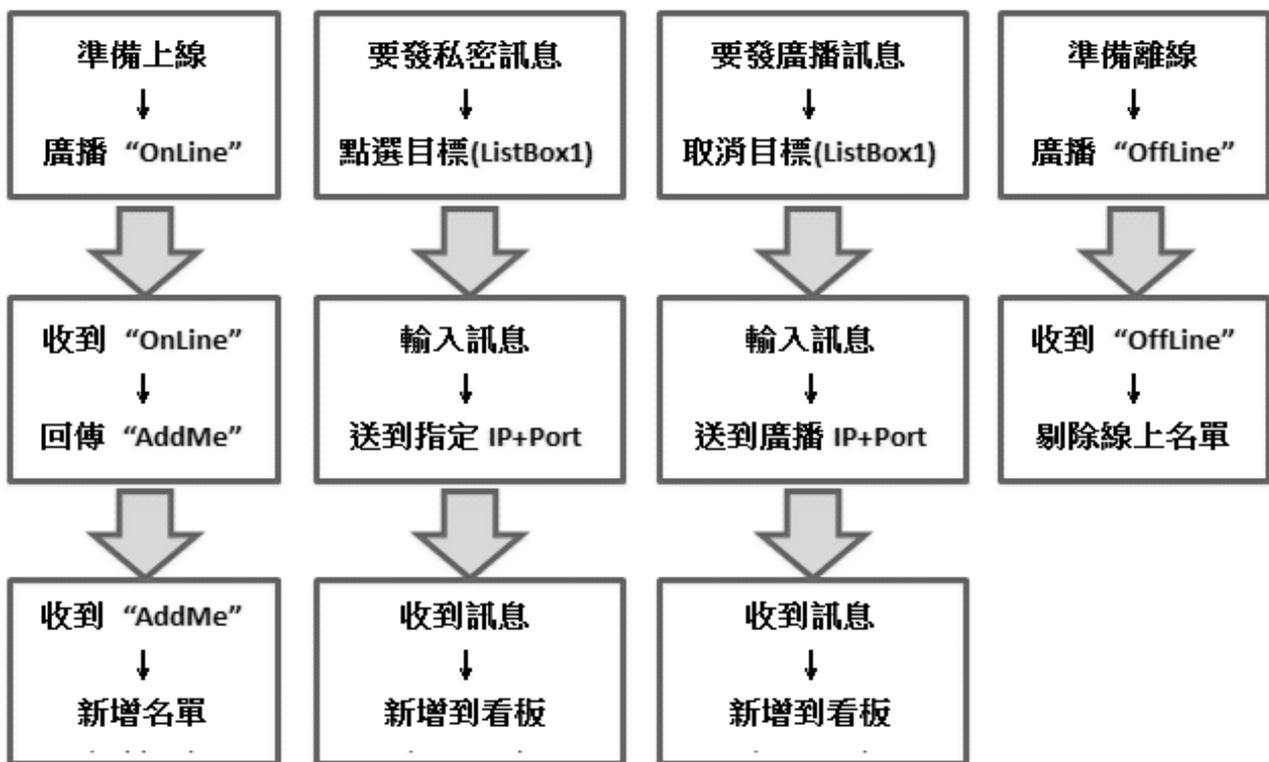


## 第五章 UDP 聊天室

### 5-1 UDP 聊天室通訊流程

聊天室的基本功能是可以讓很多人同時登入，也就是俗稱的『上線』。上線的人應該可以看到其他上線者的名單、可以指定通話對象、也能廣播訊息給所有線上的人，當然如果誰離線了也必須自線上名單移除。這類程式通常需要一個伺服器程式來管理線上名單的進出與轉傳訊息，但是直接使用 UDP 的區域網路廣播功能也能達到類似的目的，在不需要固定伺服器的情況下，一群使用者可以只用單一程式(客戶端)就達到聊天室的所有功能。下面是我們將設計的無伺服器 UDP 聊天室的主要通訊流程圖：



準備上線時就廣播一個“Online”訊息讓所有線上的人都知道你來了，收到這種訊息的客戶會自動回傳一個“AddMe”訊息，告知新上線人你的名稱與 IP，這樣剛上線的人就會陸續收到所有其他線上人的資訊，並用程式組織出自己的線上名單(顯示於一個 ListBox)。離線時情況類似，發出一個廣播訊息“OffLine”，所有人因此可以將你移出線上名單。要發私密訊息時就先指定名單成員之一，將訊息直接發給某個 IP，收到訊息者將訊息顯示出來即完成通訊。廣播時其實就是不指定對象的發言，程式會以廣播方式送出，所有線上使用者都能收到。

### 5-2 有關網路廣播的意義

在此的所謂『廣播』是區域網路的一個特殊功能，每一個網路區段都會保留一個特殊的 IP 位址，對著這個 IP 發訊息就會等於同時發出一樣的訊息給所有區域網路內的成員。這裡所謂

的區域網路或網路區段範圍決定於你的子網路遮罩(Subnet Mask)，如果你的遮罩是 255.255.255.0(最常見的狀況)，表示你的區網或網段範圍是 IP 前三碼相同，後一碼不同的各個電腦，而最後一碼 255 的 IP 就是廣播位址 IP，所以通常不會有一台電腦的 IP 最末一碼是 255 的。不過在此我們不必繼續深入學習這些網路技術的細節，可以直接使用既有的函數取得區網的廣播 IP 來寫程式。

經過這些說明你應該知道這裡所謂的『廣播』，範圍只限於一個區域網路之內，不會真的將訊息送給全世界的電腦！所以這個程式也只能在一個區域網路(譬如一間電腦教室或一棟宿舍)之內運作，算是實驗室版的程式範例。它的目的是讓我們經由較簡易的 UDP 程式模型，學習如何組織管理線上名單，以及挑選通訊對象的機制，這些都是多數網路通訊程式與線上遊戲設計的重要元素之一。

### 5-3 表單建置

請開啟一個程式專案，在表單 Form1 上建立以下介面：



上方輸入我是誰的文字方塊應該是 TextBox1，下方用於輸入訊息的是 TextBox2；『上線』按鍵是 Button1，『廣播』按鍵是 Button2；文字顯示為『IP』的是 Label4，稍後會用於顯示你的 IP；ListBox1 將用於顯示線上名單，ListBox2 用於顯示交談的歷史訊息，也就是一般說的聊天看板。

### 5-4 沿用的程式碼

本單元還是有很多可以自前面幾個單元沿用的程式碼，包括(1)匯入的命名空間：

```
Imports System.Net           '網路基礎功能相關函數
Imports System.Net.Sockets   '網路連線功能相關函數
```

```
Imports System.Text          '文字編碼功能相關函數
Imports System.Threading     '多執行緒功能相關函數
```

還有(2)取得本機 IP 的 localIP 副程序，與(3)關閉視窗時關閉監聽執行緒的 FormClosing 事件副程序：

```
'取得本機 IP (第一個合乎 IPv4規格的IP)
Function localIP() As String
    Dim ip() As IPAddress = Dns.GetHostEntry(Dns.GetHostName).AddressList '取得本機IP陣列
    For Each it As IPAddress In ip '逐一檢視IP字串
        If it.AddressFamily = AddressFamily.InterNetwork Then '如果是IPv4回傳此IP字串
            Return it.ToString
        End If
    Next
    Return "" '找不到合格IP回傳空字串
End Function

'關閉視窗
Private Sub Form1_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) _
Handles Me.FormClosing
    Try
        Send(BC, "OffLine", "") '宣告離線
        Th.Abort() '關閉監聽執行緒
        U.Close() '關閉監聽器
    Catch ex As Exception

    End Try
End Sub
```

## 5-5 公用變數宣告與表單載入

本範例的公用變數宣告與 Form\_Load 事件內容較之前幾個單元略有改變(粗體字部分)，請建立程式碼如下：

```
'公用變數
Dim U As UdpClient 'UDP監聽器
Dim Th As Thread   '監聽執行緒
Dim ips As New ArrayList '線上客戶IP列表
Dim Port As Short = 2013 '使用通訊埠
Dim MyName As String '我的線上名稱
Dim BC As String = IPAddress.Broadcast.ToString '廣播用IP
'必要設定與啟動監聽
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles MyBase.Load
    Label4.Text = "我的 IP: " + localIP() '顯示我的IP
```

```
Control.CheckForIllegalCrossThreadCalls = False '忽略跨執行緒操作的錯誤
```

```
End Sub
```

程式不同於前面幾個單元之處，首先是宣告一個用於記錄線上使用者 IP 的 ArrayList 集合物件 ips，ArrayList 集合物件與陣列相似，但是可以直接加入成員，或指定名稱移除成員，成員數目是隨時可以變動的！這種物件比固定大小的陣列更適合於線上名單的管理。接著是將監聽網路訊息用的通訊埠 (Port) 設定為 2013，變數 MyName 將代表我的線上名稱，然後用 IPAddress.Broadcast.ToString 的函數取得廣播用的 IP 字串變數 BC。

Form\_Load 事件中先將我的 IP 顯示於 Label4，接著宣告必要的忽略跨執行緒錯誤，我們之前是將此行放在啟動監聽的按鍵副程序之中，但是本例中會有使用者頻繁選擇上線或離線的可能，為避免重複多次執行這個動作，就將程式碼移到此處。

此處必須說明的是我們將之前幾個單元中可以自行設定的監聽用通訊埠 Port 設定為 2013，也未提供讓使用者修改的介面，表示這個程式只能使用 2013 這個 Port 互相通訊，寫好程式之後也無法如前幾個單元做本機內部不同 Port 之間的測試了！這其實才是一般應用程式應有的狀態，就是使用程式設定的 Port 互相通訊，你用 MSN 的時候程式應該不會問你要用哪一個 Port 吧？果真如此，使用者應該都會傻眼不知如何設定的！

所以這個程式寫好之後就只能與同區網內其他電腦中的程式實體互相測試了！以我的經驗，同間教室的同學們陸續完成這個程式，逐一上線到同一個自己建立的聊天室是非常讓人興奮的場面！上課不認真的同學此時通常會有著強烈被『疏離』的感覺！所以還是用功點比較好，而且是比較『好玩』的『好』！

## 5-6 發送訊息的程式

在此請先建立好用於發送訊息的副程序 Send，程式碼如下：

```
'發送訊息副程式
```

```
Private Sub Send(ByVal ToIP As String, ByVal msg As String, ByVal toWhom As String)
```

```
    '訊息格式：我是誰+IP+訊息+發給誰
```

```
    Dim A As String = MyName + ":" + localIP() + ":" + msg + ":" + toWhom
```

```
    Dim B() As Byte = Encoding.Default.GetBytes(A) '字串翻譯成位元組陣列
```

```
    Dim V As New UdpClient(ToIP, Port) '建立UDP物件
```

```
    V.Send(B, B.Length) '發送資料
```

```
End Sub
```

與之前幾個單元的訊息發送程式相當類似，但是發送對象的 IP 不再是來自固定的程式介面 (如 TextBox3 之類的)，事實上會與使用者在 ListBox 中點選的線上成員與對應的 ips 集合物件內容有關，所以在此將 IP 與發送對象名稱都經過副程序的引數 (ToIP 與 toWhom) 傳入，再組織成格式完整的訊息，然後臨時建立一個 UDP 物件，設定目標 IP+Port，最後將訊息發送出去，所謂的完整格式是這樣的：

**我是誰(MyName) : 我的 IP(localIP) : 訊息(msg) : 發給誰(toWhom)**

## 5-7 上線或離線按鍵

表單設計介面上的『上線』按鍵在此會同時用於『上線』與『離線』的動作，上線就是啟動監聽執行緒(Th.Start)，離線就是關閉監聽執行緒(Th.Abort)了。事實上我們之前的單元都是將離線動作與關閉視窗(FormClosing)結合，因此不可能只離線而不關閉程式。本單元的設計就比較有彈性了，Button1\_Click 的完整程式碼如下：

```
' 上線或離線選擇
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles Button1.Click
    If TextBox1.Text.Trim = "" Then '姓名未填寫
        MsgBox("請輸入姓名!")
        Exit Sub
    End If
    MyName = TextBox1.Text '我是誰
    ListBox1.Items.Clear() '清除名單
    ips.Clear() '清除IP記錄表
    If Button1.Text = "上線" Then '對外連線
        Th = New Thread(AddressOf Listen) '建立監聽網路訊息的新執行緒
        Th.Start() '啟動監聽執行緒
        Send(BC, "OnLine", "") '廣播上線訊息
        Button1.Text = "離線"
    Else '宣告離線
        Send(BC, "OffLine", "") '廣播離線訊息
        Th.Abort() '關閉監聽執行緒
        U.Close() '關閉監聽器
        Button1.Text = "上線"
    End If
End Sub
```

程式首先檢查你是否有輸入自己的名稱，雖然可以隨便取名，但是空的名稱無法列表，所以還是不允許的！接著清除線上名單，因為不管是離線或者剛剛上線，這個名單都應該是空白的。我們的設計是讓『上線』與『離線』的動作可以切換進行，Button1 顯示文字(Text 屬性)為『上線』時表示目前是離線，完成上線之後顯示文字會變成『離線』。相對的，顯示文字是『離線』時表示目前是上線的狀態，離線程式執行之後文字應該變成『上線』等待使用者再次連線。我們基本上用一個 If...Then...Else...的語法完成這個切換的選擇。

至於上線的動作就是要建立並執行監聽執行緒了，這與之前的單元內容相同，但是還必須發出一個廣播訊息"OnLine"，就是對大家宣告『我來了』！大家因此可以將你加入名單，也會將他們的資料自動回傳給你，讓你知道目前誰在線上？離線時當然就是關閉監聽執行緒，但是關閉之前必須先發出一個廣播訊息"OffLine"給大家，就是說：『我不玩了！』其他人知道後才能從名單中將你移除，繼續保持線上名單的正確性。

## 5-8 選擇訊息發送對象或廣播的機制

上線之後會做的事情除了被動接受訊息就是主動發言了！此時先要處理『發言給誰』的問題，選擇特定對象的動作就是選取線上名單，也就是 `ListBox1` 中的某一個人。稍後發訊時會依據這個選項的次序取得存在 `ips` 集合中這個人的 IP 值，來建立 UDP 通訊物件。這個『選擇』的動作是預設的功能，我們不需要在 `SelectedIndexChanged` 之類的事件副程序中寫任何程式。

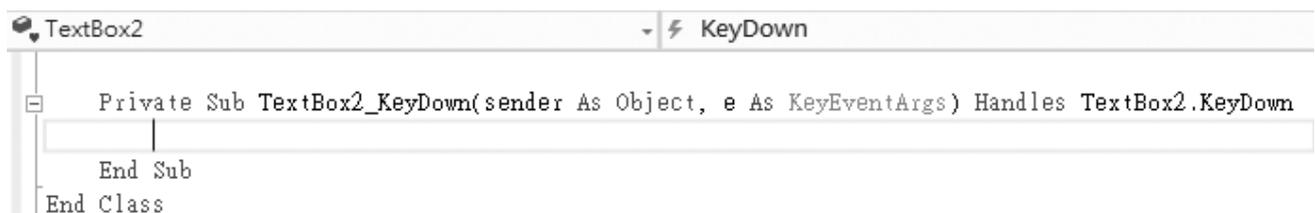
但是如果我們要發送廣播訊息時 `ListBox1` 中的選項必須是『沒有選擇』的狀態，此時就無法由 `ListBox1` 自己的操作完成這個『取消選取』(`ClearSelected`)的動作，因此才需要一個『廣播』按鍵(`Button2`)的參與，程式碼如下：

```
'清除選取，預備廣播或離線
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles Button2.Click
    ListBox1.ClearSelected()'清除選取項目
End Sub
```

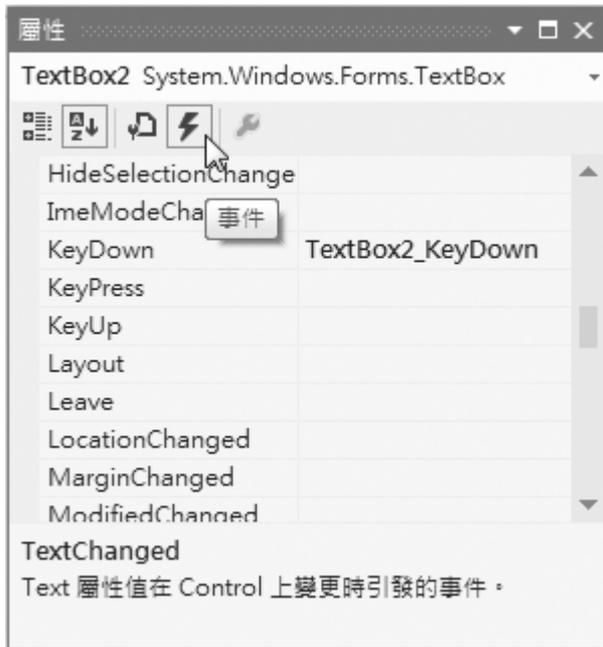
## 5-9 發送自訂訊息的程式

發送自訂訊息就是在表單介面的 `TextBox2` 中輸入訊息然後發送出去，依照一般聊天室程式的習慣，在輸入字串最後按下 `Enter` 就是表示要發送訊息了！在之前的 UDP 即時通程式是使用一個發送的 `Button` 按鍵送出訊息的，如果要改用 `Enter` 傳送表示我們必須在輸入文字的過程中持續監視使用者是不是按下了 `Enter`？因此程式必須寫在 `TextBox2` 物件的 `KeyDown` 事件之中，這個事件可以檢查每一次鍵盤按下的動作，看看是按了哪一個鍵(`KeyCode`)？

但是 `KeyDown` 並不是 `TextBox` 物件的預設事件，所以當你雙按 `TextBox` 時出現的應該是 `TextChanged` 事件而不是 `KeyDown`，所以你應該在程式碼頁面的左上方的表單物件下拉式選單中先選擇 `TextBox2` 物件，再到右上方的事件下拉式選單中選擇 `KeyDown` 來產生正確的事件副程序框架，如下圖：



或者先切換到設計頁面，在屬性視窗上方點選閃電標誌，切換到事件選單選擇 `KeyDown` 事件，一樣可以產生正確的程式框架：



程式框架出現後接著輸入程式碼如下：

'發送自訂訊息

```
Private Sub TextBox2_KeyDown(ByVal sender As Object, ByVal e As Windows.Forms.KeyEventArgs) _
    Handles TextBox2.KeyDown
```

```
    If e.KeyCode = Keys.Enter Then '按下Enter
```

```
        If ListBox1.SelectedItem Is Nothing Then '未選擇發訊對象
```

```
            Send(BC, TextBox2.Text, "") '廣播訊息
```

```
        Else
```

```
            '發送私密訊息
```

```
            Send(ips(ListBox1.SelectedIndex), TextBox2.Text, ListBox1.SelectedItem)
```

```
            ListBox2.Items.Add(MyName + "(私密):" + TextBox2.Text) '訊息寫入看板
```

```
        End If
```

```
        TextBox2.Text = "" '清除訊息填寫欄
```

```
    End If
```

```
End Sub
```

首先就是檢查鍵盤碼(**e.KeyCode**)是否為 **Enter(Keys.Enter)**的判斷式，如果是！就執行傳送訊息的動作；如果不是的話表示使用者還在輸入訊息內容，暫時不需動作。執行傳送前還必須判斷發訊對象是個人(私密訊息)還是全體(廣播)？判斷準則是 **ListBox1** 如果有已經選擇的項目就是私密訊息，如果沒有就是廣播。廣播時將訊息傳到廣播用的 IP(就是 **BC**)，接收者名稱是空字串；否則就是用 **ListBox** 選項為發送對象名稱(**ListBox1.SelectedItem**)，並以對應 **ListBox1** 被選次序(**ListBox1.SelectedIndex**)的 IP 集合物件 **ips** 的成員為發送的目標 IP，使用之前做好的 **Send** 副程序傳出訊息。

當然不論傳送何種訊息，傳完就將 **TextBox2** 的文字內容清除，同時也寫入看板(**ListBox2**)，讓使用者感覺到訊息已經送出。有點奇怪的是：為何廣播訊息在此並沒有寫入 **ListBox2** 的程式

碼？原因是廣播訊息不僅別人都能收到，自己也一樣會收到，所以在稍後的 Listen 副程序中接收訊息時就會寫到 ListBox2 了，發送時如果也寫一次就會看起來是重複的訊息。

## 5-10 接收與處理訊息的 Listen 副程序

接下來就是製作訊息接收與處理的程式，也是本範例程式的資料處理核心！這部分程式與之前的單元外層部分程式碼還是相同的，只有粗體字部分是本單元範例的獨特處理程序。完整程式碼如下：

```
'監聽副程式
Private Sub Listen()
    U = New UdpClient(Port) '建立UDP通訊物件
    Dim EP As IPEndPoint = New IPEndPoint(IPAddress.Parse(localIP), Port) '接聽端點(IP+Port)
    Do While True
        Dim B() As Byte = U.Receive(EP) '讀取網路資訊到B陣列，如果沒有訊息進入，程式在此等待
        Dim A As String = Encoding.Default.GetString(B) '翻譯資訊為字串
        '切割訊息為：C(0)=發訊者；C(1)=IP；C(2)=訊息；C(3)=發訊對象
        Dim C() As String = A.Split(":")
        Select Case C(2) '依據訊息內容處理
            Case "OnLine" '客戶上線訊息
                ListBox1.Items.Add(C(0)) '名稱加入列表
                ips.Add(C(1)) 'IP加入集合物件
                If C(0) <> MyName Then Send(C(1), "AddMe", C(0)) '回應我也在線上
            Case "AddMe" '接收誰也在線上訊息
                ListBox1.Items.Add(C(0)) '名稱加入列表
                ips.Add(C(1)) 'IP加入集合物件
            Case "OffLine" '客戶離線訊息，移除名單
                ListBox1.Items.Remove(C(0))
                ips.Remove(C(1))
            Case Else '一般訊息
                If C(3) = "" Then '公開訊息(無指定收訊者)
                    ListBox2.Items.Add(C(0) + "(公開):" + C(2))
                Else '私密訊息(有指定收訊者C(3))
                    ListBox2.Items.Add(C(0) + "(私密):" + C(2))
                End If
            End Select
        Loop
    End Sub
```

請先參考 Send 副程序的內容，可以知道我們送出的訊息格式都是這樣的：

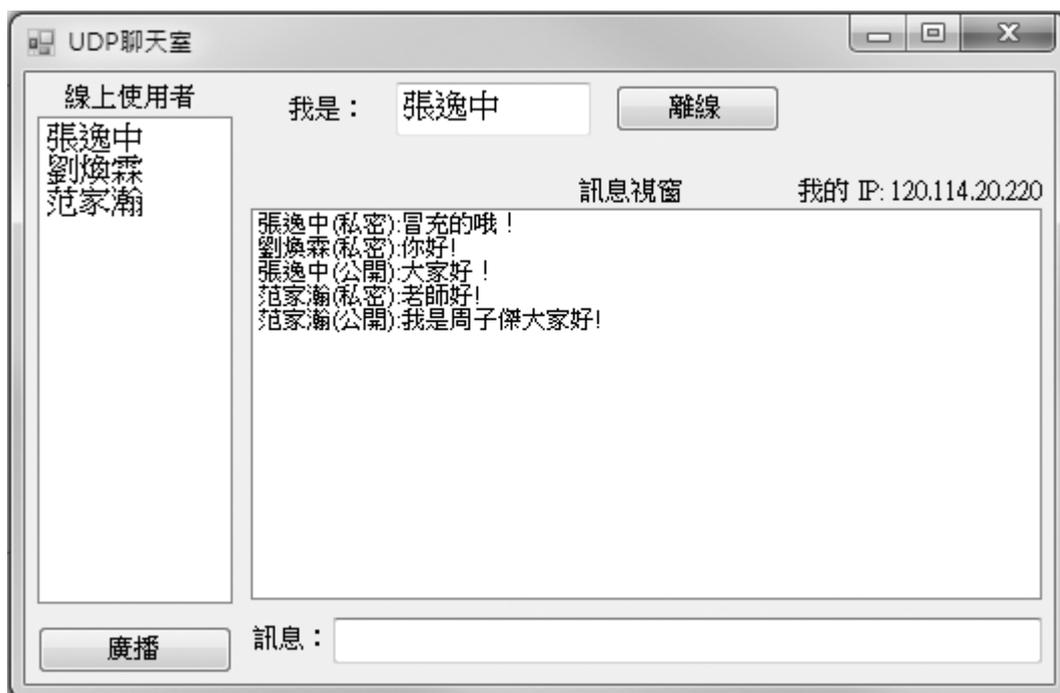
**我是誰(MyName)：我的 IP(localIP)：訊息(msg)：發給誰(toWhom)**

所以處理接收訊息時，首先以冒號將訊息切割成字串陣列 C。如果訊息格式正確，C(0)舊式發信者的名稱，C(1)是發訊者的 IP，C(2)是訊息內容，C(3)是收件者的名稱。首先我們以訊息

內容分流(Select case)處理：

- (1) 收到“OnLine”時表示發訊者剛剛上線，須將此人的名稱加入 ListBox1，IP 加入 ips 集合物件，同時要發出一個“AddMe”的廣播訊息，將自己的名稱與 IP 告訴此人。
- (2) 收到“AddMe”時表示別人通知你：他在線上，應該將此人的名稱加入 ListBox1，IP 加入 ips 集合物件。
- (3) 收到“OffLine”時表示發言者要離線了，須將他的名稱從 ListBox1 移除，IP 從 ips 移除。
- (4) 以上皆非的內容應該就是一般的聊天訊息了！此時如果收件者是空字串表示為廣播訊息，否則就是私密訊息了！這些都要適當標示然後寫在 ListBox2 的看板上。

當然，前面三種與線上名單控制有關的訊息(OnLine, OffLine & AddMe)是不必上看板的！至此程式就全部完成了！你可以邀請另一位在鄰近電腦製作程式的同學互相測試，或者自己設法將程式拷貝到另一台區網內的電腦測試。下圖是我在研究室裡與幾位製作專題的學生用三台電腦同時上線的測試畫面：



## 5-11 課後討論

這個範例之中最特殊也最有趣的部份應該就是網路廣播，原來程式可以同時對所有人發出訊息，這挺新鮮的！但是仔細想想這不奇怪，如果你的物理觀念還可以，就會知道當我們發出任何網路訊息時，其實是將代表 0 與 1 的高低電位組成的電子訊號像電波一樣傳到大家公用的網路線上，此時所有區域網路內被實體網路線連接的電腦都會收到這些訊息，也就是網路封包。

直接收到這些訊息的其實是你的網路卡，網卡收到封包之後會檢視它的收件者，如果不是

給本機電腦這個 IP 的就會被忽略，被指定給本機 IP 或廣播的封包當然就收下了！所以你到電腦後面看看網路卡時，不管你自己有沒有上網，上面的燈號總是閃個不停，意思是只要你的區域網路內有任何封包在傳遞時網卡都是一直在忙碌的！

那麼我們的網路線實際上也真的有連到全世界大部分上網的電腦，是不是我的封包就會經過全世界的每一台電腦呢？答案是不會的！否則每張網卡都要同時檢視全世界的網路封包，那多忙啊！事實上是網路上有很多的閘道器(Gateway)，他們有如執行交管的交通警察，可以管制封包的傳遞，只是在區網內部通訊的封包就不會穿過這個設備出去環遊世界，只有指定要到區網外面的封包，譬如連上某個網站的要求，才會被放行！

甚至還有一些規則可以讓迷途的封包不會永遠在網路世界流浪！譬如已經經過很多個轉接設備(通常稱為路由器 Router)，還無法傳到目的地時封包就會被銷毀。這種機制也叫作生命週期(Lifetime)，也就是說封包通常一出生的時候就開始倒數計時，如果在時間終了前到達目的地當然封包就會功成身退，如果始終傳不到指定的地方，應該就是目的地不存在或地址寫錯，應該是傳不到了，此時就應該自我了斷！這樣才不會有一些地址錯誤的封包永遠流浪在網路上。

經過前面幾個單元的範例時做，你應該已經對於網路通訊的基本元素與程式技術有了基本的認識。但是我們一開始就說過 UDP 並不是商業化網路程式用於建立穩定通訊的方式，真正網路程式中最大咖的主角是 TCP 模式的通訊，我們接下來就會詳細的介紹具有伺服器端與客戶端程式的 TCP 模式通訊架構與多個運作實例。