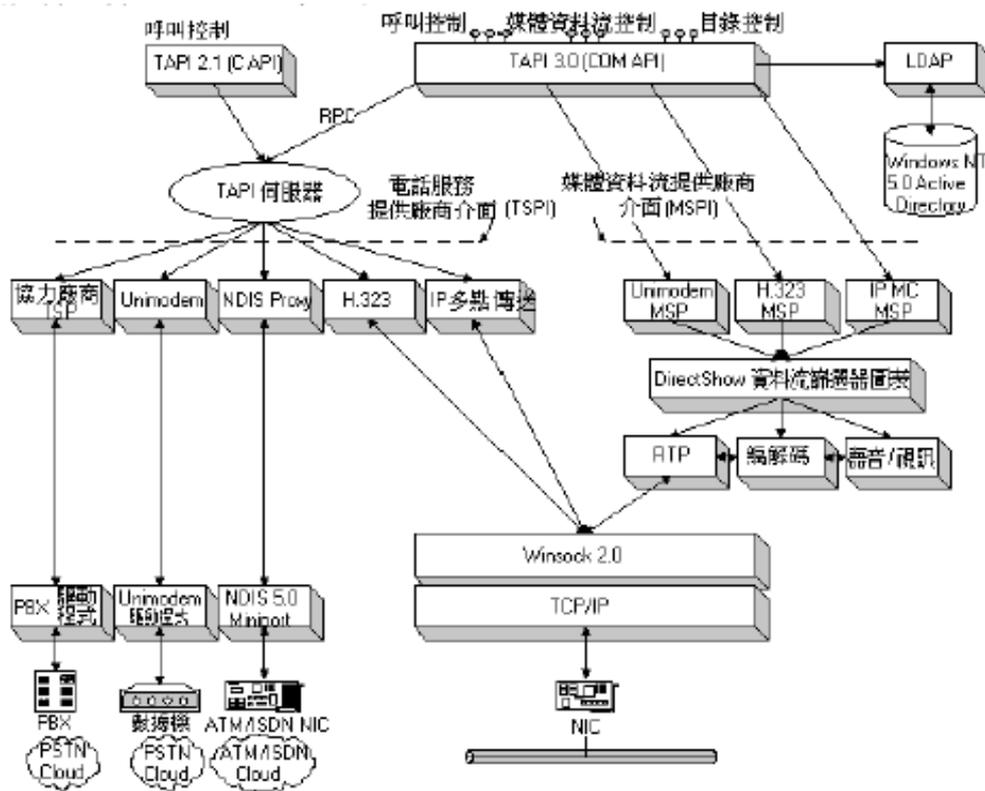


圖 2.1.2-2 TAPI 的架構圖

TAPI 3.0 有四個主要元件：

- TAPI 3.0 COM API
- TAPI 伺服器
- 電話服務提供程式
- 媒體資料流提供程式

與TAPI 2.1 相比，TAPI 3.0 API 是以COM 物件來建置的。將TAPI 移到COM 模式可讓TAPI 功能的元件升級，也可讓開發者使用任何語言來編寫啟動

TAPI 的應用程式。TAPI 伺服器程序(TAPISRV.EXE) 從TAPI 3.0 和TAPI 2.1 摘錄TSPI (TAPI 服務提供程式介面)，讓TAPI 2.1 「電話服務提供程式」搭配TAPI 3.0 使用，以維護TAPI 的內部狀態。「電話服務提供程式(TSP)」負責將TAPI 中與通訊協定無關的呼叫模式融入通訊協定專用的呼叫控制機制中。TAPI 3.0 提供TAPI 2.1TSP 回溯相容性。兩個IP 電話服務提供程式(及其相關的MSP)在出廠時就提供TAPI 3.0：H.323 TSP 和IP 多點傳送會議TSP 將在後面討論。

TAPI 3.0 提供統一的方式從呼叫中的存取媒體資料流，此方式支援DirectShow™ API 作為主要的媒體資料流處理程式。TAPI 「媒體資料流提供程式(MSP)」為特殊的TSP 的建置了DirectShow 介面，而任何利用DirectShow 資料流的電話服務都需要這個介面。一般的資料流由應用程式處理。

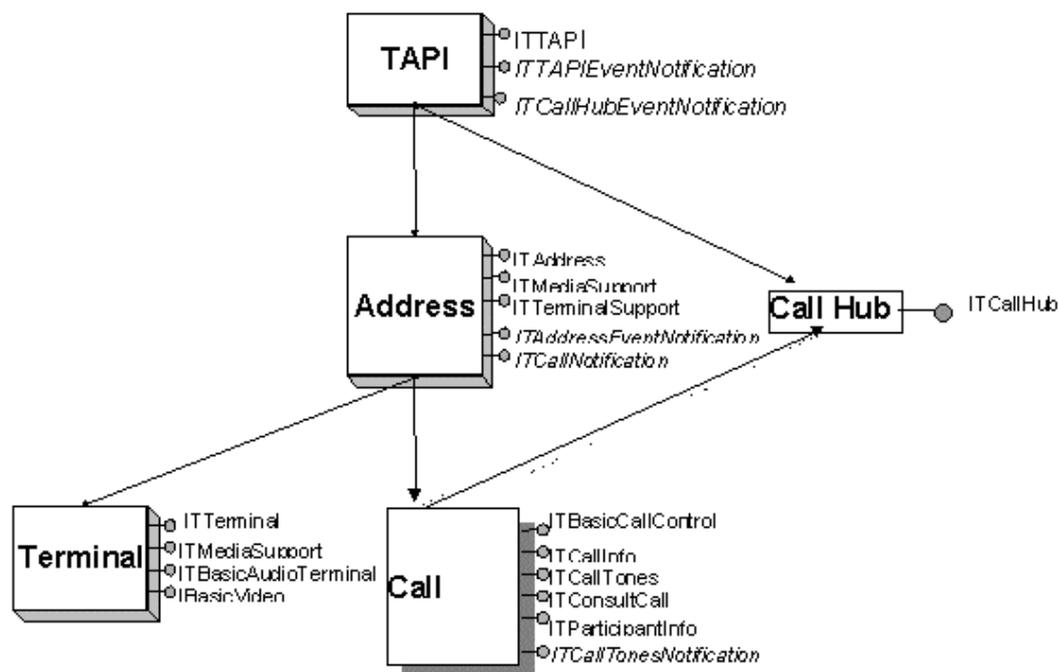


圖 2.1.2-3 TAPI 的物件關係

TAPI 3.0 API 中包含五個物件：

- TAPI
- 位址
- 終端機
- 呼叫
- 呼叫中心

TAPI 物件是TAPI 3.0 的應用程式入口。這個物件代表本機電腦可以存取的所有電話服務的資源，並容許應用程式行列出所有本機和遠端的位址。

位址物件代表呼叫來源或目的地。而從這個物件可以找到位址功能(例如媒體和終端機支援)。應用程式可以在「位址」物件等候呼叫，或從「位址」物件建立撥出呼叫物件。終端機物件代表在連線終點或起點的接收器或轉換器。

「終端機」物件可以映射到作為人機介面的硬體，例如電話或麥克風，但也可以是一個檔案，或是可接受輸入或建立輸出的其它任何裝置。

「呼叫」物件代表本機位址和一或多個其它位址之間的位址連接。(這個連接可以是直接，或是透過「呼叫中心」連接的)。你可以將「呼叫」物件想像成電話呼叫的第一組檢視。所有呼叫控制都透過「呼叫」物件來完成。「呼叫中心」的每個成員都有一個呼叫物件。

呼叫中心物件代表一組相關的呼叫。「呼叫中心」物件不能由應用程式直接建立，它是在透過TAPI 3.0 接收撥入的呼叫時間接建立的。使用者可用「呼叫中心」物件，在一個呼叫或會議中列舉其它的與會者，而且如果有足夠的許可權，也可以(由於COM 的位置獨立特性)在與這些使用者相關的遠端「呼叫」物件上執行呼叫控制。

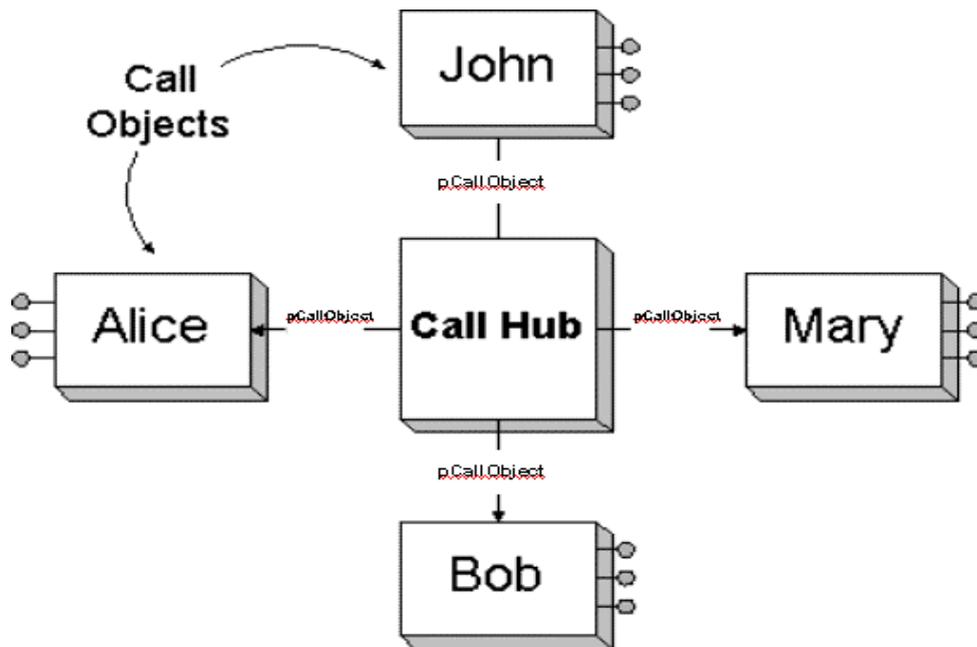


圖 2.1.2-4 呼叫與呼叫中心位址關係圖

2.1.3 TAPI 的運作

發出呼叫

1. 建立和啟動TAPI 物件。
2. 使用TAPI 物件列舉電腦上所有可用的「位址」物件(例如網路卡、資料機和ISDN 線路)。
3. 列舉各「位址」物件所支援的位址類型(例如電話號碼、IP 位址等等)。
4. 根據適當媒體(語音、視訊等)和位址類型的支援查詢來選擇「位址」物件。
5. 使用「位址」物件的建立呼叫方法來建立與特定位址相關的「呼叫」物件。
6. 在「呼叫」物件上選擇適當的終端機。
7. 呼叫「呼叫」物件的結合方法以找到呼叫。

應答呼叫

1. 建立並啟動TAPI 物件。
2. 使用TAPI 物件列舉電腦上所有可用的「位址」物件(例如網

路卡、資料機和ISDN 線路)。

3. 列舉各「位址」物件所支援的位址類型(例如電話號碼、IP 位址等等)。
4. 根據適當媒體(語音、視訊等)和位址類型的支援查詢來選擇「位址」物件。
5. 以適當的「位址」物件在特定媒體類型中登錄有興趣的物件。
6. 以「位址」物件登錄呼叫事件處理程式(亦即建置ITCallNotification 介面)。
7. TAPI 透過ITCallNotification 通知應用程式有新的呼叫，然後建立「呼叫」物件。
8. 在「呼叫」物件上選擇適當的終端機。
9. 呼叫「呼叫」物件的結合方法以找出呼叫。
10. 呼叫「呼叫」物件的結合方法以應答呼叫。

為了有效控制和操作媒體資料流，Windows R 作業系統提供可擴展的構架，稱為DirectShow。DirectShow 透過公開的COM 介面提供TAPI 3.0 統一的資料流控制。

DirectShow 的中心是稱為篩選器的可插入元件的模組化系統，它安排在稱為篩選器圖表的組態中。稱為篩選器圖表管理員的元件監視這些篩選器的結合，並控制資料流的資料流動。每個篩選器的功能由一些稱為PIN 的特殊COM 介面來說明。每個PIN 執行個體都能消滅或產生資料流資料，例如數位語音。

COM 物件通常暴露於使用者模式的程式下，DirectShow 資料流結構包含Windows 驅動程式模式的擴展，這個模式可在裝置驅動程式層直接結合媒體資料流。下圖顯示簡單的PSTN 至IP 橋接器。來自ISDN 線路的64 Kbps 聲音資料流壓縮成G.723 語音資料流，然後送到RTP 負載處理程式，再傳送到網路上。

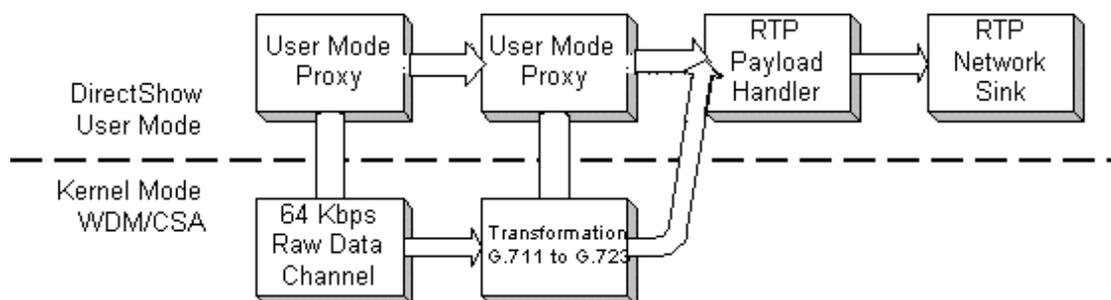


圖 2.1.3-1使用者和核心模式元件的DirectShow 篩選器圖表範例

DirectShow 網路濾鏡擴展了電腦在 IP 連線方面的串流架構。即時系統協定 (RTP)，是設計來處理有關透過不連結的網路來負載即時性的資料，傳遞 TAPI 媒體資料流而且提供適當的時間標記資訊。TAPI 3.0 包含了 RTP 核心網路濾鏡。

TAPI 3.0 利用這樣的科技來展現一個萬用的存取方法，有關在多媒體呼叫下的媒體資料流。應用程式可以藉著相符合的濾鏡圖形來管理這些資料流，它們也可以在多方的傳呼中連接這些資料流，

2.1.4 TAPI 3.0 中的IP 多點傳送會議

IP 多點傳送是IP 的一種延伸，可進行有效率的群組傳輸。IP 多點傳送是應對輕型、大小可調適的會議解決方案需要而產生的，它可以解決在帶位址的最佳資料網上作即時傳輸時的相關問題。使用IP 多點傳送還有許多優點：調適性、容錯、強健且易於安裝。

IP 多點傳送會議模式包含下列重要功能：

- 從會議加入和刪除成員時不需要全員協調。
- 要送資料到多點傳送群組時，使用者只需把資料送到多點傳送IP 位址中的一個，而不需要瞭解群組中的其他使用者。
- 要接收資料時，使用者只須在支援多點傳送路由器的特定多點傳送IP 位址登錄自己感興趣的項目，而不需要瞭解群組中的其他使用者。
- 路由器會對使用者隱藏多點傳送建置的詳細資訊。

傳統連接導向的會議會遇到許多問題：

- 使用者複雜性：使用者必須知道想要交談的每個使用者的位置，這會限制其大小的調適性和容錯功能，並增加使用者參與和退出會議的困難。
- 浪費頻寬：當某個使用者想把資料傳播給 n 個使用者時，必須透過 n 個連接，如下圖所示：

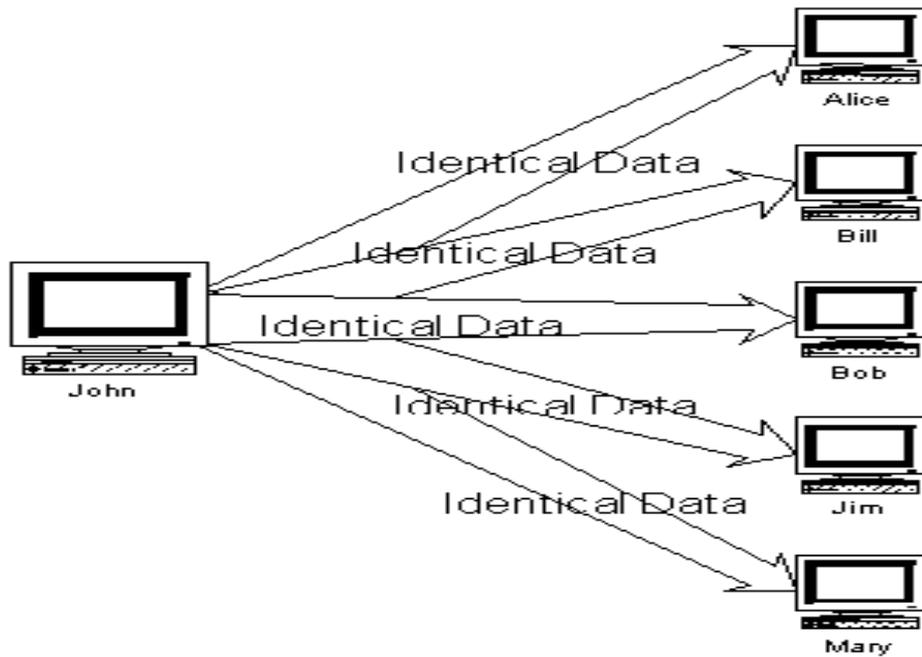


圖 2.1.4-1 網路拓樸(以傳送者的視點)

如果多方會議中的所有使用者都傳送資料，所需的總頻寬會以與會者數量的平方增加，而可能導致嚴重的調適性問題。IP 多點傳送利用實際網路拓樸的優勢來消除相同傳輸連結上的額外資料的傳輸

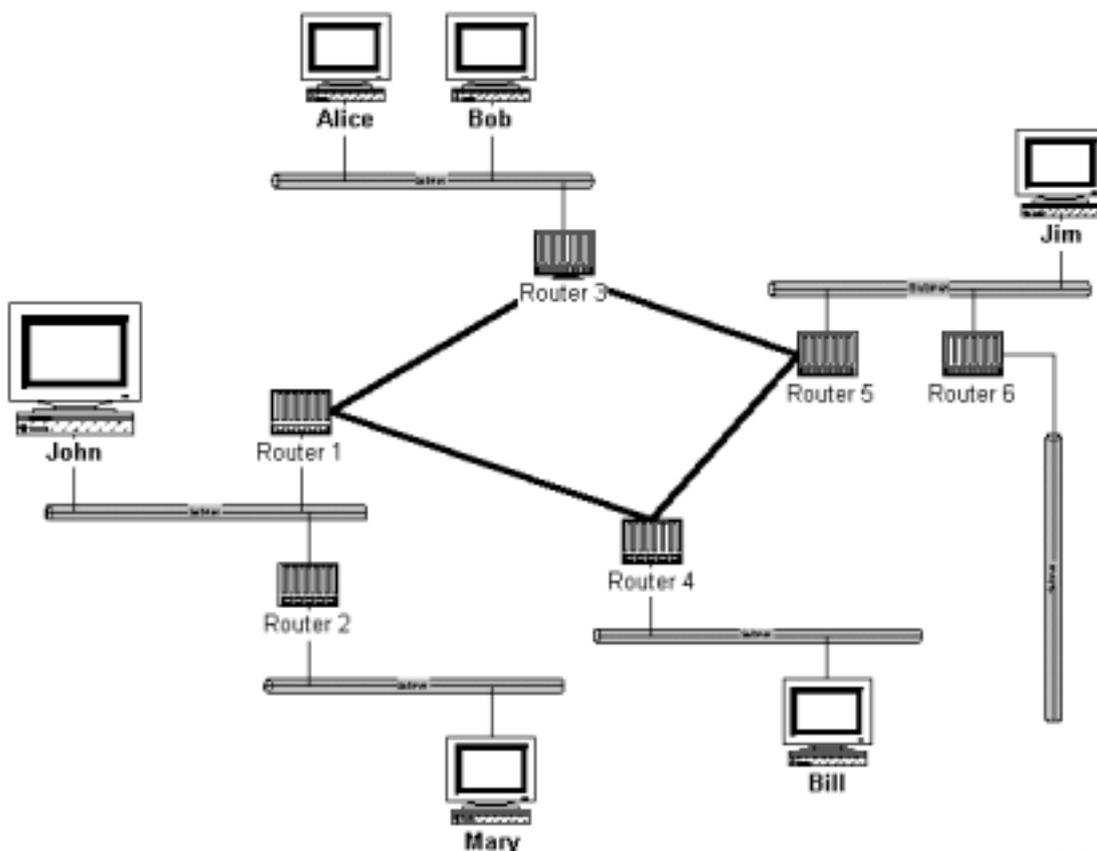


圖 2.1.4-2 實際的網路拓模

IP 多點傳送建置的是輕型、工作階段傳輸模式，這種模式給會議使用者的負擔相當小。利用IP 多點傳送時，使用者只需將一份資訊副本傳送到可送達所有收件者的一個群組IP 位址即可。IP 多點傳送的設計會隨與會者人數的增加而調適，使用者的增加並不會增加對應數量的頻寬。多點傳送還能大大降低傳送伺服器的負載。藉由擴展樹形的建立，IP 多點傳送可以有效的傳送這些一對多的資料流。在這樹型中，一個路由器到其它任何一個路由器都只有一條路徑。只在當路徑分叉時，資料流才需要複製：

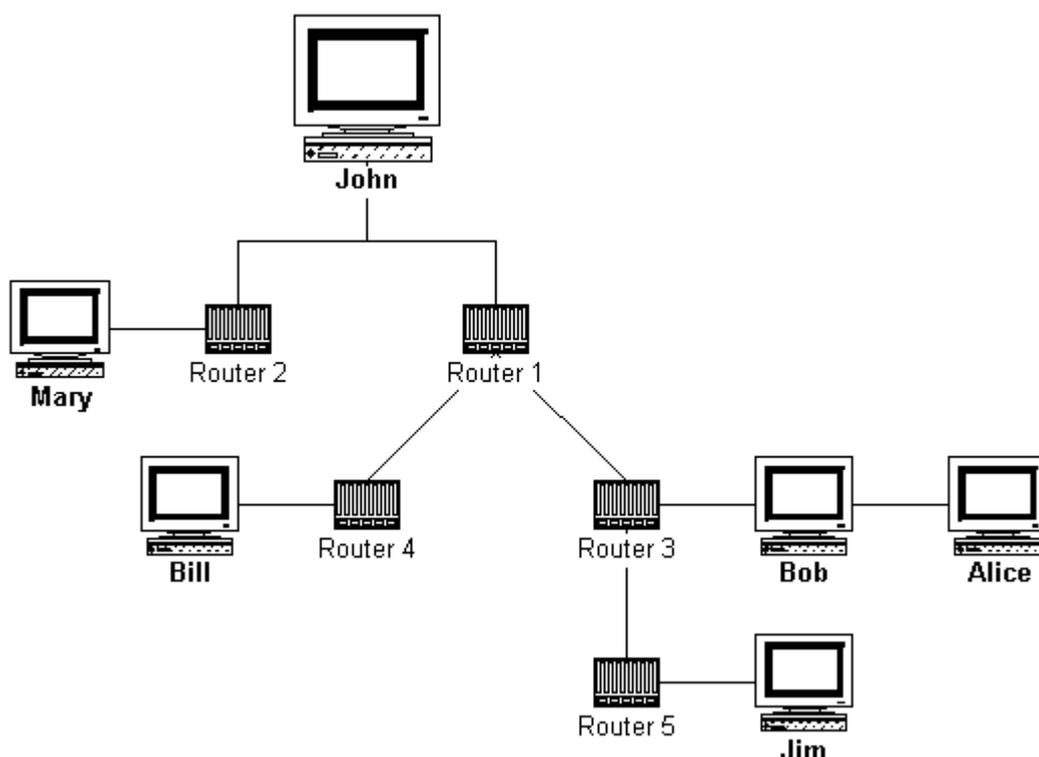


圖 2.1.4-3 使用擴展樹來實行多點傳播

如果沒有多點傳送，相同的資料就必須多次在網路上傳輸、每位收件者傳輸一次或傳播給網路上的每個人，這會消耗不必要的頻寬和處理。

IP 多點傳送使用D 級IP 位址來指定多點傳送主機群組，其範圍自224.0.0.0 至239.255.255.255。永久和暫時群組位址都支援。永久位址由「Internet 編號管理局(IANA)」來指定，其中包含224.0.0.1（亦即用來賦予區域網路上所有多點傳送主機位址的全體主機群組）和224.0.0.2（賦予區域網路上所有路由器位址）。224.0.0.0和224.0.0.255之間的位址範圍保留供路由及其它低等級網路傳輸協定使用。其它位址和範圍已經保留供應用程式使用，例如224.0.13.000 至224.0.13.255 保留給Net News（若需詳細資訊，請參閱RFC 1700 的「指定編號」，其位址是：<ftp://ftp.internic.net/rfc/rfc1700.txt>）。

IP 多點傳送的傳輸通訊協定是RTP（即時傳輸通訊協定），可以產生提供時間戳記、順序編號和有效負載格式資訊的標準多媒體標題。IP 多點傳送的應用包含視訊和語音會議、遠端交換、資料庫和Web站台複製、遠距教學、股

票報價分發和協同計算。目前最大的IP 多點傳送功能示範是Internet MBONE (多點傳送骨幹網)。

MBONE 是在實際Internet 上分層的實驗性全域多點傳送網路。它已經有大約五年歷史，目前負責IETF 會議、NASA 太空梭發射、音樂、音樂會和許多其它實況會議和演出。

IP 多點傳送會議TSP 主要負責使用儲存在ILS「動態目錄會議伺服器」上的「工作階段描述通訊協定(SDP)」會議描述字元來解析電腦名稱的IP 多點傳送位址。它由「會合」會議控制來補足，說明如下。IP 多點傳送會議MSP 負責為IP 多點傳送連接(包含RTP、RTP 有效負載處理器、編解碼器、接收器和篩選器) 建立適當的DirectShow 篩選器圖表。

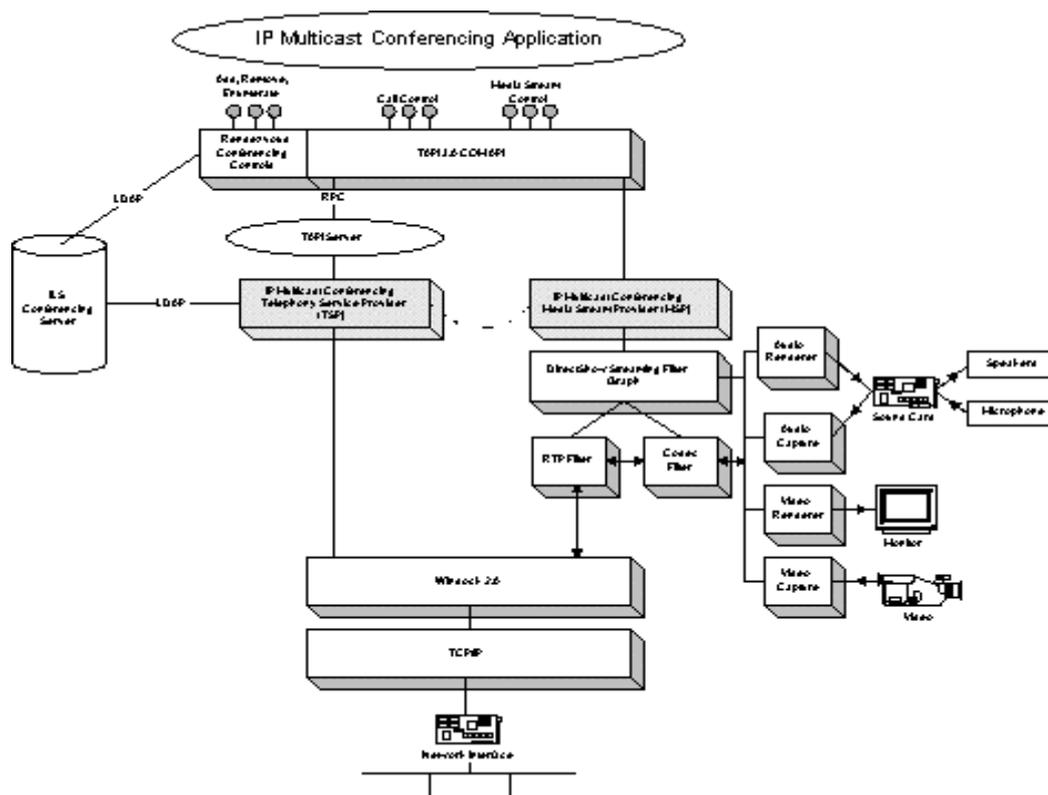


圖 2.1.4-4 IP 多點傳送的會議架構

2.1.5 H. 323

H. 323 是一種複雜的國際電信聯盟(ITU) 標準，適用於服務品質無法保證的無連接網路的多媒體傳輸(聲音、視訊和資料)，例如 IP 網路

和Internet 等。它提供點對點和多點會議的呼叫控制、多媒體管理和頻寬管理。H. 323 指令支援標準語音和視訊碼，支援透過T.120 的資料共享。而且，H. 323 標準獨立於網路、作業平台和應用程式，使任何H. 323 相容的終端機與其它任何終端機進行相互操作。

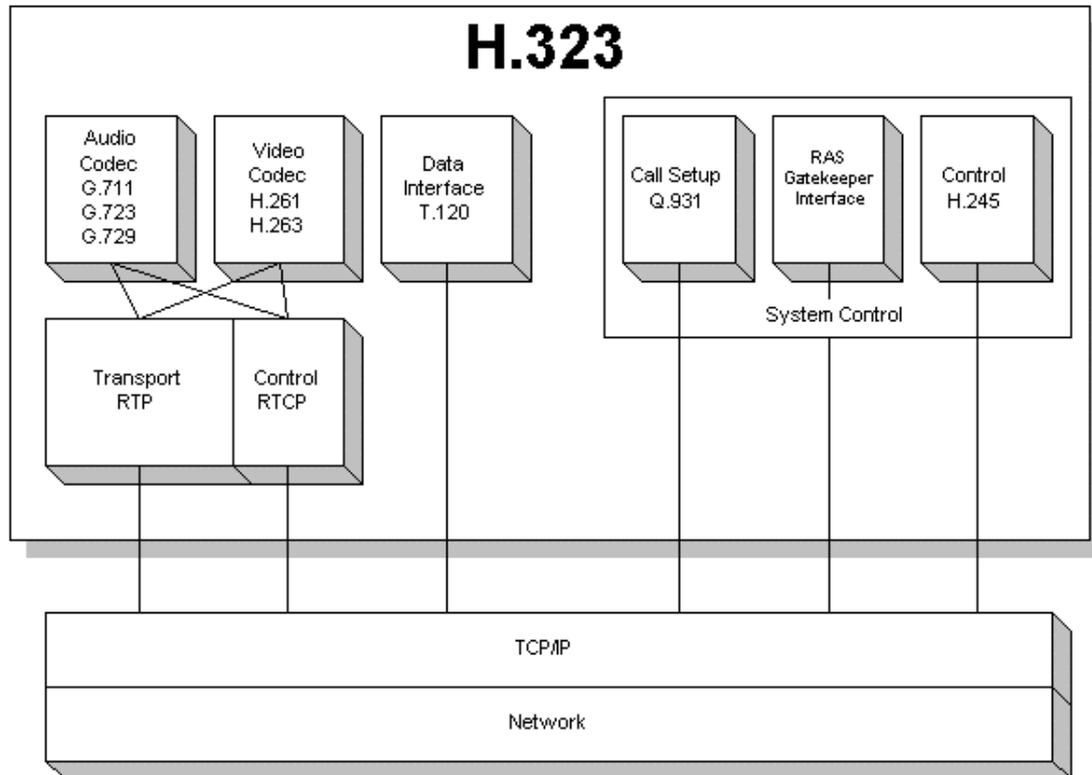


圖 2.1.5-1 H. 323 的架構

H. 323 可以在現行封包交換網路上傳送多媒體資料流。為了減小區域網路反應時間的影響，H. 323 可當作「即時傳輸通訊協定(RTP)」使用，RTP 是設計用來處理Internet 上即時語音和視訊資料流需求的IETF 標準。

H. 323 標準指定三個指令和控制通訊協定：

- 1 呼叫控制的H. 245
- 1 呼叫訊號傳輸的Q. 931
- 1 RAS (登錄、許可權和狀態) 訊號傳輸功能

H. 245 控制通道負責控制H. 323 終端機操作的控制資訊，包含功能交換、指令和指導。Q. 931 可用來在兩個終端機之間建立連接，RAS 控制登錄、許可權及端點與閘道守衛(如果沒有閘道守衛，不能使用RAS) 之間的頻寬功能。若需有關閘道守衛的詳細資訊，請參閱下節。

H. 323 定義四個主要元件，這些元件都以H. 323 傳輸系統為基礎。

1 終端機

1 閘道

1 閘道守衛

1 多點控制單元(MCU)

終端機是指網路用戶端的端點。所有終端機都必須支援聲音傳輸；視訊和資料支援則是選擇性的。

閘道是H. 323 會議的選擇性元件。閘道將H. 323 會議橋接到其它網路、傳輸通訊協定和多媒體格式。如果不需要結合其它網路或非H. 323 相容終端機，就不需要閘道。

閘道守衛執行協助維持網路強健性的兩項重要功能：位址轉換和頻寬管理。閘道守衛將區域網路別名映射到IP 位址，然後在需要時提供位址搜索。閘道守衛還執行呼叫控制功能，以限制H. 323 區域內的H. 323 連接的數量和這些連接使用的總頻寬。H. 323 系統不需要閘道守衛；但如果有閘道守衛，終端機就必須使用此服務。

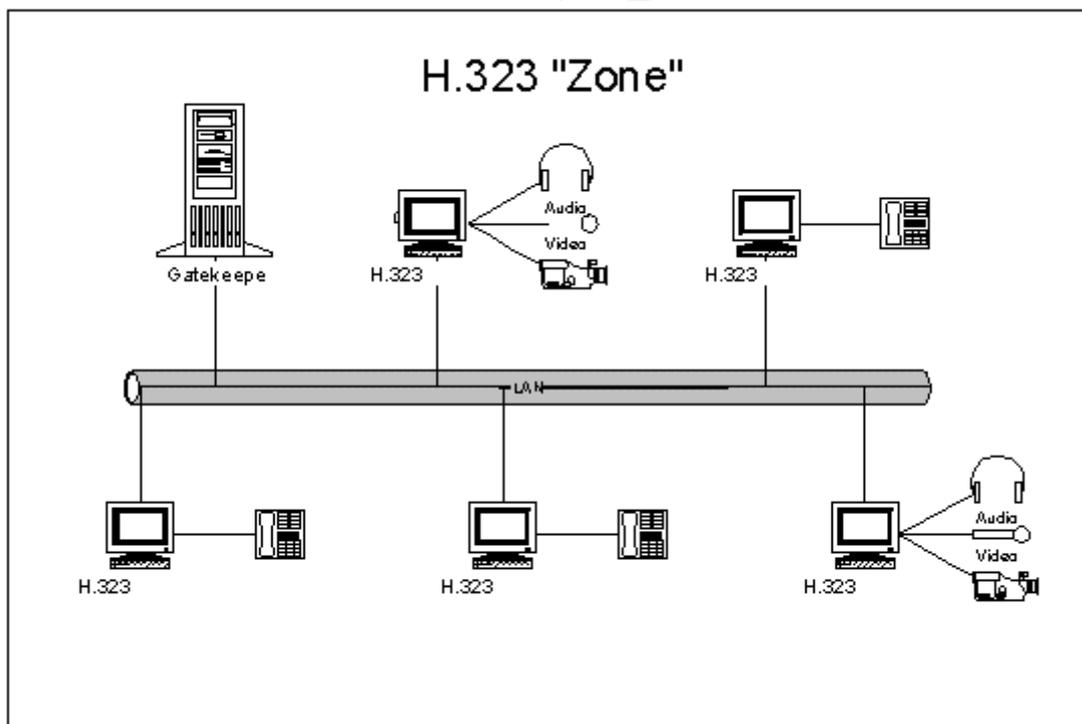


圖 2.1.5-2 H. 323 的元件

多點控制單元(MCU) 支援三個或更多端點之間的會議。MCU 由必要的多點控制器(MC) 和0 個或多個多點處理器(MP) 組成。MC 執行所有終端機之間的H. 245 協商，以決定一般語音和視訊處理功能，而多點處理器(MP) 在終端機的端點間傳輸語音、視訊和資料流。保證任何H. 323 用戶端都支援下列標準：H. 261 和G. 711。H. 261

是ITU 標準的視訊編解碼器，用來傳輸速度為64 Kbps、解析度為176x44 圖素(QCIF) 的壓縮視訊。G.711 是ITU 的標準語音編解碼器，設計用來傳輸速度為48、56 和64 Kbps 的A-law 和μ-law PCM 的語音。

H.323 用戶端可以選擇性地支援其它編解碼器：H.263 和G.723。H.263 是以H.261 為基礎並與其相容的ITU 標準視訊編解碼器。它提供優於H.261 的壓縮功能，並以176 x 44 圖素(QCIF) 解析度傳輸視訊。G.723 是設計用於低位元速率操作的ITU 標準語音編解碼器。

H.323 「電話服務提供程式」(及其相關的「媒體資料流提供程式」) 可讓啟用TAPI 的應用程式使用區域網路上任何H.323 相容的終端機來參與多媒體工作階段。

特別的是H.323 「電話服務提供程式(TSP)」建置了H.323 訊號傳輸堆疊TSP 接受許多不同的位址格式，包含姓名、電腦名稱和電子郵件位址。

H.323 MSP 負責建立H.323 連接(包含RTP、RTP 有效負載處理器、編解碼器、接收器和篩選器)的DirectShow 篩選器圖表。

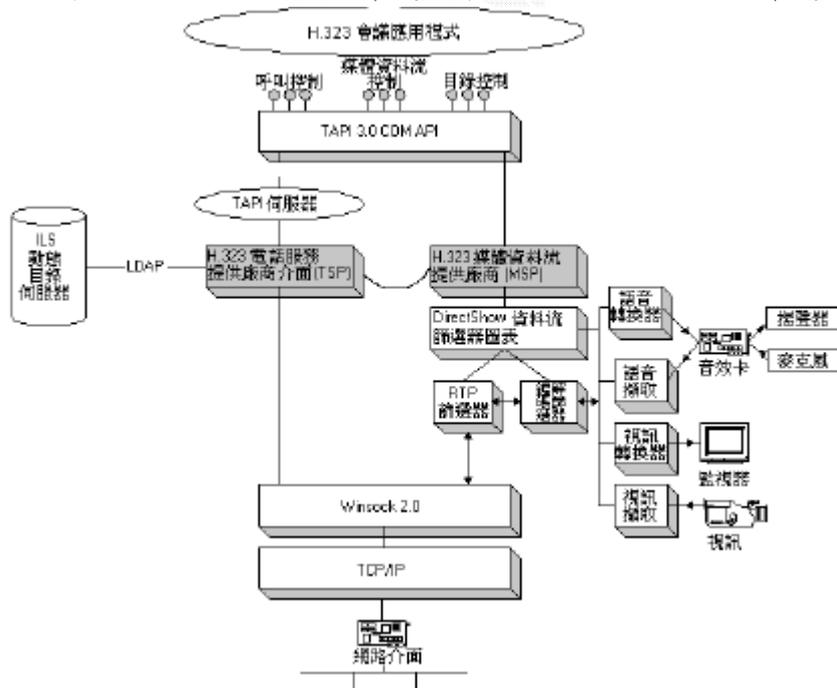


圖 2.1.5-3 H.323 的 TSP 架構

與 WIN200 的整合

由於使用者的網路位址(在這種情況下是使用者的IP 位址) 非常不穩定，在H. 323 工作階段期間不能保證維持不變，所以H. 323 電話服務是非常複雜。TAPI H. 323 TSP 使用Windows 2000 ActiveDirectory 服務執行使用者至IP 的位址解析。特別的是使用「Internet 定位器服務(ILS) 動態目錄」(Active Directory 的即時伺服器元件)，來儲存和不斷重新整理使用者至IP 的映射資訊。下列使用者分析藍本說明H. 323 TSP 中的IP 位址解析：

1. John 希望啟動與區域網路上另一位使用者Alice 的H. 323 會議。當Alice 的視訊會議應用程式建立位址物件並使它成為接聽模式之後，Alice 的IP 位址就由H. 323 TSP 加到Windows 2000 Active Directory 中。這個資訊的生存時間(TTL) 有限，會定期透過「輕型目錄存取通訊協定(LDAP)」來重新整理。

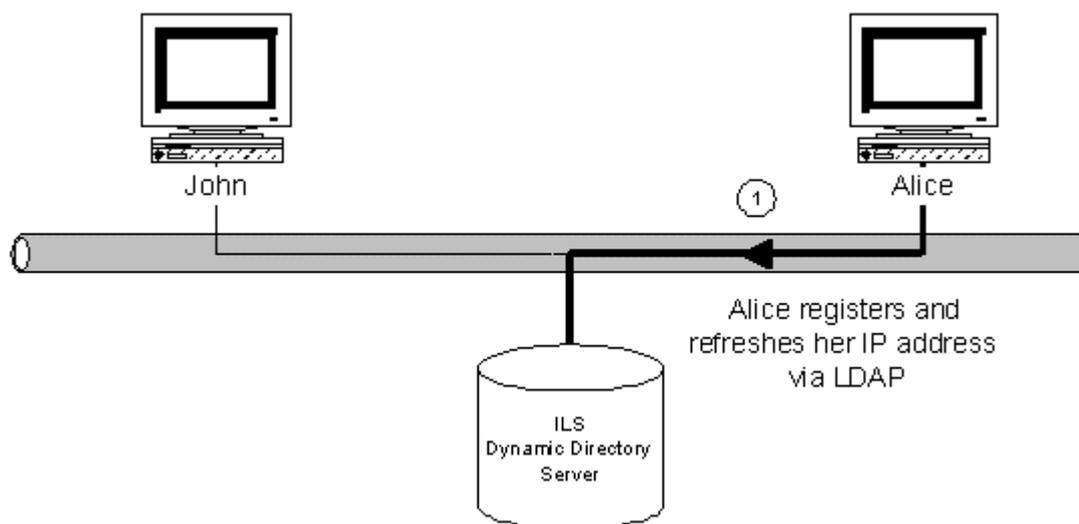


圖 2.1.5-4 Alice註冊並更新她的IP位址

2. John 的H. 323 TSP 隨後查詢ILS 動態目錄伺服器中Alice 的IP 位址。特別的是John 在Alice 的相關目錄中查詢任何和所有 RTPerson 物件。

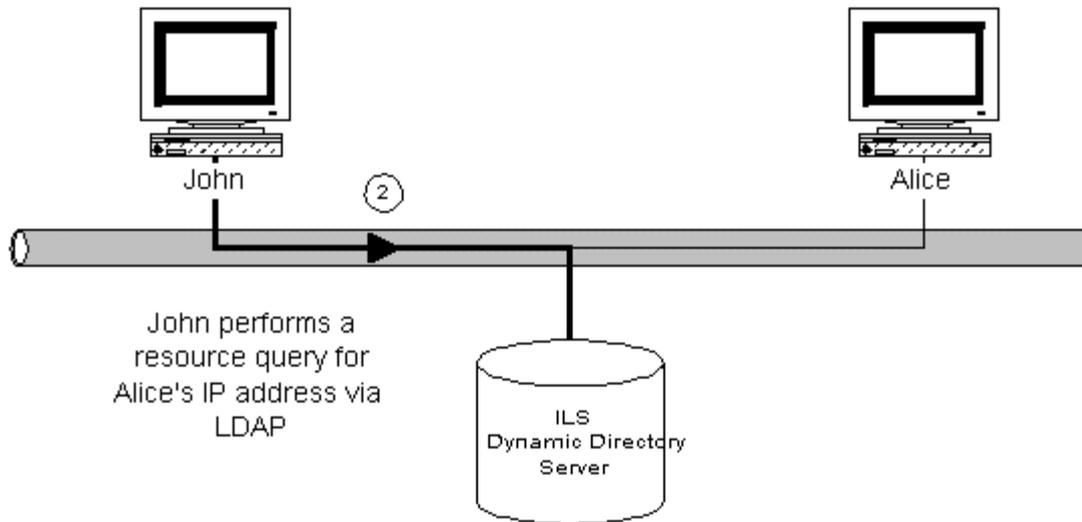


圖2.1.5-5 John查詢Alice的位址

3. 有了Alice 的最新IP 位址，John 就開始呼叫Alice 電腦的H. 323，然後兩部電腦上的同等TSP 之間就會出現H. 323 標準的協商和媒體選擇。一旦完成功能協商，兩個H. 323「媒體資料流提供程式(MSP)」都會建立適當的DirectShow 篩選器圖表，所有媒體資料流都送到DirectShow 處理。然後會議就開始了。

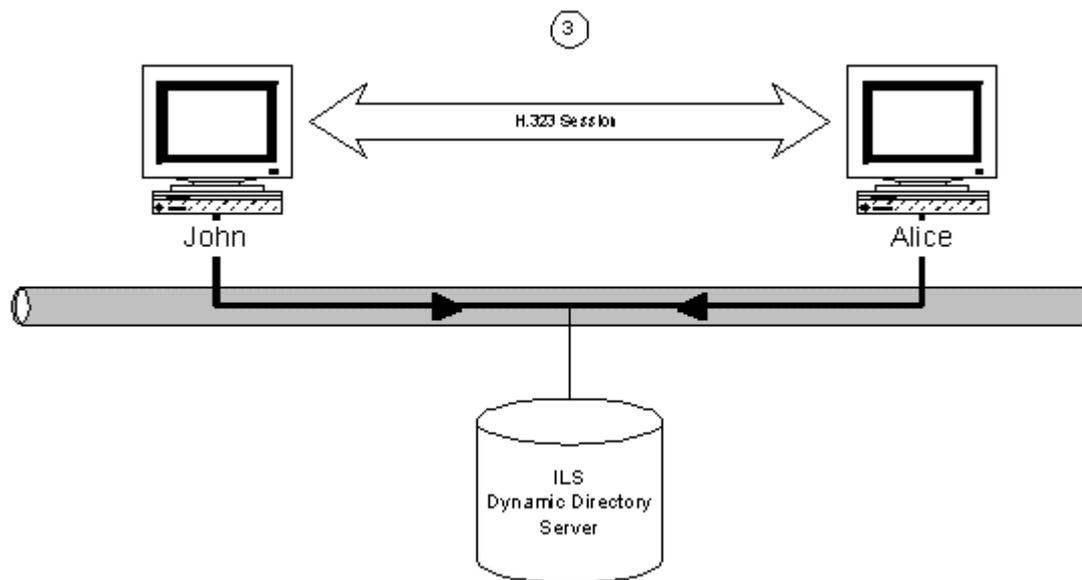


圖 2.1.5-6 協商完成會議開始

2.2 COM 的介紹

2.2.1 COM 的基本認知

讓我們來研究有關其他兩種用於建立元件的技術，來逐漸了解 COM。請試想 C++類別的繼承與封裝的使用，我們只需注意用於 windows 程式設計的框架，如 MFC，就可以看到類別是如何有效地為我們軟體提供「黑盒子」－隱藏了更深的複雜性頂層包裝程式。

然而，C++類別是特別針對 C++程式的，而且有自己的內部架構。那即是說，因為 Smalltalk 跟 C 物件的內部架構不同，所以兩者不會有太多相似之處，而且也沒有可讓它們相互操作的標準方式。事實上許多 C++的物件甚至常常無法與由不同編譯器所產生的其他 C++物件共同使用，因為 C++語言規格的實作從屬服務仍然從缺。沒有適用於已編譯的 C++物件的二元標準。

另一個方法就是 DLL。DLL 允許執行時間連接，並且較不具語言獨立性，然而這也是問題的。因為 DLL 僅已單一常式的基礎來匯出，而不是匯出全部的物件，而且只限於 Windows 平台。

而 DLL 是 Windows 基礎的一部份，為什麼我們要用 COM 呢?原因有下：

- DLL 名稱與位置是傾向強制編碼的，應用程式會按照名稱來載入 DLL，如果名稱有所變更，DLL 就會變成隱形的。DLL 亦必須常駐於應用程式目錄中、Windows 系統目錄中或是路徑上，以便尋找。
- 在 Win32 底下，DLL 是很難處理之間共享的，因為 DLL 會被載入至想要使用它們的各處理的位址空間中，而此舉既浪費時間又浪費空間。不適合應用程式軟體。
- 多重服務 DLL 需要協調序數或函數名稱，DLL 有固定的經常性耗用時間，因此結合功能與單一 DLL 常常是很有用的，但是如果來自不同服務的常式

之前已配置相同的序數，此舉就可能產生問題。

- DLL 並不支援版本的問題，既然 DLL 只是按照名稱來加以附註，因此不一定需要分辨你系統上的 Fred.dll 是代表 1.0、1.1 或是 2.0 版。一部機器上很有可能具有不同版本的 DLL，而且這種情況有時可能會引起很難解決的問題。覆寫較舊的 DLL 版本就是有關的問題之一。

2.2.2 COM 的運作方式

COM 為我們提供了一種像在 C++ 中般使用物件的方法，即使這些物件並不位於相同的程式中(或是甚至在不同的機器上)，也不是以相同的語言來編寫的。執行此工作的方法就是指定軟體物件以二元格式來配置的方式(換言之，就是應如何在記憶體裡配置)，以及它們如何能夠依靠特定的機器或語言，就可彼此聯繫，使之成為「結構中立」(architecture-neutral)

COM 物件是由「介面」(interface)的集合所構成的，介面是物件將其功能顯在出來的方法。在 COM 中，介面就是一張包含從物件所實行的指標到函式的表，該表代表介面，而他所指向的函式就是該介面的方式。一個物件可以依其選擇顯示許多的介面。

所定義的 COM 物件的二元配置會管理其可公開使用的函式表的配置方式，意即格式與用於編寫物件及執行機器的語言是無關的。這種表與 C++ 的「虛擬函式表」(virtual function tables)類似，稱為 vtables。在 Visual C++ 中，COM 的 vtables 與 C++ 的虛擬函式表有著相同結構。對其他編譯器來說，這不一定是正確的，也就是說他們在代表 C++ 中 COM 物件時，必須要執行較多的工作。

使通訊標準化意指一個 COM 物件可以在執行時間尋找另一個物件，並且知道要如何交談，既然已經知道了介面的配置，所以任何 COM 物件應該都有可能可以使用另一個物件所顯示的函式。

有些 COM 的文件中，會到許多次的「伺服器」(serves)「用戶端」(clienes)。伺服器是指提供他人可能想要使用的介面的 COM，而用戶端則是藉由呼叫其中一項或多項函式的介面，來使用介面的 COM 物件。

COM 物件是軟體中已經理想化的部分。在實際的情況中，必須要這些物件包裝起來，然後找個位置安置。在 Windows 環境中，有三種主要的方式可以包裝物件：

- 在 DLL 中
- 在 EXE 中
- 從其他地方來提供，作為遠端服務

從未自行執行，而且只有被其他應用程式來當作元件可以包裝在 DLL 中，這種情形的典型範例之一就是 ActiveX 控制，其中一項或多項都是包裝到 DLL 中，且是在有需要時由用戶端載入的。DLL 中的物件會在用端的位址空間執行，而常常成為「連線處理」(in-process 或 in-proc)

包裝在 EXE 檔案中的物件會在他們自己位址空間中執行，而有時稱為「離線處理」(out-of-process)。的例子是在提供 COM 物件作為較大應用程式的一部份時，例如由 WORD 與 EXCEL 等應用程式所顯示出來的「自動化」物件。

如果我們正在使用 LAN、WAN 等網際網路，那麼我們可能會使用已包裝好的 COM 物件來作為一項服務，由來自他處的另一部機器與我們聯繫。我們不知道在另一端服務我們的呼叫的是什麼，不過有木正在網路上與我們交談，也就是物件透明化的觀念，簡單的說，用戶端並不在乎正在使用的物件位於何處，碩要如何存取。只要是與用戶端有關，在 DLL 中使用物件與經由網路在另一部機器上或是在相同但為離線處理的機器上存取物件是完全相同的。

2.2.3 COM 介面

無疑的，介面對於 COM 是非常重要的，這是 COM 物件能夠聯繫的唯一方式。介面是一份函數指標表，他會依次指向在某方面有關聯（例如處理檔案）的函

式，或是執行特定的計算。下圖顯示此一概念：

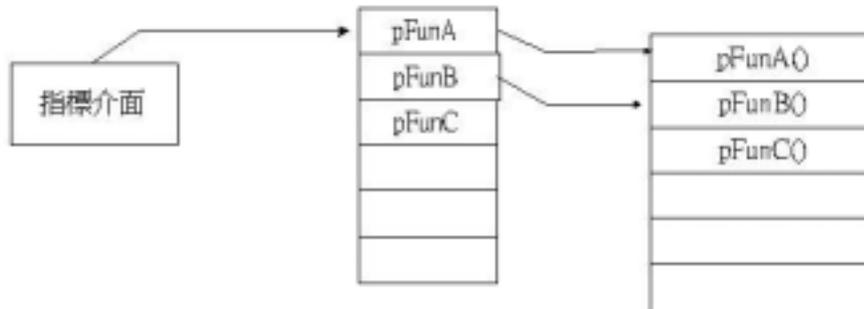


圖 2.2.3-1 COM 介面和 vtable 的關係

介面是由許多指向實際函式本身的函式指標（pFunA, pFunB 等等）所構成的，如果我們取得了介面的指標，那麼我們就可以呼叫函式。

雖然將任何一組的函式一同歸類到一個介面是有可能的，不過像 C++ 類別，你就會期望在成員之間有一種基本的凝聚力。按照習慣，見面的名稱是以 I 開始的，例如 Iunknown 與 Idispatch。

下圖顯示一般取得 COM 介面的方式，其方塊代表物件，而圈圈代表物件所顯示的介面

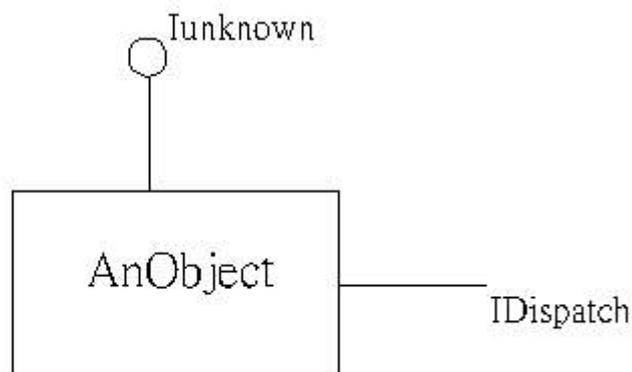


圖 2.2.3-2 COM 介面的取得方式

COM 介面可說是以一種物件導向的方式來表顯的，至少只要是與它罷部分的著名功能有關－資料分離、封裝與同名異式。

在建立 COM 元件時，我們可以使用介面使發生版本問題的可能性減到最小。方法很簡單，一旦已經定義並出版一種介面（換言之，就是一但用戶端可以使用），就已成定局了，沒有東西能改變使用該介面的方式。

介面就像是物件與用戶端的協議，如果你需要改變介面，該怎麼辦呢？在 COM 中，你會建立一個新的介面，然後從現有的介面來合併你所需要的元素，重點是在於物件仍然會支持舊的介面，因此那些有需要的較舊的用戶就可以找的到它，下圖顯示其運作方式。

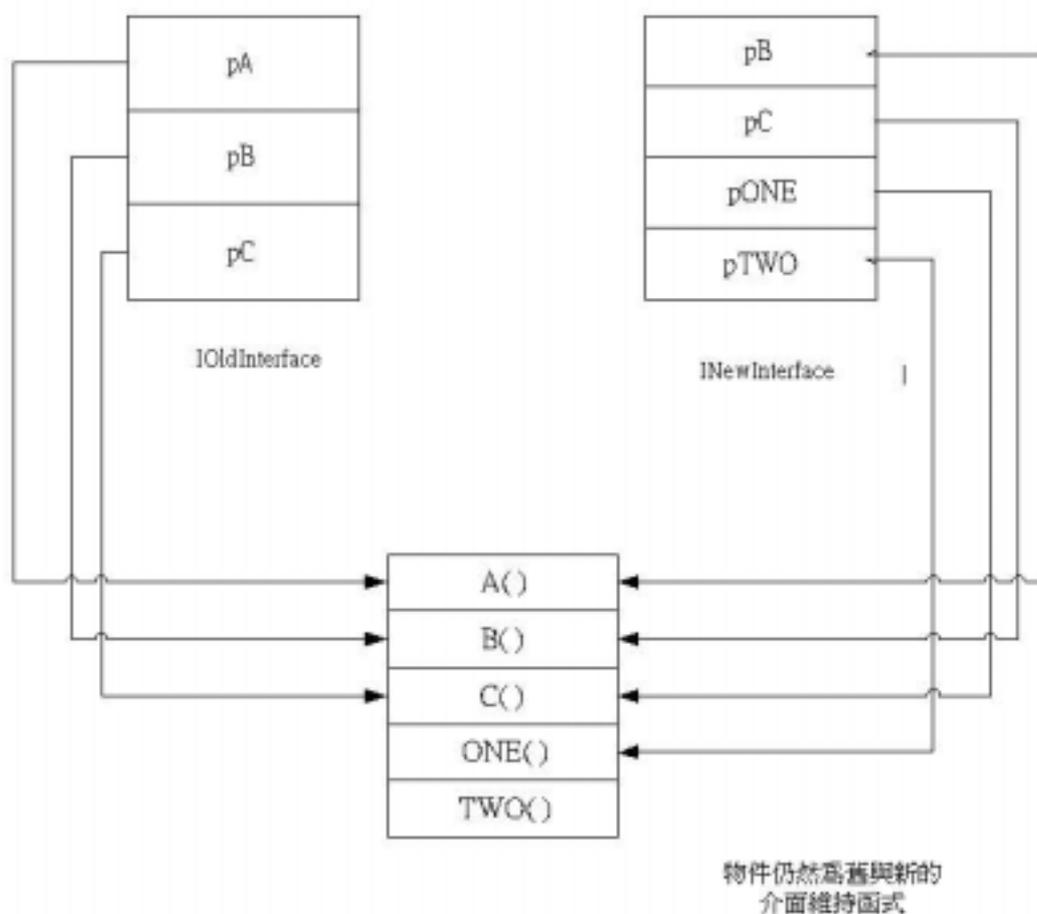


圖 2.2.3-3 新舊 COM 介面版本的處理方式

在上圖中，COM 物件有個支援三項函式的簡單原始介面，這三項函式稱為 A () B () 與 C ()，然後有了新的需求，突然新增函式 ONE() TWO()，並將參考資料移致 A ()。因此產生了 INewInterFace，它合併了 IoldInterFace 的其中兩種函式，然後在新增自己的。結果皆大歡喜，較舊的應用程式仍然可以找到 IoldInterFace，而較新的應用程式則會得到 IoldInterFace。

2.2.4 COM 基本介面介紹：

Iunknown：

當談論到 COM 介面時，我們必須從最基本的 Iunknown 開始。它的用途並不多，不過它的獨特性是來自

- 所有 COM 物件都必須要實行 Iunknown 介面
- 所有 COM 物件都是從這三項 Iunknown 函式開始的，意即每個介面多少是 Iunknown 繼承而來的
- Iunknown 為 COM 物件提供控制其醫生的方法，並未使用者在介面之間導覽的能力

QueryInterface：

所有的 COM 物件與介面都有不同且專用的識別碼(ID)，因此我們可以呼叫 QueryInterface () 並傳送我們的有興趣介面的 ID，來詢問物件是否支援特定的介面。當我們將一個介面傳給 IDQueryInterface 時，如果可以它就會傳回一個指標給介面，不行的話，就會傳回 NULL。

2.3 與 TAPI 相容的語音 MODEM

什麼是語音 MODEM

大多數的 MODEM 都可被歸類為下面這幾類

- 資料
- 資料/傳真
- 資料/傳真/語音/對講電話

而語音 MODEM 就是一個 MODEM 在一個電話路上撥放及紀錄聲音。當大部分的 MODEM 都包含有資料/傳真功能的時候，越來越多的 MODEM 也包含語音的功能。

什麼是 TAPI 相容的 MODEM

TAPI(Telephony Application Programming Interface)是微軟所提供的一種程式寫作標準。它允許軟體開發者在發展電信方面的軟體，可以適用於每一種跟 TAPI 相容的 MODEM。

為了讓 MODEM 可以跟 TAPI 相處的愉快，這 MODEM 必須要有語音和聲音的驅動程式。WIN9X/ME 藉著 Unimodem/V 而 WIN2000/XP 藉著 Unimodem 5 來語音驅動程式做溝通。聲音的驅動程式通常語音 modem 的製造商都會提供

如何確認 MODEM 的聲音驅動程式是否安裝

假使你的 MODEM 支援語音，那麼 MODEM 的聲音驅動程式必定會列在 WINDOWS 的系統裝置管理員上

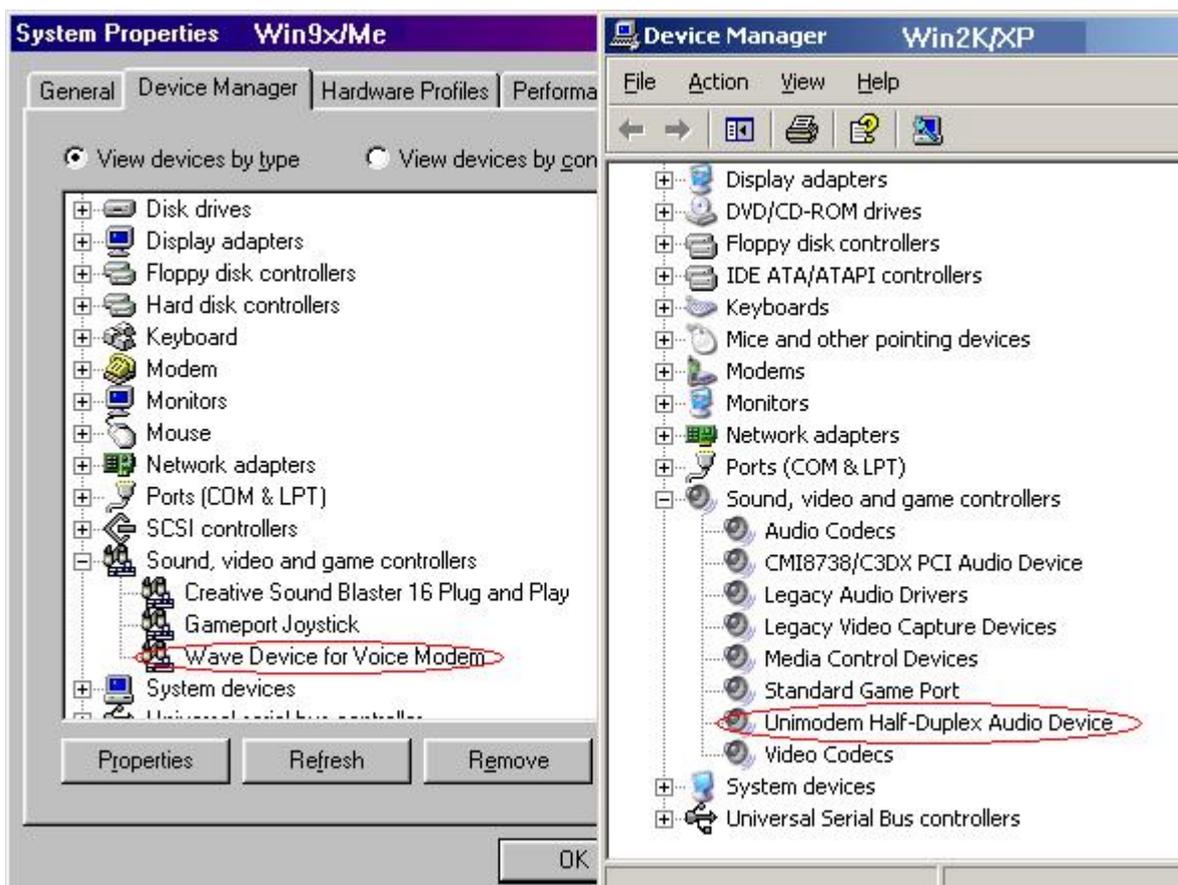


圖 2.3-1 Win9X 及 2K/XP 下的語音裝置位址

我的 MODEM 支援語音但我不能找到語音裝置

Windows 2000/XP 使用一個"generic"的 MODEM 驅動程式來包含所有多人使用的晶片組，在粉多的情況下，增加、改變或移除 MODEM 是粉簡單的。隨插即用，但它不總是完全適用的。

在許多情況下，Windows 2000/XP 會認為你的資料/傳真/語音是只有資料 MODEM 功能，並且安裝 Windows 內建的"generic"MODEM，所以你不能看到 MODEM 的語音裝置列在裝置管理員上。這樣你只能撥號到網際網路上而不能用你的 MODEM 紀錄電話呼叫。

假設你確定你的 MODEM 支援語音，請更新或者重新安裝正確的 MODEM 驅動程式

如何更新你的語音 MODEM 驅動程式

1. 找到正確的 MODEM 驅動程式：MODEM 的附的 COROM 或者廠商所提供的下載位址。
2. 開啟裝置管理員，可參考下面的圖。
3. 開啟數據機的前面的+號來展開，再來雙擊你的 MODEM 裝置選擇驅動程式那一頁，就能看到更新驅動程式按鍵
4. 現在你就能更新你的驅動程式

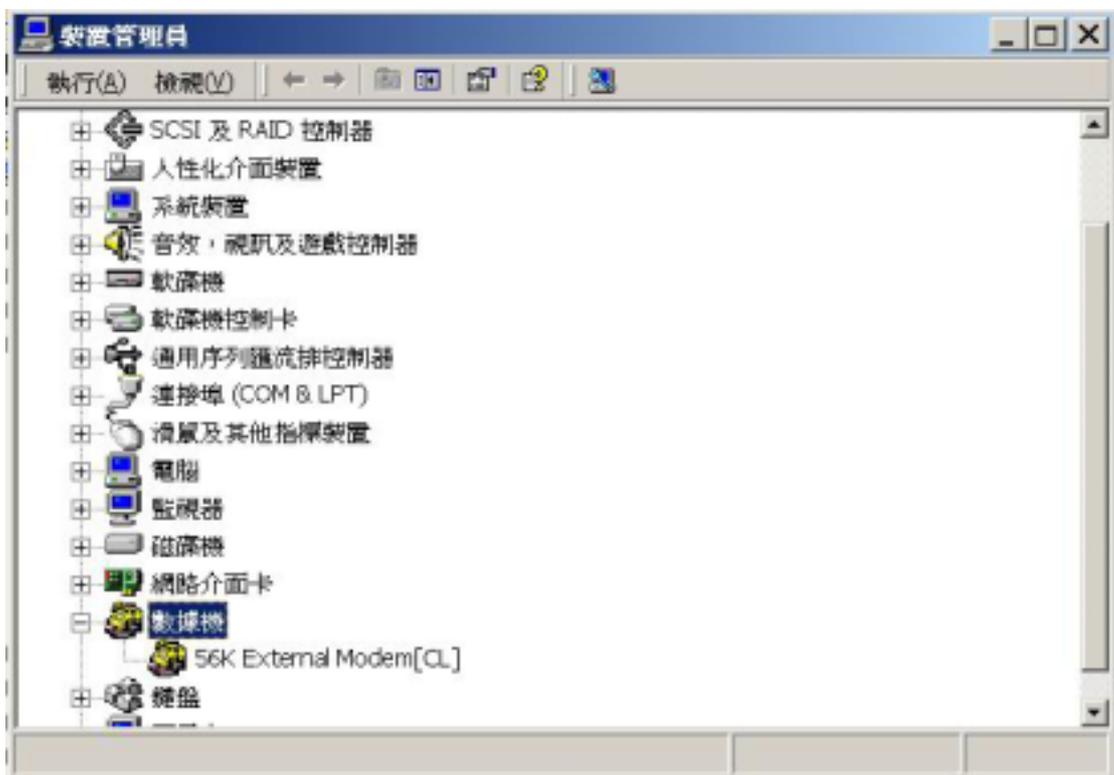


圖 2.3-2 如何更新你數據機的驅動程式 Part I



圖 2.3-3 如何更新你數據機的驅動程式 PartII

如何重新安裝驅動程式

1. 找到正確的 MODEM 驅動程式：MODEM 的附的 COROM 或者廠商所提供的下載位址。
2. 開啟裝置管理員，可參考下面的圖。
3. 開啟數據機的前面的+號來展開，再來 MODEM 裝置上按右鍵，就能看到解除驅動程式的選項
4. 現在你就能解除你的驅動程式，再來使用新增/移除軟體就行了

建議的 MODEM

型號	如何買
Aopen FM56-PX (內接 PCI)	MyAopen : http://store.yahoo.com/myaopen/aopenmodem Axiontech : http://www.axiontech.com/prdt.php?item=16259 NCIX (Canada) : http://www.ncix.com/index.php?affiliateid=2632562
Aopen FM56-EXV (外接式)	MyAopen : http://store.yahoo.com/myaopen/aopenmodem NCIX (Canada) : http://www.ncix.com/index.php?affiliateid=2632562

第三章 系統架構

3.1 系統需求

3.2 系統流程圖

3.1 系統需求

Speaker(喇叭)

Voice Modem(語音數據機)

MicroPhone(麥克風)

WebCamara(網眼)

Windows 2000/XP 以上



3.2 系統流程圖

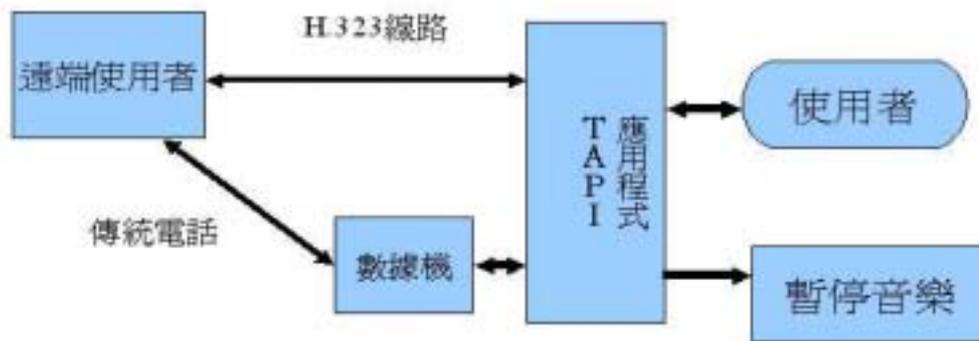


圖 3.2-1 系統應用流程圖

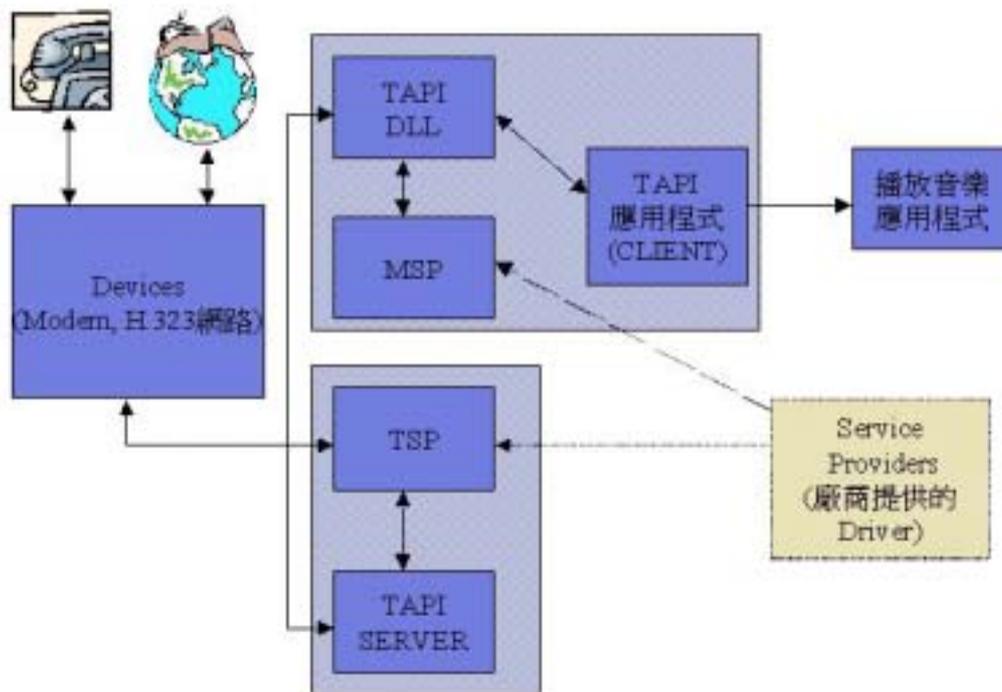


圖 3.2-2 TAPI Application 與電話架構示意圖

Initial
TAPI



圖 3.2-3 Initial TAPI Object

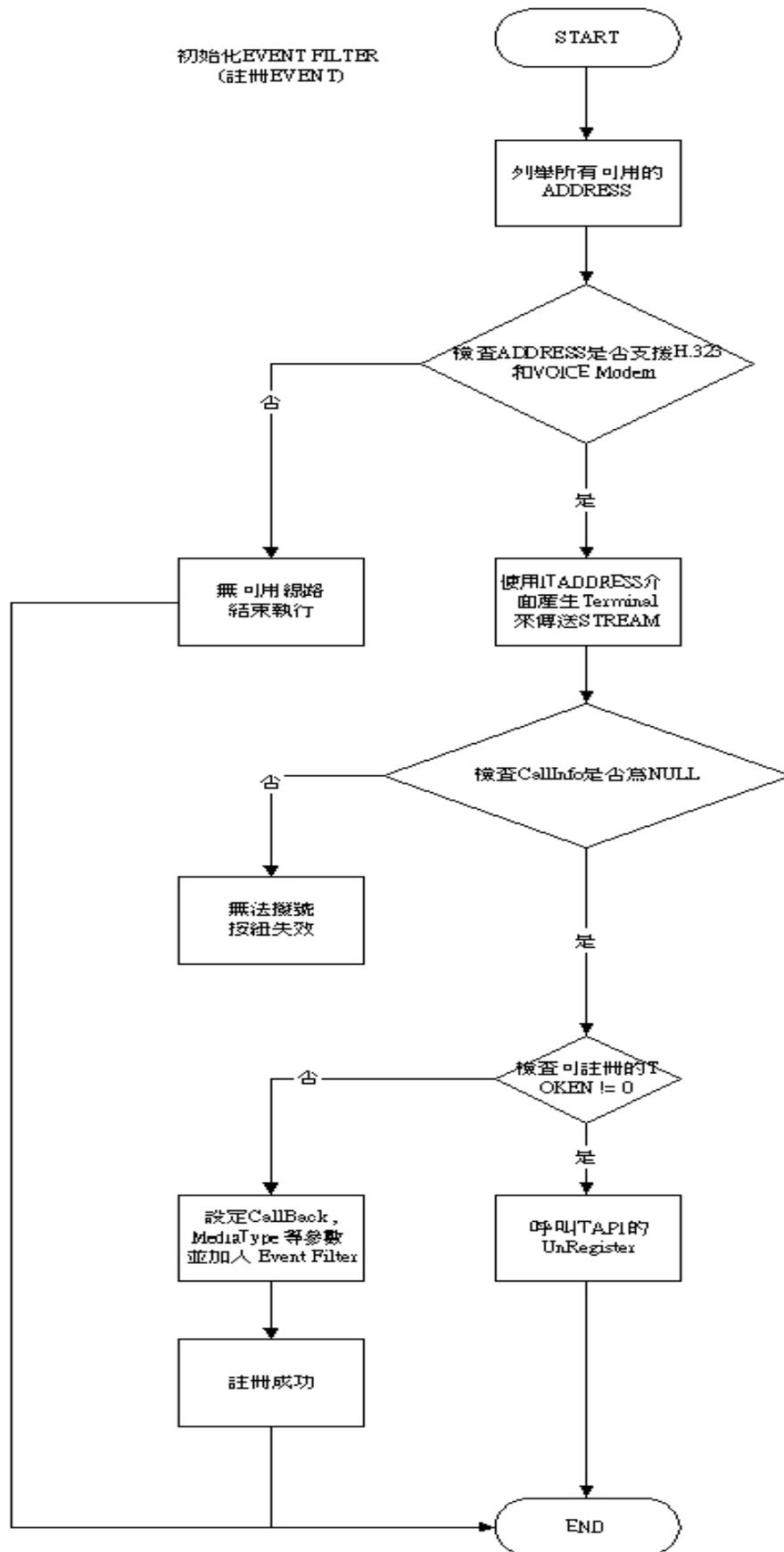


圖 3.2-4 初始化 Event 及 Register Filter

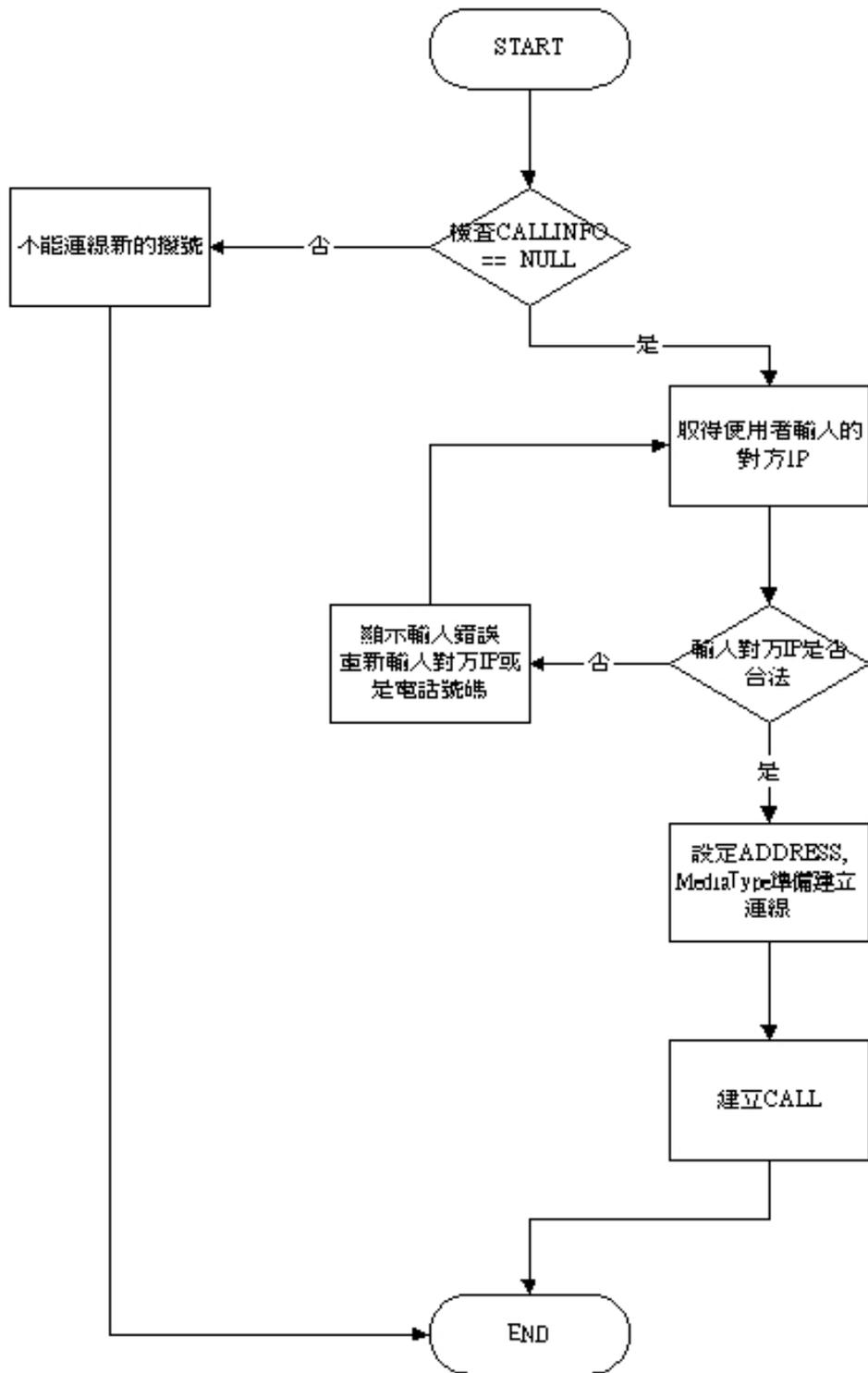


圖 3.2-5 建立連線及撥號

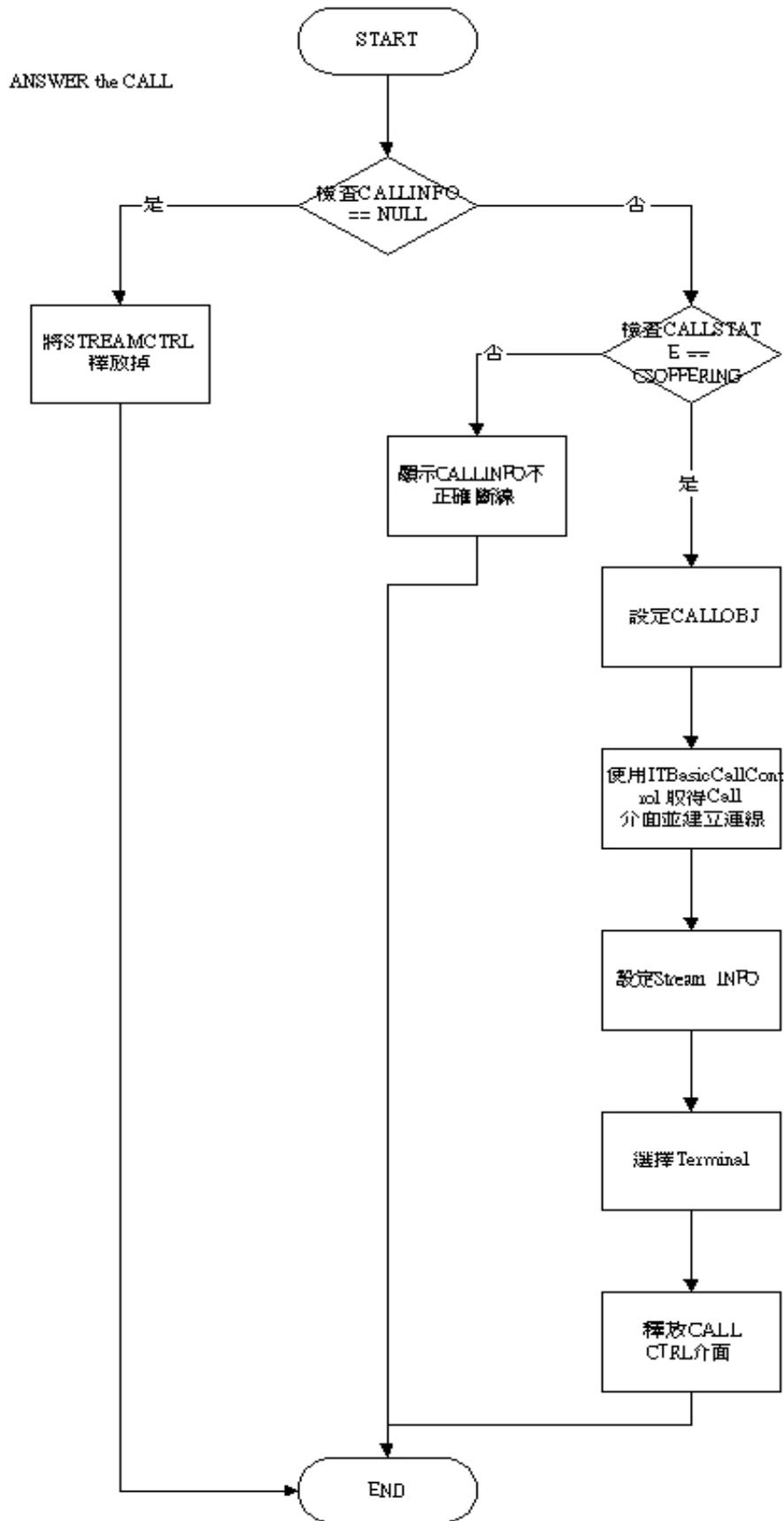


圖 3.2-6 接聽電話

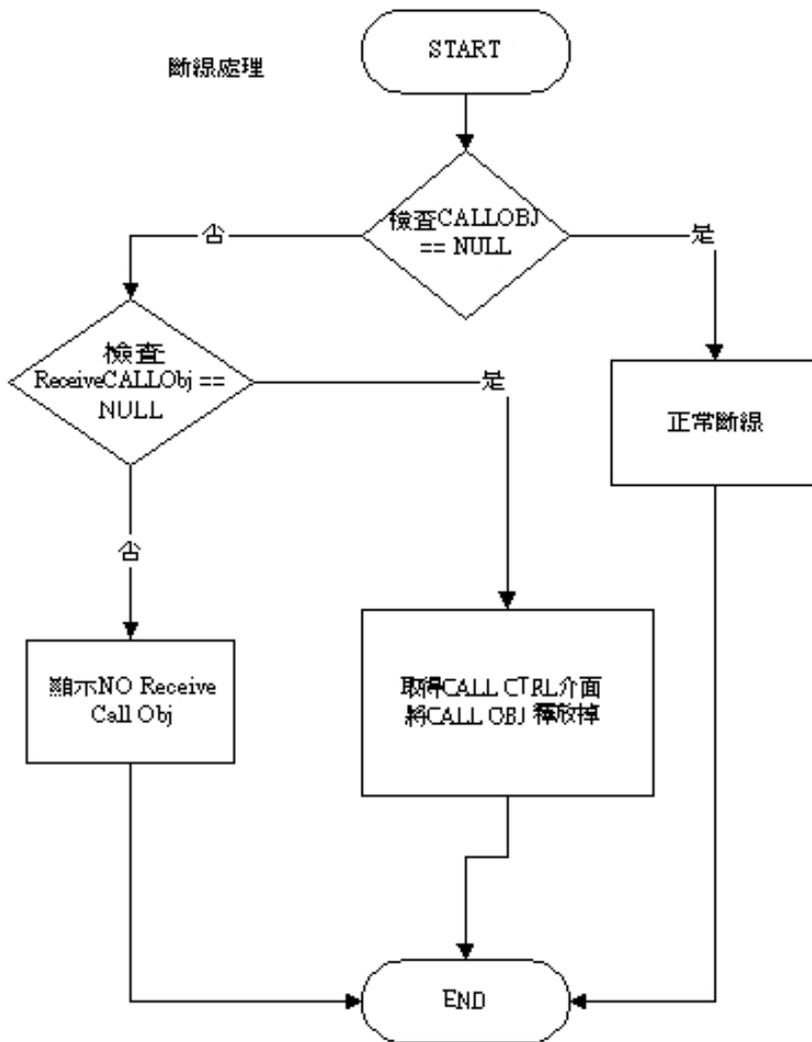


圖 3.2-7 斷線處理

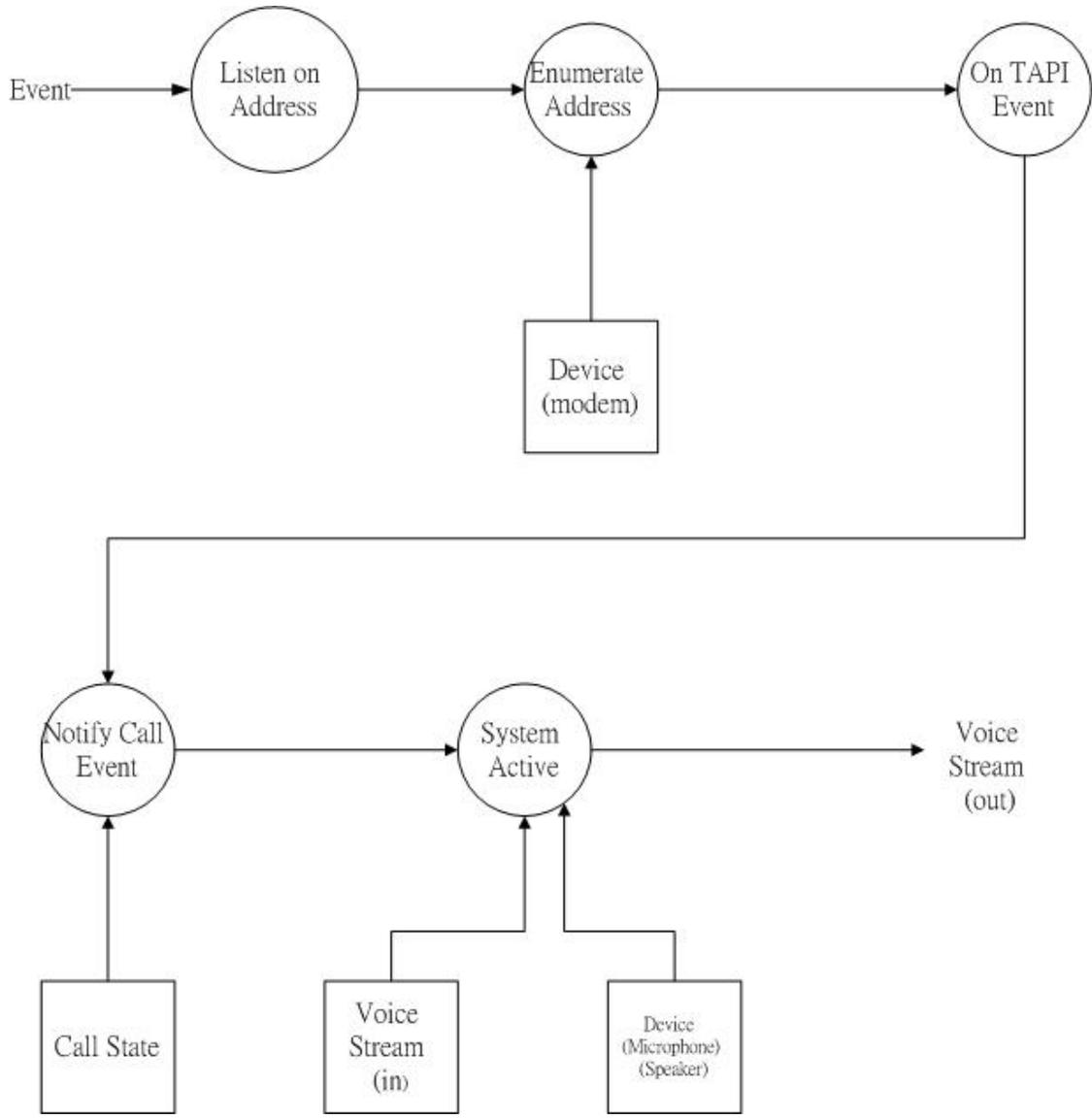


圖 3.2-8 接聽電話 event 資料流程圖

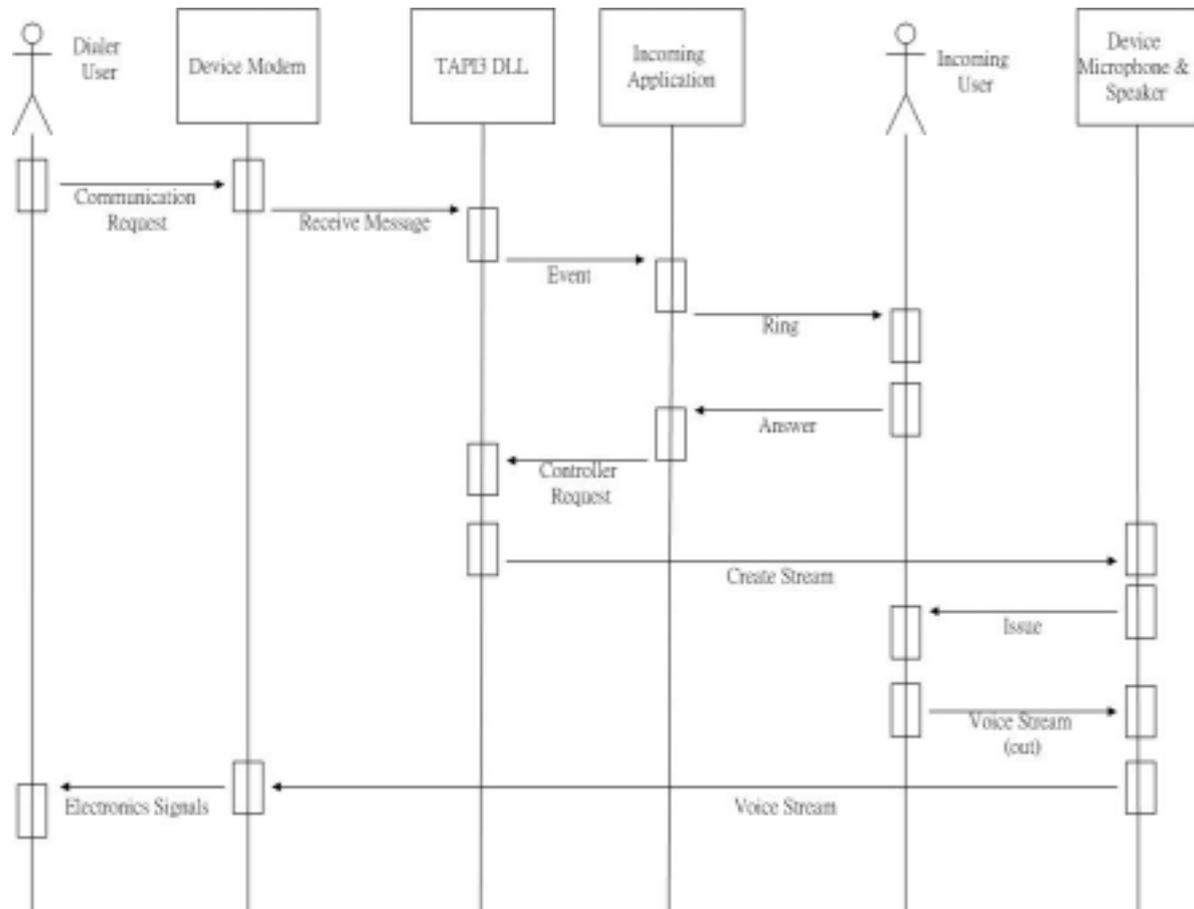


圖 3.2.9 接聽電話 USE-CASE 圖型

第四章 結論

- 4.1 軟體功能介紹
- 4.2 個人心得
- 4.3 應用與展望
- 4.4 工作分配

4.1 軟體功能介紹

原本分成 Outgoing(撥號) 和 Incoming(接聽) 兩個單獨的應用程式。後來又改成整合成一個應用程式，共用一個 Call 物件。

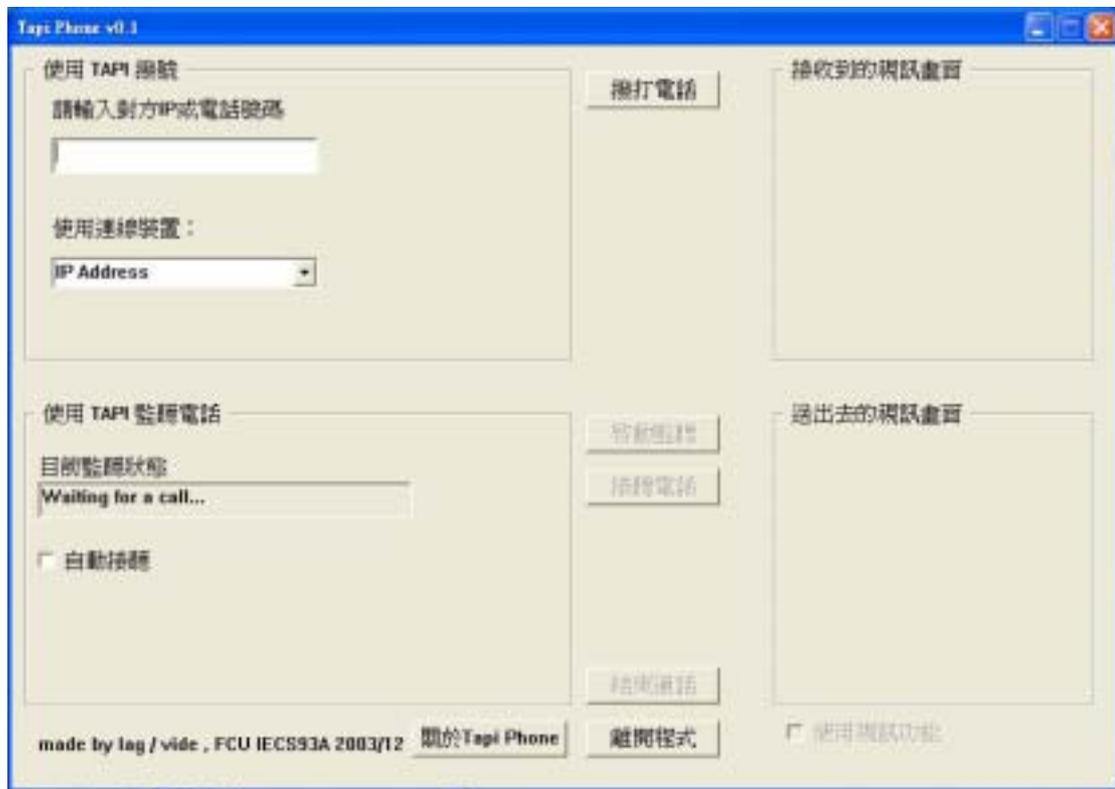


圖 4.1-1 等待接聽電話的圖片

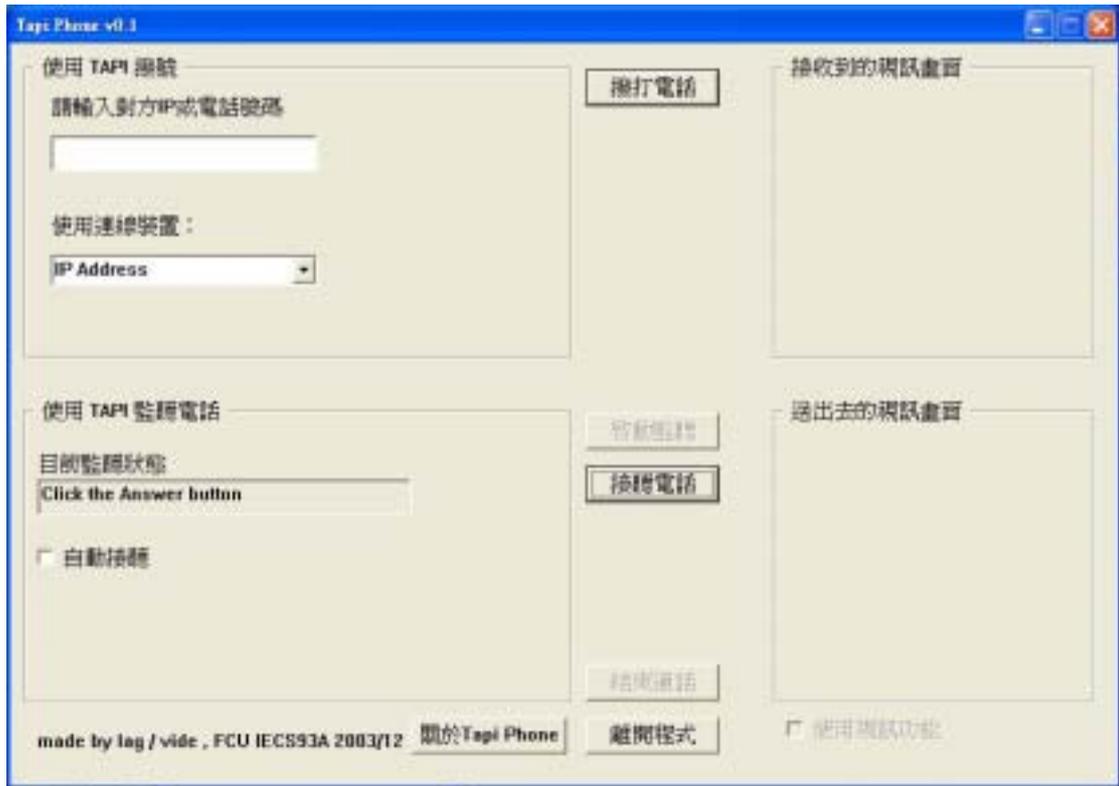


圖 4.1-2 電話訊息來電中



圖 4.1-3 通話中的圖片

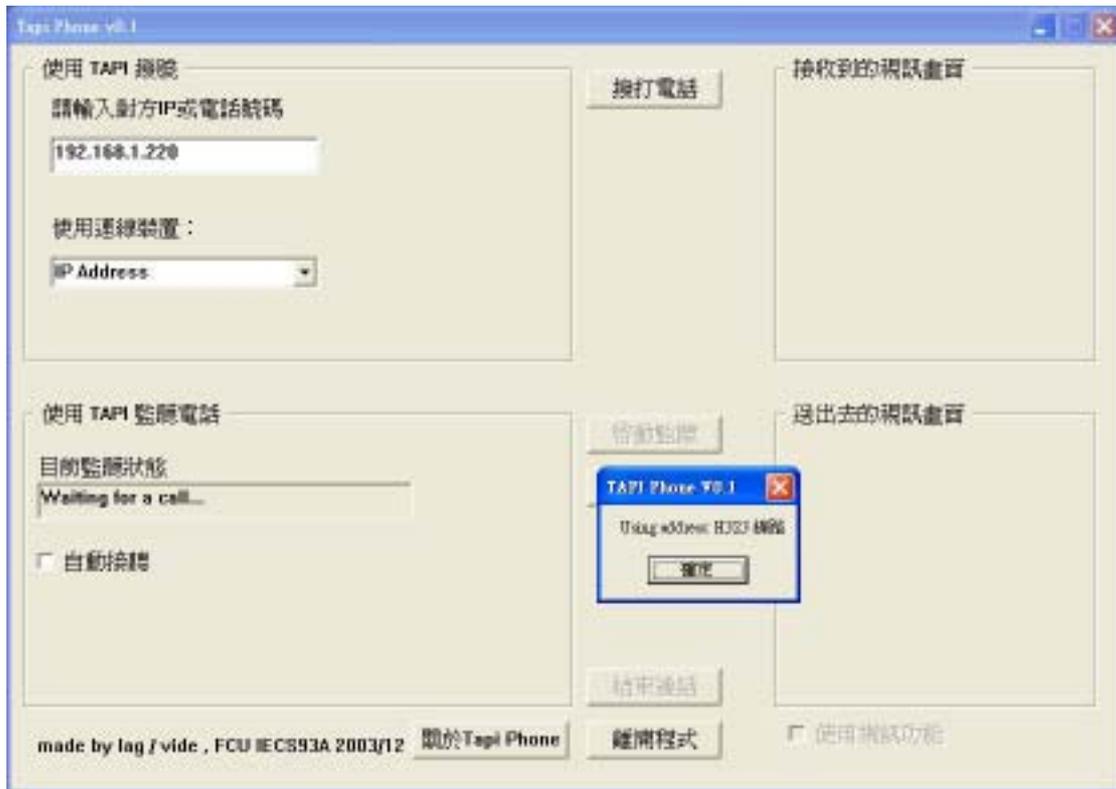


圖 4.1-4 撥號中的圖片



圖 4.1-5 撥號部份的通話中畫面

4.2 個人心得

徐麒翔：

在這個專題中，我們第一次碰到有關 TAPI 的東東，有很多地方都不知道該如何下手，四處問人無門後，兩個人只好開始從 COM 的架構開始看起，因為一開始連該如何 COM 都有問題，更遑論寫一個 TAPI3.0 的程式了。在大致上了解 COM 的架構組成後，還一度以為要自己做一個 COM 物件來使用，結果發現就像物件導向的觀念一樣，我們只要學會如何用 TAPI3.0 給我們的物件就行了。

至於語言方面，我們利用 visual C++ 6.0 來寫程式，畢竟 C++ 是兩個人都比較熟的關係，而在做這個專題時，我們學會該如何透過網路來尋找我們要的資訊，遇到問題經過討論還是不知如何下手的時候，網路就成了我的好幫手，當找到答案，那感覺好像是雨降甘霖，此外，並且體認到討論的重要，之前在作業沒有遇到這麼大的合作經驗，所以試著溝通、如何正確表達自己的意思是一大課題，也學習到如何分工的把問題解決。

一路上跌跌撞撞，也遇到硬體無法配合的問題，一部號稱有語音功能的 MODEM 在實際測試及使用上，居然完全送不出聲音，這使我們困擾了許久，連用三四台不同的語音 MODEM 也是不行，想著如果我們要換較好的 MODEM 才能來使用這程式的話，那是否意味著使用者也要購買一台更好的 MODEM？在老師的教導方面，我學到如何更有創意的來做一件事，一個創意可以讓你的程式更加活性化，對大家而言。

很榮幸能夠做這次的專題，也感謝一起努力過來的夥伴，沒有他，有些地方還不到怎麼開始動手。

方繼輝：

TAPI 的真正困難處是在於 COM 物件導向式的思考模式，以及一些 COM 運作的規則。雖然 TAPI 本身將電話處理部份包裝一個通用型態的元件，但是實際應用上還是有許多的限制以及 API 跟 modem 本上的溝通不良。另外現在要找到 TAPI-相容的 modem 實在是有點難。

原本是以為學長有做過類似的題目，在資源上，應該會比較豐富，卻完全沒想到，手邊僅有的參考資料還真是寥寥無幾。TAPI 不能算是新的東西了，但是 TAPI 3.x 此版本的出現卻採用了微軟提倡的 COM 技術。對於 Windows Programming 不熟稔的我，在此幾乎就是以敗北慘淡收場。所幸，花了將近整個暑假都在研究，漸漸的終於了解，也許要寫出一個純粹的 COM 元件讓人使用是一件困難的事，但是非常幸運地，我們不用寫 COM，只要懂得使用 TAPI 這個微軟寫好的 COM 元件就行了。

在所有目前的範例及應用上，TAPI 不能驅動 modem 來當做一個電話的接聽器，只能撥號出去。原因是必須 modem 廠商要符合 TAPI 的製造規格。而這些 modem 通常只有在美國的一些廠牌有支援。使得我們的專題走向只能藉由無遠弗屆的網路來完成。由網路來取代 PSTN 的撥號和接聽機制。同時也因此才加上視訊的功能。

這次寫程式真是學到了許多東西，也從其他同學和前輩那得到相當多的程式技巧。而且也因為這個題目，讓我第一次接觸到外國全球性的論壇以及 news group。也才能深刻的體會到語言的重要性。

4.3 應用與展望

現階段完成的功能都是比較陽春型的，希望以後能加以擴展，期待以後能加入以下部分

- (1) 利用新硬體達成整合電話接聽的功能
- (2) 加入語音壓縮：利用語音壓縮技術，把佔用的頻寬大幅縮減

既然是偏向應用性，當然更希望無線耳機和麥克風的誕生，讓這款應用軟體有真正價值的地方。

4.4 TAPI 本身限制及優缺點

TAPI 的優點

偏向高階的軟體語言以及整合性的 COM 技術介面，使得發展程式較為簡化，不用觸及實際的硬體層面和繁瑣的流程控制。

統一化的輸出介面，由於 Microsoft 的整合，將所有相關的 Telephony 應用程式都統一由 TAPI 來做連接的動作，使得程式元件可以重覆利用。例如 FAX 的軟體，在接駁 Modem 的 I/O 部份，便是使用 TAPI。

TAPI 的缺點

雖然 TAPI 可以使用統一的介面來驅動 Modem 或其他 Terminal，但是卻非常依賴硬體廠商所提供的 TSP 和 MSP 的驅動程式。現今多以美州地區支援的程度較高。

由於 TAPI 3.x 不能直接觸及硬體層面的溝通，所以要有效的 Device 控制會有相當的難度。例如我們在程式中加上去的視訊畫面，曾經抓到另一張顯示卡內建的電視 Cable 畫面，Tapi 並無法精確的控制來源。

4.5 曾經遭遇困難的解決方法及探討

不能針對 EventNotify 介面註冊失敗?

Modem 提供的 Driver 不支援 TAPI，更新到最新的 Driver。但不一定能相容，除非有微軟的認證，或是廠商有依照微軟提供的 TSP 標準去寫 Driver。

Modem 可以撥號，可以聽到對方聲音，但對方聽不到 Modem 送出的聲音?

Modem 要有支援 VOICE 的晶片和 TAPI 相容的 Driver 才可正常的送出聲音，否則對方只會聽到雜音或沒聲音。

我們曾經試過了 8 台的 Modem，從古老的 33.6K 一直到目前台灣買得到最新的蝙蝠俠系列的外接式 56K Data/Fax/Voice Modem。對於傳統電話線路的撥號及來電接聽的測試上都嘗到不少挫敗。一直以為是 TAPI 的問題到最後深入研究，一個一個問題去測試才發現問題出在 Modem 的 Voice 處理晶片的 Driver 出了問題。

到底是哪邊出了錯？這有點跟 TAPI 推出的時間有關。

在 TAPI 2.x 時代，TAPI 僅支援傳統的電話網路(PSTN)，也就是 Windows 95/98 時期推出，但受到的重視程度並不大。因為推出的 TAPI 標準必須有硬體廠商的配合，TAPI 才能發揮它的控制功能。在此時某些硬體廠商還是有作出支援 TAPI 2.x 的 Driver。

Windows 2000 推出時，微軟也一併推出了 TAPI 3.0 的版本。同時以 COM base 改寫 TAPI 2.x 的函式庫，而且加上了對 IP 網路的支援，使得使用 TAPI 可以不用去處理麻煩的 Win Socket 通到問題就可以做簡單的多點傳輸的網路電話會議。

即使如此，此時亦是 ADSL，Cable 等寬頻時代的來臨。56Kbps 的 Modem 逐漸淡出市場，而且被內建在主機版上的一塊晶片組取代。所以 56K Modem 支援 Voice 的晶片被拿掉了，當然一般的 Modem 功能一撥接上網則沒有問題。

因此到了 Windows XP 系統推出時，TAPI 已經升級到 3.1 版，更是大大的加強了網路方面的功能，開始亦趨複雜。而支援 TAPI 相容的 Modem 就更難找了。我們找到了幾款是有相容 TAPI 的 Modem，但是大都是美國廠商生產的 Modem，要購買需要一點肯花錢的魄力。

再者，一個比較實際的問題。使用 Modem 來銜接電話線路時，若是可以成功的撥打電話或接聽來電，需要將耳機和麥克風裝在 Modem 提供的輸出輸入孔上。若以我們題目而言，使用者帶著無線的麥克風或耳機在聽音樂的同時，這些裝置都是要裝在音效卡上的，若插在 Modem 上就聽不到音樂了。除非外接的裝置可以做到同時接兩種線路的功能。

另外一種解決方法，去使用幾種市面上所謂的 FAX 傳真電話答錄機軟體和聲稱支援 TAPI 的軟體，發現，大部份的廠商都是自己寫 UniModem 的驅動程式，即利用標準的 AT 指令集，但是一來大部份的 Connect Control 和 Event 處理還有音訊的輸出輸入便得自己動手寫程式處理了。自己寫 AT 指令集去驅動 Modem 也許是可行的方法，但是以目前我們組員的能力和時間無法完成，實在是可惜。

4.6 工作分配

資料收集：

方繼輝 50%

徐麒翔 50%

程式規劃：

方繼輝 65%

徐麒翔 35%

程式設計：

方繼輝 60%

徐麒翔 40%

偵錯 & 測試：

方繼輝 40%

徐麒翔 40%

其他人 20%

報告：

方繼輝 30%

徐麒翔 70%



參考文獻

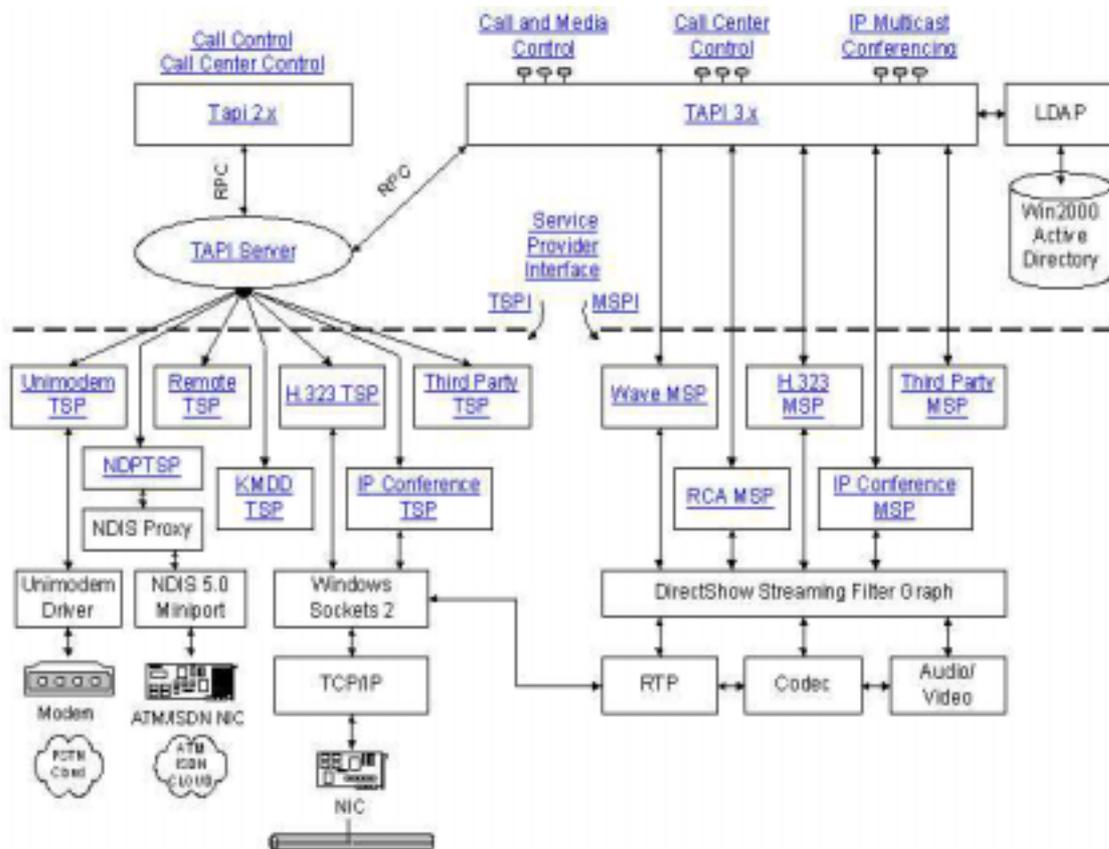
- [1]位元文化研究室，”精通視窗程式設計 適用Visual C++6.0”
文魁資訊，2002年7月
- [2]Charles Petzold著 余孟學譯，”Programing Windows 程式開發設計指南”，華彩股份有限公司，2000年6月
- [3]大新資訊，”MFC COM 程式設計教學教學“，p21~p166，基峯資訊，2000年四月
- [4]MircoSoft，”Telephony Application Programming Interfaces”，
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/tapi/tapi3/telephony_application_programming_interfaces.asp
- [5] Andreas Marschall' s TAPI and TSPI FAQ
[http://www.i-b-a-m.de/Andreas_Marschall' s_TAPI_and_TSPI_FAQ.htm](http://www.i-b-a-m.de/Andreas_Marschall's_TAPI_and_TSPI_FAQ.htm)
- [6]Voice Modem' s FAQ
<http://www.kengolf.com/en/voice-modem.htm>

附錄

這是我們再寫程式時，會參考到的東東

附錄1

TAPI架構的總覽



附錄一 TAPI架構總覽

附錄2 所需要的函式庫

- Advapi32.lib
- Amstrmid.lib (ActiveMovie™ GUIDs)
- Kernel32.lib
- Mdhcpid.lib (multicast GUIDs)
- Ole32.lib (COM)
- Oleaut32.lib (COM Automation)
- Rendid.lib (Rendezvous GUIDs)
- Rpcndr.lib
- Rpcns4.lib
- Rpcrt4.lib
- Sdpblbid.lib (Session Descriptor Protocol (SDP) GUIDs)
- Strmiids.lib
- User32.lib
- Uuid.lib
- Wldap32.lib
- Ws2_32.lib

附錄 3 錯誤代碼

TAPI_E_NOTENOUGHMEMORY

The buffer passed in to this method was not big enough. ((HRESULT)0x80040001L)

TAPI_E_NOITEMS

No items exist that match the request. ((HRESULT)0x80040002L)

TAPI_E_INVALIDMEDIATYPE

An invalid [media type](#) was passed in. ((HRESULT)0x80040004L)

TAPI_E_OPERATIONFAILED

The operation failed for an unspecified reason. ((HRESULT)0x80040005L)

TAPI_E_ALLOCATED

The device is already in use. ((HRESULT)0x80040006L)

TAPI_E_CALLUNAVAIL

No call appearance is available. ((HRESULT)0x80040007L)

TAPI_E_COMPLETIONOVERRUN

Too many call completions are outstanding. ((HRESULT)0x80040008L)

TAPI_E_CONFERENCEFULL

The conference is full and more members cannot be added. ((HRESULT)0x80040009L)

TAPI_E_DIALMODIFIERNOTSUPPORTED

The dial modifier is not supported. ((HRESULT)0x8004000AL)

TAPI_E_INUSE

The device is already in use. ((HRESULT)0x8004000BL)

TAPI_E_INVALIDADDRESS

The address is invalid or not properly formatted. For example, if address type is LINEADDRESSSTYPE_PHONENUMBER, this error code would mean that the data entered did not appear to be a real phone number.

((HRESULT)0x8004000CL)

TAPI_E_INVALIDADDRESSSTATE

The operation is not permitted in the current [address state](#). ((HRESULT)0x8004000DL)

TAPI_E_INVALIDCALLPARAMS

The [LINECALLPARAMS](#) structure is invalid. ((HRESULT)0x8004000EL)

TAPI_E_INVALIDCALLPRIVILEGE

The [call privilege](#) is invalid. ((HRESULT)0x8004000FL)

TAPI_E_INVALIDCALLSTATE

The operation is not permitted in the current [call state](#). ((HRESULT)0x80040010L)

TAPI_E_INVALIDCARD

The calling card is invalid. ((HRESULT)0x80040011L)

TAPI_E_INVALIDCOMPLETIONID

The call completion ID is invalid. ((HRESULT)0x80040012L)

TAPI_E_INVALIDCOUNTRYCODE

TAPI應用-插斷式視訊電話

The country code is invalid.	((HRESULT)0x80040013L)
TAPI_E_INVALIDDEVICECLASS	
The device class identifier is invalid.	((HRESULT)0x80040014L)
TAPI_E_INVALIDDIALPARAMS	
The dialing parameters are invalid.	((HRESULT)0x80040015L)
TAPI_E_INVALIDDIGITS	
The digits are invalid.	((HRESULT)0x80040016L)
TAPI_E_INVALIDGROUPID	
The group pickup ID is invalid.	((HRESULT)0x80040017L)
TAPI_E_INVALIDLOCATION	
The location ID is invalid.	((HRESULT)0x80040018L)
TAPI_E_INVALIDMESSAGEID	
The message ID is invalid.	((HRESULT)0x80040019L)
TAPI_E_INVALIDPARKID	
The park ID is invalid.	((HRESULT)0x8004001AL)
TAPI_E_INVALIDRATE	
The rate is invalid.	((HRESULT)0x8004001BL)
TAPI_E_INVALIDTIMEOUT	
The timeout value is invalid.	((HRESULT)0x8004001CL)
TAPI_E_INVALIDTONE	
The tone is invalid.	((HRESULT)0x8004001DL)
TAPI_E_INVALLIST	
An invalid list was passed as a parameter.	((HRESULT)0x8004001EL)
TAPI_E_INVALIDMODE	
An invalid mode was passed as a parameter.	((HRESULT)0x8004001FL)
TAPI_E_NOCONFERENCE	
The call is not part of a conference.	((HRESULT)0x80040020L)
TAPI_E_NODEVICE	
The device was removed, or the device class is not recognized.	((HRESULT)0x80040021L)
TAPI_E_NOREQUEST	
No Assisted Telephony requests are pending.	((HRESULT)0x80040022L)
TAPI_E_NOTOWNER	
The application does not have OWNER privilege on the call.	((HRESULT)0x80040023L)
TAPI_E_NOTREGISTERED	
The application is not registered to handle requests.	((HRESULT)0x80040024L)
TAPI_E_REQUESTOVERRUN	
The request queue is already full.	((HRESULT)0x80040025L)
TAPI_E_TARGETNOTFOUND	
The call handoff failed because the specified target was not found.	

TAPI應用-插斷式視訊電話

((HRESULT)0x80040026L)

TAPI_E_TARGETSELF

No higher priority target exists for the call handoff. ((HRESULT)0x80040027L)

TAPI_E_USERUSERINFOTOOBIG

The amount of user-user information exceeds the maximum permitted.

((HRESULT)0x80040028L)

TAPI_E_REINIT

The operation cannot be completed until all TAPI applications shut down and reinitialize.

((HRESULT)0x80040029L)

TAPI_E_ADDRESSBLOCKED

You are not permitted to call this number. ((HRESULT)0x8004002AL)

TAPI_E_BILLINGREJECTED

The calling card number or other billing information was rejected.

((HRESULT)0x8004002BL)

TAPI_E_INVALFEATURE

The device-specific feature is invalid. ((HRESULT)0x8004002CL)

TAPI_E_INVALBUTTONLAMPID

The button or lamp ID is invalid. ((HRESULT)0x8004002DL)

TAPI_E_INVALBUTTONSTATE

The button state is invalid. ((HRESULT)0x8004002EL)

TAPI_E_INVALIDDATAID

The data segment ID is invalid. ((HRESULT)0x8004002FL)

TAPI_E_INVALHOOKSWITCHDEV

The hookswitch device ID is invalid. ((HRESULT)0x80040030L)

TAPI_E_DROPPED

The call was disconnected. ((HRESULT)0x80040031L)

TAPI_E_NOREQUESTRECIPIENT

No program is available to handle the request. ((HRESULT)0x80040032L)

TAPI_E_REQUESTQUEUEFULL

The queue of call requests is full. ((HRESULT)0x80040033L)

TAPI_E_DESTBUSY

The called number is busy. ((HRESULT)0x80040034L)

TAPI_E_DESTNOANSWER

The called party does not answer. ((HRESULT)0x80040035L)

TAPI_E_DESTUNAVAIL

The called number could not be reached. ((HRESULT)0x80040036L)

TAPI_E_REQUESTFAILED

The request failed for unspecified reasons. ((HRESULT)0x80040037L)

TAPI_E_REQUESTCANCELLED

The request was cancelled. ((HRESULT)0x80040038L)

TAPI_E_INVALPRIVILEGE

TAPI應用-插斷式視訊電話

The application's [privilege](#) level is invalid for the current operation.

((HRESULT)0x80040039L)

TAPI_E_INVALIDDIRECTION

The [TERMINAL_DIRECTION](#) passed in was invalid.

((HRESULT)0x8004003AL)

TAPI_E_INVALIDTERMINAL

The terminal passed in was invalid for this operation. ((HRESULT)0x8004003BL)

TAPI_E_INVALIDTERMINALCLASS

The [terminal class](#) is invalid. ((HRESULT)0x8004003CL)

TAPI_E_NODRIVER

The service provider was removed. ((HRESULT)0x8004003DL)

TAPI_E_MAXSTREAMS

The maximum number of streams was reached. ((HRESULT)0x8004003EL)

TAPI_E_NOTERMINALSELECTED

The operation could not be performed because it requires terminals to be selected.

((HRESULT)0x8004003FL)

TAPI_E_TERMINALINUSE

The operation could not be performed because the terminal is in use.

((HRESULT)0x80040040L)

TAPI_E_NOTSTOPPED

The operation could not be performed because it requires the stream to be stopped.

((HRESULT)0x80040041L)

TAPI_E_MAXTERMINALS

The maximum number of terminals has been reached. ((HRESULT)0x80040042L)

TAPI_E_INVALIDSTREAM

The stream passed in was invalid for this operation. ((HRESULT)0x80040043L)

TAPI_E_TIMEOUT

The call failed because of a timeout. The timeout interval is two minutes.

((HRESULT)0x80040044L)

TAPI_E_CALLCENTER_GROUP_REMOVED

The ACD Proxy has removed this group. Operations on this object are invalid.

((HRESULT)0x80040045L)

TAPI_E_CALLCENTER_QUEUE_REMOVED

The ACD Proxy has removed this queue. Operations on this object are invalid.

((HRESULT)0x80040046L)

TAPI_E_CALLCENTER_NO_AGENT_ID

The Agent object was created with [CreateAgent](#) and has no ID. If an ID is needed, use [CreateAgentWithID](#).

((HRESULT)0x80040047L)

TAPI_E_CALLCENTER_INVALIDAGENTID

The agent ID is invalid. SULT)0x80040048L)

TAPI_E_CALLCENTER_INVALIDAGENTGROUP

The agent group is invalid. SULT)0x80040049L)

TAPI_E_CALLCENTER_INVALIDPASSWORD

The agent password is invalid. (ESULT)0x8004004AL)

TAPI_E_CALLCENTER_INVALIDAGENTSTATE

The agent state is invalid. (SULT)0x8004004BL)

TAPI_E_CALLCENTER_INVALIDAGENTACTIVITY

The agent activity is invalid. (ESULT)0x8004004CL)

TAPI_E_REGISTRY_SETTING_CORRUPT

The registry setting is corrupt. (ESULT)0x8004004DL)

TAPI_E_TERMINAL_PEER

The peer for one of these bridge terminals has already been assigned.

((HRESULT)0x8004004EL)

TAPI_E_PEER_NOT_SET

The peer for this bridge terminal must be set to complete this operation.

((HRESULT)0x8004004FL)

TAPI_E_NOEVENT

There is no event in the MSP's event queue. ((HRESULT)0x80040050L)

TAPI_E_INVALIDADDRESSTYPE

This address does not support the specified [address type](#). ((HRESULT)0x80040051L)

TAPI_E_RESOURCEUNAVAIL

A resource needed to fulfill the request is not available.

((HRESULT)0x80040052L)

TAPI_E_PHONENOTOPEN

The phone is not open.

TAPI_E_CALLNOTSELECTED

The specified call is not currently selected.

TAPI_E_WRONGEVENT

This information is not available for this type of event.

TAPI_E_NOFORMAT

The format is unknown.

TAPI_E_INVALIDSTREAMSTATE

The operation is not permitted in current stream state.

TAPI_E_WRONG_STATE

The operation requested is not permitted for the current state.

TAPI_E_NOT_INITIALIZED

The object has not been initialized.