

# 階層式 SIP 網路多媒體會議嵌入式系統之設計與實作

許晁瑋、盧俊佑、蔡孟倩、王楓宜、蘇暉凱\*  
南華大學資訊工程學系  
hksu@mail.nhu.edu.tw\*

**摘要** — 隨著高速網際網路與網路電話技術進步，網路電話與多媒體會議服務已開始公眾化，但伴隨著使用需求加大，傳統集中式 SIP 多媒體網路會議架構容易造成系統的過度負擔，因此集中式 SIP 多媒體網路會議系統已無法滿足系統規模成長之需求。本論文參考集中式會議架構，將 Video Mixer 模組化，提出階層式 SIP 多媒體會議系統，以分散視訊資料處理之負載，並提高多媒體會議通訊品質。最後，本論文以 IXP 465 嵌入式系統平台發展本系統雛型，驗證系統功能性與可行性。<sup>1</sup>

## 一、前言

隨著網際網路技術進步，加快了人們前進的腳步，並且也縮短了人與人之間的距離，無論是數據交換與即時影音通訊，網際網路都可以提供低成本、高存取度的服務。VoIP 即是一種可以在 IP 網路上互傳音訊或視訊的一種技術，並以 VoIP 技術為基礎，實現多人通訊的網路多媒體會議服務[1]。網路多媒體會議提供一種虛擬會議環境，讓來自各地的會議參與者可以遠距離通訊，交換語音、影像、數位檔案...等多媒體資料，就好像大家面對面坐在同一個會議桌開會一樣。

寬頻網路技術的進步，使 VoIP 服務品質大大提升，加上低通訊成本、多元化應用整合的好處，也讓高傳輸品質的即時多媒體應用普及化。過去數年間，IP 通訊市場一直以 H.323 通訊協定為主流。但由於 H.323 的規定太嚴謹、擴展性不佳，然而 SIP (Session Initiation Protocol) 協定擁有容易開發、內容簡潔以及擴充性高、相容性佳之優點，並且容易與其它網際網路應用整合，因此，帶動 SIP 技術快速發展，將 SIP 即時訊息、網路電話與視訊會議服務應用於有線網路、無線網路、3G/4G 通訊網路等環境。

SIP 網路會可分為預約會議 (Reserved Conference) 與隨意會議 (Ad-hoc Conference)。預約會議必須事先預約，預約參加會議的成員人數、會議室位址與開會時間，屆時再由參與者自行加入網路會議室。隨意會議是由會議發起人，動態邀請參與者加入會議，撥打網路電話模式跟一般撥打電話一樣，因此使用彈性佳且友善度高，也較受使用者接受。但相對地，隨意會議在會議資源動態分配技術部分，也較為複雜。

無論是預約會議服務或隨意會議服務，大部分都是採用集中式會議服務架構，將會議控制與會議資源集中管理，可以確保會議服務品質，但集中式架構容易造

成系統負載集中，超過系統負荷而出現視訊延遲或音訊斷訊等問題，影響整體會議服務品質與效益。

[2]已透過 Open Source Asterisk，在 Linux PC 環境實現集中式多媒體會議服務。但經過實際測試後，觀察到混影像高負載的問題。因為該系統是以 CPU 運算，用軟體方式進行 Video 與 Audio 混合，所以當處理一個有三個成員會議的混音、混影像工作時，CPU 負載即時達到 100%。因此，對於多人多會議的環境，會議通訊品質穩定性將會快速下降。

因此，本論文將負載最重的 Video Mixer 模組化與分散化，提出階層式 SIP 多媒體會議系統，改良傳統集中式多媒體會議缺點，並增加 URI-list 功能實現隨意會議，以複雜的隨意會議驗證本系統之 Video Mixer 資源動態分配之效益。最後，以 IXP 465 嵌入式系統實現與驗證系統可行性。本論文具備集中式管理的方便性，並可整合分散式 Video Mixer 之資源，降低系統的負載，進而提供穩定的通訊品質，增加系統穩定性。

## 二、背景

### 2.1 多媒體會議技術發展

IETF 組織近幾年，已完成了許多網路電話協定標準制定；目前也正積極發展的 SIP 多媒體會議架構與會議控制協定標準，並定義 Focus 與 Mixer 兩個元件。

- Focus：負責多媒體會議建立、釋放、控制與管理。
- Mixer：提供聲音與影像混合功能。

SIP 多媒體會議模型主要可分為兩種，分散式架構 (Loosely Coupled Model) 與集中式架構 (Tightly Couple Model)。分散式會議架構將 Focus、Mixer 設計在各個 UA (User Agent) 上，並應用於多點傳播的網路傳輸環境上。其優點在於可節省資源，不需額外配置高階伺服器設備，即可達到多人會議之目標。但在多人會議連線時，由於每個 UA 擁有各自的 Focus 及 Mixer 資源，且每一端的網路環境及終端機處理能力皆有不同，因此將造成會議品質不穩定、資源管理不易、安全性不佳等問題；另一方面，此技術門檻較高，目前發展進度相對也比較緩慢。

集中式會議架構，將 Focus、Mixer 集中在會議伺服器上，會議的進行與離開必須由會議伺服器管理與控制。在此前提下的會議品質及安全控管都有一定程度的保障。然而，會議人數增加，將造成會議伺服器的負擔加大；但就資源管理層面而言，此架構機制運作簡單，可行性較高。除系統架構外，IETF 也發表一系列 SIP 多媒體會議相關規範，如：會議需求[3]、多媒體會議

<sup>1</sup>本研究由國科會贊助，計畫編號 NSC-96-2221-E-343-001。

架構[4]、會議控制協定[5]、會議發言控制權需求[6]、BFCP[7]、BFCP 串流的 SDP 格式[8]等。

## 2.2 嵌入式系統介紹

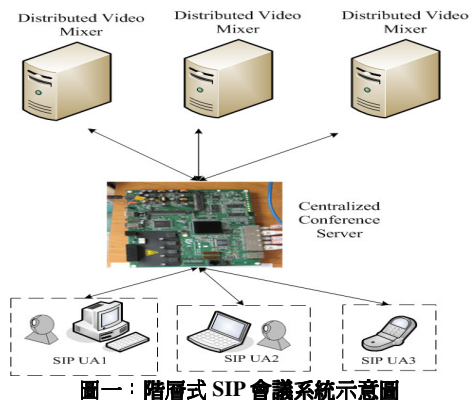
依據英國電機工程師協會定義：「嵌入式系統為控制、監視或輔助設備、機器或甚至工廠運作的裝置」。在新興的嵌入式系統產品中，常見的有手機、PDA、GPS、Set-Top-Box 或是嵌入式伺服器 (Embedded Server) 及精簡型終端設備 (Thin Client)。

由於本系統需要大量 CPU 運算，以及處理混音與處理影像多媒體資料工作，因此我們選用 Intel IXP 465 嵌入式系統實現；IXP 465 嵌入式系統提供 667MHz CPU 處理能力、128MB RAM，且具備完整有線與無線網路介面功能[9]，因此可以處理 Focus 與簡單 Mixer 工作。在嵌入式系統有限資源情況下，透過實作與測試，驗證本系統之功能性與產品化可行性。

## 三、系統架構

### 3.1 系統簡介

本論文採用 Open Source Asterisk 作為系統的基礎架構，並重新設計 Centralized Conference Server、Distributed Video Mixer 與 SIP UA 三個元件，以實現階層式 SIP 多媒體會議系統。



圖一：階層式 SIP 會議系統示意圖

圖一為本論文所提出之階層式 SIP 會議系統架構，元件說明如下：

- **Centralized Conference Server :**  
提供 Focus 與 Audio Mixer 服務；除了一般 Focus 與 Audio Mixer 功能外，當 Centralized Conference Server 接收到會議發起者之隨意會議邀請時，它可以從 INVITE 訊息中，知道該邀請哪些參與者加入此會議，以及此會議共有多少人數，最後透過 Video Mixer 資源分配機制，決定出最合適的 Distributed Video Mixer 來負責此會議影像混合處理，並將此會議之所有影像串流導向到該 Video Mixer。
- **Distributed Video Mixer :**  
負責影像畫面切割與重組，將會議參與者的影像混合到同一畫面，讓所以參與者可以看到彼此之影像畫面。在本系統中，Centralized Conference Server 可以管理與控制多台 Distributed Video Mixer 之資源，透過 Video Mixing 資源動態分配，

我們可以將系統視訊處理的負載分散到所有 Distributed Video Mixer。

- **SIP UA :**

除了一般 UA 功能外，SIP UA 必須具備處理 URI-list 與 SIP MIME Content-Type 之能力，以實現隨意會議。

### 3.2 問題定義

本論文將 Centralized Conference Server 和 Distributed Video Mixer 分開設計，並使用一個 Centralized Conference Server 管理多個 Distributed Video Mixer。例如：當 UA1 要邀請 UA2 跟 UA3 加入會議，UA1 會發送一個 INVITE 訊息給 Centralized Conference Server，然後 Centralized Conference Server 必須選擇負荷最輕的 Distributed Video Mixer 來負責此會議影像混合工作；並且邀請 UA2 跟 UA3 加入會議，將所有成員的影像串流導向給負載輕的 Distributed Video Mixer 處理。如果有新的會議被要求，Centralized Conference Server 必須再分配另一個負載最輕 Distributed Video Mixer 來服務該會議。

在此架構中，我們將要解決以下三個問題來達到系統目標：

1. SIP 通訊協定功能性與相容性支援的問題；
2. 以有限資源嵌入式系統實現之可行性；
3. 如何分配較佳的 Distributed Video Mixer 來處理會議影像混合作。

因此，為了解決這三個問題，本論文擴充 SIP 功能，加強 SIP URI-List 功能以增加 UA 和 Centralized Conference Server 之間攜帶會議成員資訊之能力，並且引入 CVM Protocol 解決 Centralized Conference Server 與 Distributed Video Mixer 之間的通訊能力；此外，本系統採用 IXP 465 為主，PC 為輔的方式，以 IXP 465 實現 Centralized Conference Server 與部分 Distributed Video Mixer，以 PC 實現部分 Distributed Video Mixer 與 SIP UA，解決嵌入式系統可能面臨的硬體上資源不足的問題；最後再透過 Distributed Video Mixer 資源管理機制與資源分配演算法的設計，動態分配最佳 Distributed Video Mixer 資源。

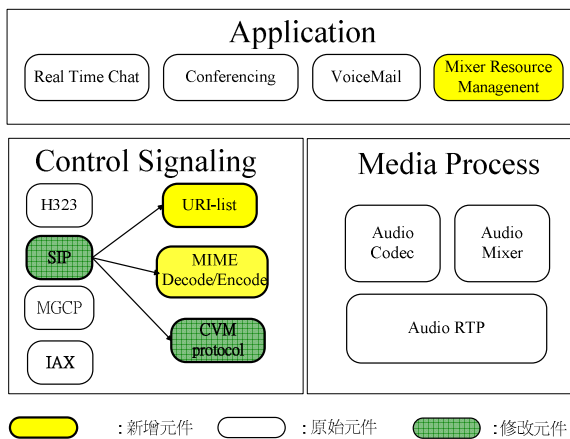
## 四、元件設計

### 4.1 Centralized Conference Server 之設計

#### 4.1.1 功能設計

本論文在 Centralized Conference Server 功能方塊圖如圖二所示，我們採用 Open Source Asterisk [10] 作為基礎架構。現有 Asterisk 已提供完整 Focus 與 Audio Mixer 功能。為提供隨意會議，以及達到動態 Video Mixer 資源分配，分散 Video Mixer 負載，我們必須加強 SIP 模組以攜帶 MIME Content-Type 的處理能力，並加入 URI-list 功能用來攜帶會議成員的資訊；此外，我們必須增加 Mixer Resource Management 機制，以及修改 CVM Protocol 以達到動態控制多台 Distributed Video Mixer 功能。

Centralized Conference Server 修改與新增功能說明如下：



圖二：Centralized Conference Server 功能方塊圖

```
<?xml version="1.0" encoding="UTF-8" ?>
<?Resource-lists xmlns="urn:ietf:params:xml:ns:resource-list"
xmlns:cp="urn:ietf:params:xml:ns:copycontrol" >
  <list>
    <entry uri="sip:monc@example.com" cp:copyControl="to" />
    <entry uri="sip:maggie@example.com" cp:copyControl="cc" />
    <entry uri="sip:mc@example.org" cp:copyControl="bcc" />
  </list>
</resource-lists>
```

圖三：URI-List 格式範例

- MIME Decode/Encode :  
Focus Server 收到 INVITE 訊息時，如果具有多重本體 (MIME Content-Type)，也就是 INVITE 除了 SDP 協商多媒體的本體之外，還有 URI-list 本體，而 Centralized Conference Server 為了區分這兩個本體的不同，使用 MIME 去解碼，藉此以分出 SDP 本體與 URI-list 本體。
- URI-list :  
會議發起者會將欲邀請加入會議成員的 URI，放在所傳送建立會議 INVITE 夾帶的 URI-list 中，告知 Centralized Conference Server 該會議有哪些會議成員；URI-list 是用 XML resource-list 來表示會議成員 URI。因此，Centralized Conference Server 須具備解讀 XML 的能力，從 URI-list 將每位需要邀請的成員 URI 逐一解析出來再做邀請的動作。URI-List 的格式如圖三所示，由兩大部分組成：一是根據 IETF 以 XML (Extensible Markup Language) 的「resource-list」為格式，列出初始成員的 URI 成為 URI-list 訊息本體；二則是使用 XML 擴充 Copy Control 屬性，加在 URI 後面用來描述 URI 的屬性，如圖三內「copyControl」。
- SIP :  
除了必須修改 SIP 可以支援 MIME 的能力外，我們還必須讓 Focus 可以增加與修改 SIP Content。當 Centralized Conference Server 收到 URI-list 並且解析出每位需要邀請至會議成員的 URI 之後，Conference Server 除了要建立會議之外，還要主動去邀請 URI-list 的成員，因此 Centralized Conference Servers 須具備主動送出 Outgoing INVITE 給每位成員之能力。

- CVM Protocol :  
Centralized Conference Server 透過 CVM (Conference Video Mixer) Protocol 與 Distributed Video Mixer 通訊，並控制 Distributed Video Mixer 進行視訊串流混合建立與釋放。

#### 4.1.2 Distributed Video Mixer 資源分配機制

Centralized Conference Server 另一個設計重點就是 Mixer Resource Management 之資源分配機制設計，其目的就是在階層式 SIP 會議系統中，動態管理 Distributed Video Mixer 資源，達到分散 Video Mixer 負載之目標。

在 Video Mixer 資源分配機制中，我們定義兩個計算式，數學式 (1) 與 (2)，相關參數如表 I 所示。

表 I  
計算式參數定義表

參數	定義說明
M	系統中的 Video Mixer 集合，為已知條件。
R	系統進行中會議之集合，為已知條件。
$c_i$	Video Mixer $i$ 可處理的串流個數，此參數由系統給定， $i \in M$ 。
$n_j$	Conference $j$ 中的成員個數，此系統參數由 URI-List 決定， $j \in R$ 。
$x_{ij}$	表示 Conference 與 Video Mixer 之間的對應關係為一已知的矩陣；當 $x_{ij}=1$ ，表示 conference $j$ 作用在 Video Mixer $i$ 上；如果 $x_{ij}=0$ ，則 conference $j$ 沒有作用在 Video Mixer $i$ 上， $i \in M$ ， $j \in R$ 。
$n'$	欲建立之新會議成員個數。

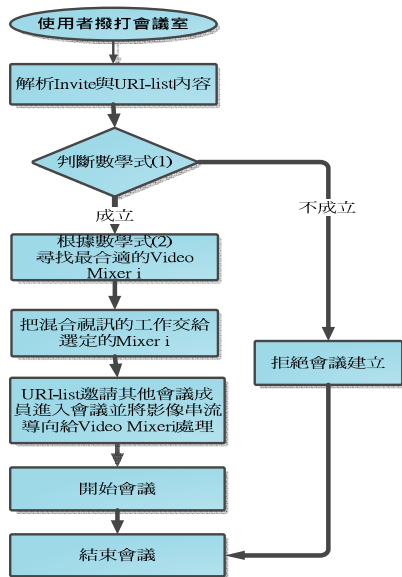
數學式 (1) 為條件式子，用來判斷欲建立之新會議成員個數  $n'$ ，系統中是否有任一 Distributed Video Mixer 可以滿足該視訊混合需求。在滿足數學式 (1) 的條件下，透過數學式 (2)，挑選出負載 (資源使用率) 最小的 Distributed Video Mixer  $i$  來負責新會議的視訊混合作。當最小負載的 Mixer 存在多台時，系統會以 Mixer ID 最小為優先選擇。

$$n' \leq c_i - \sum_{j \in R} x_{ij} n_j, i \in M \quad (1)$$

$$\min_{i \in M} \frac{\sum_{j \in R} x_{ij} n_j}{c_i} \quad (2)$$

Video Mixer 資源分配機制處理流程圖如圖四所示，當一會議發起者欲邀請參與者建立多媒體會議時，會議發起者會送出挾帶 URI-list 之 INVITE 請求訊息給 Centralized Conference Server，Centralized Conference Server 除了處理一般 Focus 工作外，亦解析 INVITE 訊息內容與 URI-list 資訊，並根據 URI-list 得到參數  $n'$ 。接著，根據我們定義的數學式 (1)，判斷條件是否成立。如果不成立，表示目前所有 Distributed Video Mixer 皆無能力處理該新會議視訊混合作，因此必須拒絕開會議建立之請求；如果成立，表示系統中最少會有一個 Distributed Video Mixer 可以處理該新會議視訊混合，因

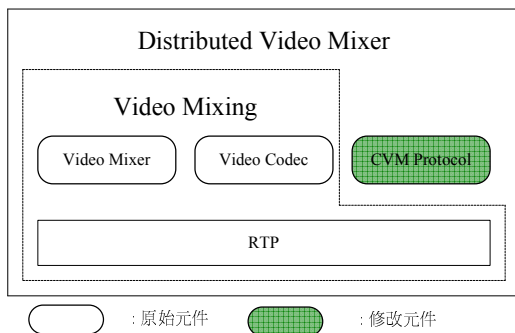
此接著必須根據數學式 (2)，挑選出最合適的 Distributed Video Mixer  $i$  負責該會議視訊混合。最後，再根據 URL-list 去邀請 list 中的所有成員加入該會議，並將視訊串流導向給 Distributed Video Mixer  $i$  處理，開始多媒體會議通訊。



圖四：Distributed Video Mixer 資源分配機制處理流程圖

#### 4.2 Distributed Video Mixer 之設計

本系統之 Distributed Video Mixer 是以 Open Source CONFIDANCE Video Mixer [11]為基礎來開發，功能方塊圖如圖五所示，主要功能說明如下：



圖五：Distributed Video Mixer 功能方塊圖

- Video Mixing : Distributed Video Mixer 的 Video Mixing 功能主要採用 Confiance Video Mixer 所提供的，CONFIDANCE Video Mixer 是以軟體方式實現視訊影像串流壓縮、解壓縮、視訊畫面大小調整、視訊畫面切割與重組，因此視訊品質可接受，且需大量系統 CPU 運算資源。
- CVM Protocol : Distributed Video Mixer 透過 CVM Protocol 與 Centralized Conference Server 作溝通，現有 CONFIDANCE Video Mixer 與 Asterisk 只支援一對一配置，因此為達到一台 Centralized Conference

Server 支援多台 Distributed Video Mixer，並且支援 Video Mixer 動態配置，我們必須額外加強 CVM Protocol 功能，其他皆引用現有 Video Mixer、Video Codec 與 RTP 功能。

#### 4.3 SIP UA

SIP 制定初期未考慮提供會議服務，一般 SIP UA 無法攜帶會議成員資訊，即使會議伺服器可以提供多人會議功能，但由於無法從會議發起者的請求訊息中得到會議成員資訊，會議伺服器仍無法自動邀請會議參與者，達到隨意會議服務。因此，我們必須在加強一般 SIP UA 功能，加入處理 MIME 多重本體與 URI-list 之能力。本論文以 Open H323 該組織研究 OPAL (Open Phone Abstraction Library) 專案下提供的 SimpleOpal[12] 為 User Agent 開發基礎[2]。

### 五、 案例說明與功能驗證

#### 5.1 會議建立

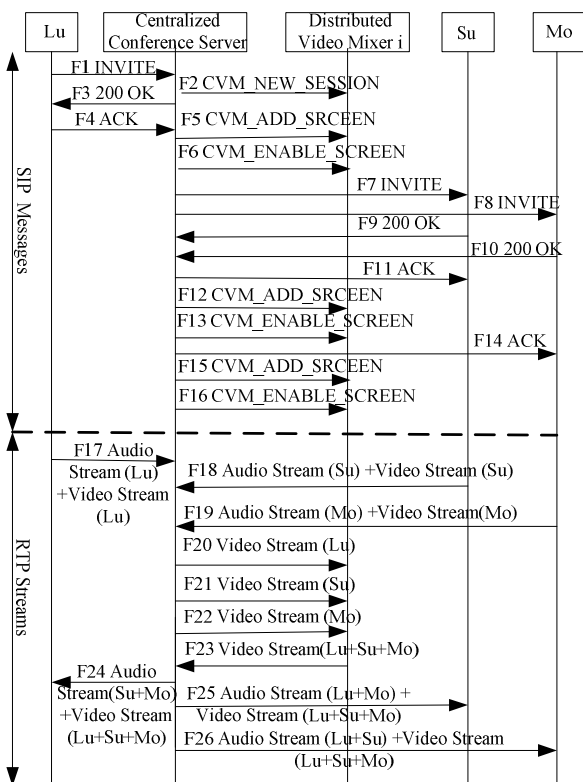
在此我們以一案例說明多媒體會議建立流程，Lu 為會議的發起人，他要邀請 Su 與 Mo 一起開會。如圖七，會議詳細建立流程說明如下：

1. 一開始 Centralized Conference Server 等待 Client 端的建立會議請求，而 F1 INVITE 則是由 Lu 送出給 Centralized Conference Server 要求建立會議的需求，其中 URI-list 明確標示欲邀請 Su 與 Mo 建立多媒體會議。
2. 當 Centralized Conference Server 接受到 F1 訊息後，Centralized Conference Server 根據 INVITE 的 URI-list 得知新會議的成員與成員個數，以 Distributed Video Mixer 資源分配機制，透過數學式 (2) 決定出負載最輕且滿足數學式 (1) 的 Distributed Video Mixer  $i$ 。最後，傳送 F2 CVM\_NEW\_SESSION 給選定的 Distributed Video Mixer  $i$ ，準備初始化混合視訊的工作。
3. 接著，Centralized Conference Server 回傳 F3 200 OK 給 Lu，告知 Lu 已成功建立會議，Lu 再傳回 F4 ACK 訊息確認。當會議發起者完成會議初始化，Centralized Conference Server 會傳 F5 CVM\_ADD\_SOURCE 與 F6 CVM\_ENABLE\_SOURCE 這兩個訊息給 Distributed Video Mixer  $i$ ，並且將 Lu 的影像串流轉向給 Distributed Video Mixer  $i$  處理，以及將 Distributed Video Mixer  $i$  回傳已完成混視訊的串流轉向給 Lu。
4. 當 Centralized Conference Server 收到具有 URI-list 的 INVITE 後，會根據 URI-list 上的成員 URI 逐一邀請到會議中。如範例中 Su 與 Mo 分別為 URI-list 上的成員，所以 Centralized Conference Server 分別傳遞 F7 和 F8 兩個 INVITE 訊息給成員，當成員接聽後再分別傳回 F9 和 F10 200 OK 訊息給 Centralized Conference Server，而 Centralized Conference Server 傳回 ACK 確認訊息給參與者後，會再傳送 F13 和 F15 的 CVM\_ADD\_SOURCE 與 F14 和 F16 的 CVM\_ENABLE\_SOURCE 訊息給 Distributed Video Mixer  $i$ ，並將 Su 與 Mo 的影像串



流轉向給 Distributed Video Mixer i 處理，以及將 Distributed Video Mixer i 回傳已完成混視訊的串流轉向給 Su 和 Mo。

- 完成會議建立後，參與此會議的成員會開始傳送影像與語音串流，在此為了簡化複雜度，我們將以箭頭方向表示資料串流流向，並以文字說明裡面包含哪些串流。Lu 會將自己的聲音與影像透過 F17 Audio Stream + Video Stream 傳送給 Centralized Conference Server；其他成員亦同，如：F18 與 F19。Centralized Conference Server 再將參與者的 Video Stream 傳給 Distributed Video Mixer i 處理，如：F20、F21 與 F22。接著，Distributed Video Mixer i 將完成視訊混合，含有三人混合視訊的影像的串流 F23 回傳給 Centralized Conference Server。最後，再由 Centralized Conference Server 將完成視訊混合與聲音混合之所有串流，導向給所有會議參與者，讓會議參與者可以聽到其他參與者的聲音，以及看得到其他人的視訊影像。



圖七：多媒體會議建立控制訊息時序圖

## 5.2 Distributed Video Mixer 資源分配計算

此小節，以一案例說明 Distributed Video Mixer 資源如何動態分配。假設一會議發起者，邀請三位參與者，欲建立一個成員數為四人的新會議 ( $n' = 4$ )，當前系統狀態與已知參數如表 II。

目前系統擁有兩台 Distributed Video Mixer，編號 1 與 2；兩台 Distributed Video Mixer 處理能力分別為，最多處理 12 個 Video Stream 與 18 個 Video Stream；目前有

三個多媒體會議正在進行，會議編號為 1、2 與 3；三個會議的成員個數分別為 3 人、4 人與 5 人。目前 Conference 1 的視訊資料是由 Distributed Video Mixer 1 處理；Conference 2 與 Conference 3 的視訊資料是由 Distributed Video Mixer 2 處理。

首先，我們判斷數學式 (1) 是否成立，在此範例中，Distributed Video Mixer 1 與 Distributed Video Mixer 2 皆滿足數學式 (1) 條件。在滿足數學式 (1) 的條件下，透過數學式 (2)，挑選出負載最小的 Distributed Video Mixer 1 來負責新會議的視訊混合作。 (Distributed Video Mixer 1 負載為 0.25，Distributed Video Mixer 2 負載為 0.5。)

表 II  
參數範例數值

參數	範例數值
Video Mixer 集合	$M = \{ 1, 2 \}$
各 Video Mixer 負荷量	$(c_1, c_2) = (12, 18)$
Conferences 集合	$R = \{ 1, 2, 3 \}$
各 Conference 成員數	$(n_1, n_2, n_3) = (3, 4, 5)$
Conference 與 Video Mixer 之間的對應關係矩陣	$x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$

## 六、結論

本論文提出階層式多媒體會議系統，將 Video Mixer 模組化，以一台 Centralized Conference Server 管理多台 Distributed Video Mixer 資源。透過隨意會議之實現，驗證系統 Video Mixer 資源動態分配之功能。最後，以 IXP 465 嵌入式系統實現 Centralized Conference Server 與部分 Distributed Video Mixer，在有限硬體資源下，驗證系統效能與系統產品化之可行性。本系統現階段已完功能性測試，在區域網路內初步測試平均會議建立時間約 0.8 秒，未來將針對整體效能進行更深入探討與分析，相信對未來網路多媒體會技術與產業發展會有實質幫助。

## 參考文獻

- Jackson, B, "History of VoIP," University of Texas at Dallas, Nov 2002.
- 張昆旭, 蔡昱航, 陳威志, 蘇暉凱, "以 SIP 為基礎多媒體會議系統之設計與實作," 2007年全國電信研討會 (NST 2007), Nov 23-24, 2007.
- Levin, O. and Even, R. "High-Level Requirements for Tightly Coupled SIP Conferencing," RFC 4245, Nov 2005.
- Rosenberg, J. "A Framework for Conferencing with the Session Initiation Protocol (SIP)," RFC 4353, Feb 2006.
- Johnston, A. and Levin, O. "Session Initiation Protocol (SIP) Call Control-Conferencing for User Agents," RFC 4579, Aug 2006.
- Koskelainen, P. and Ott, J. and Schulzrinne, H. and Wu, X. "Requirements for Floor Control Protocols," RFC 4376, Feb 2006.
- Camarillo, G. and Ott, J. and Drage, K. "The Binary Floor Control Protocol (BFCP)," RFC 4582, Nov 2006.
- Camarillo, G. "Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams," RFC 4583, Nov 2006.
- Intel IXP465 官方網站, <http://www.intel.com/design/network/products/npfamily/ixp465.htm>.
- Asterisk 官方網站, <http://www.asterisk.org>.
- CONFIANCE 官方網站, <http://confiance.sourceforge.net>.
- SimpleOPAL 官方網站, <http://www.voxgratia.org>.