

國立交通大學

多媒體工程研究所

碩士論文



雲端服務視訊互助平台之互動影音服務之研究

The Study of Interactive Video Service for Video Based Cloud

Remote Service Platform

研究生：林金昇

指導教授：傅心家 教授

中華民國一百年八月

雲端服務視訊互助平台之互動影音服務之研究

The Study of Interactive Video Service for Video Based Cloud

Remote Service Platform

研究生：林金昇
指導教授：傅心家

Student：Jin-Sheng Lin
Advisor：Prof. Hsin-Chia Fu

國立交通大學

多媒體工程研究所

碩士論文



College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

August 2011

Hsinchu, Taiwan, Republic of China

中華民國 一 百 年 八 月

雲端服務視訊互助平台之互動影音服務之研究

研究生：林金昇

指導教授：傅心家 教授

國立交通大學多媒體工程研究所

摘要

雲端互助視訊服務平台是綜合應用影音串流與多媒體網路通訊等技術，讓遠距各地的人士無須長途旅行，能如同面對面的請教及回答疑難問題。在困難問題解決後，再將協助解決問題過程的影音資料利用在伺服器叢集（Server Cluster）來編輯及儲存，供更多的網友日後搜尋及參考。本平台擬建構三個伺服器分別處理(1)使用者介面和用戶資訊、(2)影音發佈和接收給其他使用者、(3)客戶端間的訊息傳遞。

本論文著重在探討客戶端影音資訊如何產生和在不同使用者的電腦之間傳遞，設計一套連線結構能與平台的三伺服器溝通合作，與其他客戶端連接進行視訊互助活動。為了進一步方便及輔助使用者進入本平台進行互助視訊連線，以發展 Browser/Server 架構下的客戶端程式為主，建立影音伺服器與客戶端之間、客戶端與客戶端之間的通訊機制和影音串流的在客戶端與伺服器之間的傳輸結構。

另外，針對在不同的網路頻寬下，在客戶端設計一套演算法能夠偵測緩衝區使用量了解目前網路的擁塞情況，以調整發佈的串流資料量，使得客戶端能減少因頻寬不足而發生視訊不順暢、影音不同步等問題。

使用畫筆標誌功能，輔助互助視訊活動進行。活動結束，客戶端所有的影音串流、畫筆標誌、文字對話、關鍵圖片皆會留存下來，並且能在平台觀看這些記錄，這些合作經驗能給下一位有需要的人。

最後，設計多個測試去支持這些設計方法。

The Study of Interactive Video Service for Video Based Cloud Remote Service Platform

Student: Jin-Shang Lin

Advisor: Prof. Hsin-Chia Fu

Institute of Multimedia Engineering
Nation Chiao Tung University

Abstract

Video Based Cloud Remote Service Platform Integrated application streaming and multimedia networking of communications technology, people can ask and answer difficult problems as face to face on the platform. After solving difficult problems, the platform can collect problem-solving process and use a server cluster to edit and save for the future, then users search and reference. The platform construct three servers to process (1) user interface and user information, (2) publish and receive audio and video to other users, (3) multi client message transform.

The paper focus on the client-side audio and video information ,how to generate and transfer between different user's computer. Design a connection structure with three-server, connect other clients to interactive video activities. For Convenience and assist users to enter the video platform, the paper focus on development client application based on the Browser / Server , establish communication mechanism between Media Server and client or between client and client, transfer architecture of Video-Audio Streaming .between Media Server and client .

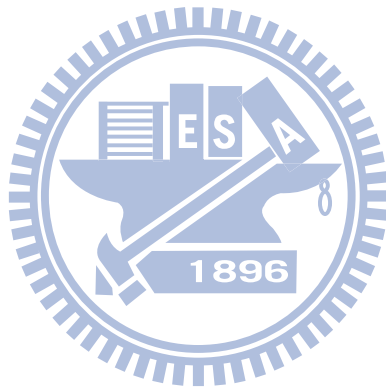
In addition, targeted at different network bandwidth, design a “Streaming Data Control Algorithm” can detect buffer utilization on the client, to understand the current network bandwidth. ,and control publishing streaming data size. Although, client can reduce problem such that video not occur smoothly, audio and video sync problem by lack of bandwidth.

Use the “Brush Mark System”, assisted working with interactive video activities. End of the activities, all the streaming client, brush marks, chat dialogue, key picture will record in the data base of platform. After , users can view these records on the platform. These record can give the next person who needs.

Finally, the paper design many test to support it.

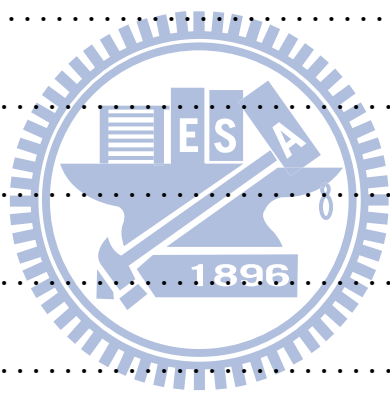
致謝

本論文得以順利完成，首先要感謝我的指導老師傅老師幫助我找到研究方向，並給予我鼓勵和支持，在此向老師致上最誠摯的敬意和感謝。同時，感謝博士班學長政龍在我遇到難題時給予協助，感謝一起進行雲端服務視訊互助平台研究的永彰和冠宏彼此激勵和指教，感謝碩士班學弟祐豪、醇豐、宜叡幫忙集思廣益。另外，也感謝家人給予支持和鼓勵，讓我能專心在學業上。



目錄

摘要	I
ABSTRACT	II
致謝	III
目錄	IV
表目錄	IX
圖目錄	X
第 1 章 前言	1
1.1 研究動機	1
1.2 研究目標	1
1.3 章節介紹	2
第 2 章 相關研究	3
2.1 影音串流	3
2.1.1 串流媒體(STREAMING MEDIA)	3
2.1.2 串流媒體伺服器 (STREAMING MEDIA SERVER)	3
2.1.3 RTMP 通訊協定 (REAL-TIME MESSAGING PROTOCOL)	3
2.2 視訊會議系統	4
2.2.1 ADOBE CONNECT	4
2.2.2 GOOGLE+ HANGOUTS	5
2.3 頻寬與影音之相關研究	6



2.3.1	ADAPTIVE BITRATE STREAMING.....	6
2.3.2	CAMERA CONTROL.....	6
第 3 章	研究方法與內容.....	7
3.1	雲端服務視訊互助平台架構.....	7
3.1.1	平台架構架構與各伺服器介紹.....	7
3.1.2	影音伺服器系統架構.....	8
3.1.3	使用者介面與設計.....	9
3.1.3.1	開始視訊連線活動.....	10
3.1.4	雙人影音連線架構設計.....	13
3.1.4.1	建立雙人連線的流程.....	13
3.1.4.2	單人連線.....	16
3.1.4.3	訪客功能.....	16
3.1.5	多人影音連線架構設計.....	16
3.1.5.1	連線架構.....	16
3.1.5.2	流量結構.....	17
3.1.5.3	加入訪客後的流量結構.....	18
3.2	影音串流設計.....	19
3.2.1	影音合併或分流設計.....	19
3.2.1.1	串流傳輸結構設計.....	20
3.2.1.2	二客戶端串流發送切換的設計.....	21
3.2.1.3	流量結構比較.....	24

3.2.1.4 儲存結構比較.....	25
3.2.2 串流資料丟失問題與緩衝區設計.....	25
3.2.2.1 緩衝區的概念.....	26
3.2.2.2 丟棄資料量計算.....	30
3.2.3 影音同步問題.....	33
3.2.3.1 發佈端影音同步問題.....	33
3.2.3.2 客戶端間影音同步問題.....	33
3.3 串流資料量調控演算法設計.....	34
3.3.1 設計概念.....	34
3.3.2 串流資料量調控演算法.....	35
3.3.2.1 輸入與輸出.....	35
3.3.2.2 演算法流程.....	37
3.3.3 設計實驗與討論.....	43
3.3.3.1 不同網路環境之演算法運行測試.....	43
3.4 畫筆標誌系統.....	45
3.4.1 設計方法.....	45
3.4.2 多客戶端間的畫筆同步.....	47
3.4.3 標誌自動清除.....	48
3.4.3.1 設計概念.....	48
3.4.3.2 清除門檻的制定.....	49
3.5 互助歷程之記錄與重現系統.....	50

3.5.1	記錄系統設計.....	50
3.5.2	重現系統設計.....	51
3.5.2.1	設計概念與結構.....	51
3.5.2.2	介面介紹.....	52
3.5.3	重現系統之同步問題.....	53
3.5.3.1	系統標準時間軸.....	53
3.5.3.2	多串流播放.....	54
第4章	系統測試與評估.....	55
4.1	影音伺服器負載測試.....	55
4.2	多人連線結構之效能測試.....	59
4.2.1	客戶端間訊息傳遞與接收串流反應時間測試.....	59
4.2.2	多人連線發佈與接收串流之效能測試.....	62
4.3	多人影音同步問題.....	63
4.3.1	影音分流或合送之影格丟失測試.....	63
4.3.2	不同頻寬限制下的串流合併或分送資料丟棄率測試.....	64
4.3.3	畫筆標誌透過訊息傳遞伺服器傳送的基本完成時間測試.....	67
4.4	記錄與重現系統測試.....	68
4.4.1	多人連線記錄之同步播放測試.....	68
第5章	結論與建議.....	70
5.1	結論.....	70
5.2	未來展望.....	70

5.2.1	公式化的進階串流資料量調控演算法.....	70
5.2.2	視訊連線客戶端應用於行動裝置平台.....	70
5.2.3	標誌智慧型追蹤系統.....	71
5.2.4	記錄重現系統進階功能.....	71
第 6 章	參考文獻.....	72
附錄一	串流資料量調控演算法完整流程圖.....	73



表目錄

表 3-1 不同使用者身分對應的客戶單程式功能差異表	13
表 3-2 發佈串流有緩衝區對各種客戶端的影響	28
表 3-3 發佈端緩衝區滿時會出現的現象：	28
表 3-4 演算法運作所需的輸入參數	36
表 3-5 串流資料量調控演算法主要調控的參數項目	36
表 3-6 演算法使用的主要變數	38
表 3-7 調升調降函數用的變數	40
表 3-8 各種畫筆標誌類型的標誌訊息結構	48
表 4-1 平台中的網站、訊息、影音伺服器硬體規格	55
表 4-2 測試用客戶端的硬體規格	55
表 4-3 客戶單間標誌訊息送出到另一客戶端標誌完成的基本完成時間	68



圖目錄

圖 3-1 互助平台的整體架構	7
圖 3-2 客戶端 FLASH 程式—三伺服器運作狀態檢查介面	10
圖 3-3 視訊與麥克風裝置設定介面	11
圖 3-4 成員發佈端的主畫面	11
圖 3-5 成員接收端的主畫面	12
圖 3-6 訪客接收端的主畫面	12
圖 3-7 雙人連線流程圖 1.....	13
圖 3-8 雙人連線流程圖 2.....	14
圖 3-9 雙人連線流程圖 3.....	14
圖 3-10 雙人視訊連線進行時的系統架構圖.....	15
圖 3-11 多人連線系統架構圖.....	17
圖 3-12 左：輸入至影音伺服器流量示意圖；右：從影音伺服器輸出流量示意圖.....	18
圖 3-13 串流合送架構圖.....	20
圖 3-14 串流分送架構圖.....	20
圖 3-15 客戶端提出切換請求的示意圖.....	21
圖 3-16 合併串流切換步驟圖 1	22
圖 3-17 合併串流切換步驟圖 2	22
圖 3-18 分流串流切換步驟圖 1	23
圖 3-19 分流串流切換步驟圖 2	23
圖 3-20 測試 4.3.1 的結果：視訊流合併與分流之丟棄影格數.....	26
圖 3-21 測試 4.3.2 的結果：視訊流合併和分流之資料丟棄率比較圖	31
圖 3-22 測試 4.3.2 的結果：音訊流合併和分流之資料丟棄率比較圖	31
圖 3-23 串流資料量調控演算法示意流程圖.....	37
圖 3-24 調升函數運作流程圖.....	41

圖 3-25 調降函數運作流程圖	42
圖 3-26 演算法有無之視訊流資料丟棄率比較圖	43
圖 3-27 演算法有無之音訊流資料丟棄率比較圖	44
圖 3-28 演算法有無之視訊流送出資料量比較圖	44
圖 3-29 畫筆標誌系統.....	46
圖 3-30 使用者操作畫筆的介面	46
圖 3-31 標誌訊息傳送的架構圖.....	47
圖 3-32 與各伺服器溝通取得記錄的結構圖.....	51
圖 3-33 重現系統播放器介面	52
圖 3-34 播放畫筆記錄與文字對話的示意圖	53
圖 3-35 播放各記錄時間示意圖圖.....	53
圖 4-1 影音伺服器負載測試之伺服器輸出流量圖	57
圖 4-2 伺服器平均 CPU 使用率圖.....	57
圖 4-3 伺服器平均記憶體使用率圖，總量為 3.7GB.....	58
圖 4-4 伺服器處理客戶端接收串流之反應時間	58
圖 4-5 客戶端 FLASH 程式要求其他客戶端接收影音串流的通知路徑	59
圖 4-6 使用實機之測試結果	61
圖 4-7 改以軟體模擬使用者之測試結果	61
圖 4-8 發佈開始到所有接收端完成接收之接收端最大延遲時間差圖	62
圖 4-9 測試 4.3.1 的結果：視訊流合送與分流之丟棄影格數.....	64
圖 4-10 測試 4.3.2 結果：視訊流合併和分流之資料丟棄率比較圖	66
圖 4-11 測試 4.3.2 結果：音訊流合併和分流之資料丟棄率比較圖	66
圖 4-12 測試 4.3.2 結果：串流合併和分流之資料發送量比較圖	66
圖 4-13 多人串流同時播放延遲時間趨勢圖.....	69

第1章 前言

1.1 研究動機

這幾年網路多媒體技術在畫面品質及傳輸速度都有卓越的進步，透過網路在遠端可以如身歷其境的看/聽到現場狀況。如今需要的是整合多媒體資訊與網路技術，來建構視訊互助平台，讓各地需要幫助的人們僅須用筆電或手機就可連到平台上，請求網友們幫助解決困難問題。利用網路突破了空間的限制，許多以前需要在特定的時間、地點才能進行的會議、教學以及各種的服務工作，都可以在平台上隨時進行。加上目前日益成熟的雲端技術，藉由雲端伺服機制來儲存及處理前述遠端視訊連線所產生的大量影音資料，建構可隨時查詢的影音資料庫，讓平台的遠距服務功能更趨完備。

在本平台的設計理念之下，本論文將著重在協助使用者進行視訊連線活動。設計一個客戶端與伺服器、客戶端與客戶端通訊機制進行多人視訊連線活動，解決使用者間互動與溝通會產生的問題。



1.2 研究目標

為了讓使用者隨時隨地更容易地進行視訊連線活動，本論文著重在雲端互助服務平台中的應用影音串流與多媒體網路通訊等技術的客戶端程式開發。

主要的研究目標分為三個方向切入：

(1) User Interactive 使用者互動——協助使用者之間溝通描述問題和現場狀況

現行的畫面標誌技術大多是在靜態圖像進行，對視訊的動態畫面描述不足夠。本論文設計的畫筆標誌系統是一個嶄新概念，將在動態視訊畫面直接進行標誌，在多人視訊連線活動時，讓使用者更進一步的描述畫面內容。

(2) Auto Control 自動控制——輔助使用者使用其網路攝影機和可用的頻寬

本論文將探討不同的串流結構設計、有限頻寬下影音資料丟棄情形和設計控制影音

資料產生量的方法，讓使用者無需繁雜操作，亦能有效利用頻寬並且改善因影像、聲音資料產生量不同造成的影音同步問題。

(3) Record 記錄——完整記錄互助過程，供其他使用者日後有效利用

一個多人視訊連線活動將會有多位使用者進入相互溝通，溝通過程將包含影音、文字對話、畫面標誌等資料，且不同使用者有其各自的溝通資訊，本論文設計記錄重現系統將記錄和播放所有使用者的溝通資訊，重現複雜的多人視訊連線過程。

1.3 章節介紹

第二章蒐集了關於網路影音服務的相關資料，如網路視訊會議、串流媒體的通訊協定和視訊資料產生的參數控制等。第三章介紹本論文設計之本平台中與影音服務相關的重點技術，如影音伺服器 and 影音服務在本平台的地位和功能、客戶端與平台個伺服器端的連線架構、客戶端程式的互動功能、改善影音串流和同步問題的核心演算法等。第四章測試本平台的伺服器負載和與第三章內容相關的驗證性測試。第五章結論和未來能進一步改進的方向。

第2章 相關研究

2.1 影音串流

2.1.1 串流媒體 (Streaming media)

串流媒體 (Streaming media) [1], 簡稱串流 (Streaming) 是指將一連串的媒體資料壓縮後, 經過網路分段傳送資料, 在網路上即時傳輸影音以供觀賞的一種技術與過程, 此技術使得資料封包得以像流水一樣發送, 讓客戶端與伺服器保持連線

(connection); 如果不使用此技術, 就必須在使用前下載媒體檔案, 這種方式又稱為漸進式 (Progress)。串流傳輸可傳送現場影音或預存於伺服器上的影片, 當觀看者在收看這些影音檔時, 影音資料在送達觀賞者的電腦後立即由特定播放軟體播放 (如 Windows Media Player、Real Player、Adobe Media Player 或 QuickTime Player)。

2.1.2 串流媒體伺服器 (Streaming Media Server)

本研究用的影音伺服器是一種串流媒體伺服器 (Streaming Media Server), 為串流媒體技術應用的核心系統, 向使用者提供視訊服務的關鍵平台。其主要功能是對串流媒體進行收集、緩存、調度和傳輸播放, 串流媒體應用系統的主要性能表現都取決於媒體伺服器的能力和程式結構。性能好的媒體伺服器能夠提供更多的客戶端連接數、更高的資料速率和更快的客戶端連接時間。目前市面上較知名的串流媒體伺服器有 Adobe Media Server、Red5 Media Server、Wowza Media Server 等。

2.1.3 RTMP 通訊協定 (Real-Time Messaging Protocol)

RTMP (Real-Time Messaging Protocol) [2] 協定是做為客戶端與伺服器端的即時傳輸協定, 是一種專門為高效率傳輸視訊、音訊和訊息資料且基於 TCP/IP 的通訊協定。以 RTMP 傳遞的串流可以包含視訊流 (Video)、音訊流 (Audio)、訊息資料流 (Message

Data)，RTMP 將這些資料分開管理，可視為有三條資料流在一個串流中流動。

綜合 2.1 小節，串流媒體為本論文研究客戶端與影音伺服器間傳輸的主要媒介，本平台的影音伺服器即是一種串流媒體伺服器，主要使用的通訊協定為 RTMP。

2.2 視訊會議系統

視訊會議系統是一種在位於兩個或多個地點的多個使用者之間提供語音和運動彩色畫面的雙向即時傳送的視聽會話。視訊會議著重在確保使用者間的畫面和語音清晰順暢，會議進行時大多有一位主持人主持會議進行，由主持人決定誰可發言、分享檔案或視訊畫面等。大型視訊會議系統在軍事、政府、商貿、醫療等部門有廣泛的應用。視訊會議依建置方式分成硬體、軟體、網路視訊會議 [3]。

現行多數的視訊會議軟體是 C/S 架構，即桌面用戶端(Client)軟體與伺服器(Server)端軟體。正常執行前，需要先配置網路和伺服器，下載並安裝客戶端程式；另外一種軟體是 Browser/Server (B/S)¹ 架構，由 C/S 發展而來，好處是使用者可直接在網頁瀏覽器上進行視訊會議，方便好上手是最大的優點，但能力也會受限於 Browser。

視訊會議著重在確保使用者間的畫面和語音清晰順暢，會議進行時大多有一位主持人主持會議進行，由主持人決定誰可發言、分享檔案或視訊畫面等。

以下是幾個新型態的線上視訊通話系統：

2.2.1 Adobe connect

Adobe Connect 為一個完全基於 Adobe Flash 開發的視訊傳送、桌面共享、檔案共

¹在本雲端服務視訊互助平台中，Browser/Server 架構的 Browser 相當於端、Server 相當於雲，但廣義來說 Browser/Server 架構也是一種 Client/Server 架構，只是為了強調本平台使用的客戶端程式是應用在 Browser 上而特別提出來。

享的網路視訊會議系統。僅需在瀏覽器安裝 Adobe Flash Player²即可直接在瀏覽器上進行視訊會議，除了傳統視訊會議故有的白板、檔案、桌面共享功能外，應用了 Adobe 發展純熟的 RTMP 通訊協定、串流媒體伺服器和串流媒體技術，成為一個 B/S 架構中能媲美 C/S 架構的視訊會議系統。

2.2.2 Google+ Hangouts

由 Google 開發的社群網站「Google+」，其中包含一個線上多人影音通話系統「Hangouts」，不具備白板、檔案、桌面共享等功能，透過 Google+ 的社群功能，可在自己的交友圈找人加入影音通話中。Google Hangouts 與一般視訊會議取向不同，其重點不是會議，而是在提供一個便利的網路互動平台，便利的多人語音和視訊，故一直致力在免安裝上努力以減少使用者的負擔。Google 在推廣開放原始碼聊天平台 WebRTC³ 加入瀏覽器中，目前已經有 Mozilla 和 Opera 支援 WebRTC，如此一來使用這兩種瀏覽器將不需要額外安裝外掛程式。

綜合 2.2 小節，本平台的視訊連線系統核心概念並非是視訊會議，而是 Adobe connect 和 Hangouts 概念的整合。有使用現在網路主流的串流媒體技術作為主要傳輸媒介，Google Hangouts 的社群與視訊通話相輔相成，結合文字交流的社群討論區和可以即時互動的視訊連線，進一步達到互助合作的目的。另外本平台將會使用 Adobe Flash

² Adobe Flash Player 是一種廣泛使用的、專有的多媒體程式播放器。它最初由 Macromedia 編寫，在 Macromedia 被 Adobe 收購後由 Adobe 繼續開發並分發。Flash Player 使用的 SWF 檔案可能由 Adobe Flash、Adobe Flex 或者其他軟體或第三方工具建立。2011 年已廣泛的被建立在各種瀏覽器中，故通常使用者不需要特別去安裝，但是需要使用更新檢查程式定期更新版本。

³ WebRTC 是 Google 推出的開放原始碼(BSD)語音視訊聊天軟體。透過簡單的 JavaScript API 使 Chrome 擁有即時通訊能力(RTC)，並與 Mozilla 和 Opera 等瀏覽器廠商合作，推動 WebRTC 能成為互動網路視訊和點對點通信的標準，並在此基礎之上制定網路通訊協定。

作為開發平台，其主要原因就是因為 Adobe Flash Player 已廣泛的內建在瀏覽器中，降低使用者進入本平台的門檻。

2.3 頻寬與影音之相關研究

2.3.1 Adaptive bitrate streaming

Adaptive bitrate streaming 是一種應用於串流媒體的技術，它的運作原理在於檢測客戶端的頻寬和 CPU 能力後，在伺服器端即時調整對應 bit rate 的影音串流資料傳送給該客戶端，好處是可以客制化，減少客戶端的緩衝和擁有流暢的畫面。缺點是對伺服器能力要求高，使用的編碼器要可以將單一來源的視訊資料編碼成多種 bit rates，依客戶端頻寬不同而切換不同的 bit rates。若是即時 (live) 視訊連線，則伺服器會被要求在更短的時間完成。

一些商用的串流媒體伺服器已經有提供此技術支援，如 Adobe Flash Media Server。

2.3.2 Camera control

與 Adaptive bitrate streaming 不同，Camera control 旨在控制攝影機的參數以改變產生的資料量，屬於一種客戶端控制硬體的技术，大多控制攝影機每秒抓取的張數 (FPS)、畫面解析度 (Resolution)，其調整參數的反應速度會受到攝影機的硬體能力影響。

綜合 2.3 小節，Adaptive bitrate streaming 技術已有許多商用伺服器發展成熟。本論文著重在客戶端的程式開發，對頻寬不足的處理將朝 Camera control 方向去研究。

第3章 研究方法與內容

3.1 雲端服務視訊互助平台架構

3.1.1 平台架構架構與各伺服器介紹

本平台建立伺服器叢集(Server cluster)負責使用者互助連線之間的資訊傳輸、記錄儲存,依其功能和需求我們建立三個伺服器。伺服器叢集包含網頁伺服器(Web Server, WBS)、訊息傳遞伺服器(Message Server, MSS)、及影音伺服器(Media Server, MDS)三台伺服器。互助平台的整體架構如圖 3-1 所示。

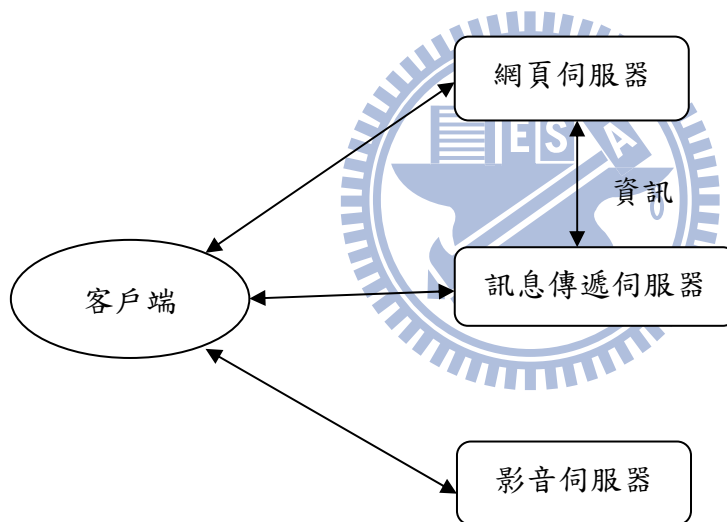


圖 3-1 互助平台的整體架構

各伺服器在平台中的功能為：

- (1) 網頁伺服器：提供客戶端的使用者透過瀏覽器以瀏覽網頁方式來連接互助平台。伺服器上有儲存平台所有資料（包括影音、文字、對話及使用者註冊等資訊）的中央資料庫及管理系統，並提供討論區介面讓使用者可在平台中找到互助合作的夥伴，建立起自己的「交友圈」，找到興趣相近或是能夠互相幫忙解決問題的同好。
- (2) 訊息傳遞伺服器：主要負責活動進行時的傳遞使用者之間的互動訊息。整個互助平台中負責扮演訊息傳遞中心的角色，透過網路接受來自客戶端、網頁伺服器的訊息轉

送到適當的目的地，並且在伺服器端記錄其文字溝通、影音標註歷程等資訊，以供後續觀看使用。

(3) 影音伺服器：負責活動進行時使用者之間的串流影音接收及傳送、儲存使用者產生的各類影音串流和提供活動結束後影音串流的記錄播放。

為了方便使用者進入本平台進行視訊互助合作。在此平台架構下，本論文專注在發展 Browser/Server (B/S) 架構下的客戶端程式，並發展影音伺服器與客戶端之間的通訊機制、客戶端與客戶端之間的同步通訊機制和影音串流的在客戶端與伺服器之間的傳輸結構。而與網站伺服器的相關研究請參考 [4]，與訊息傳遞伺服器相關研究請參考 [5]。

本平台進行視訊互助合作的流程為：安裝和設定所需的影音裝置(如 Webcam 及麥克風)後，接著使用瀏覽器進入本平台網站，操作網頁建立一個視訊連線的房間，在本論文稱此為一個「活動」。使用者建立活動，並邀請合作夥伴加入此活動。在這個活動中，使用者可以使用客戶端程式提供的功能如文字對話、語音通話、視訊傳送與切換、畫面標誌等，系統會將這些互助過程皆記錄下來，成為有意義的資訊可供後人再利用，亦可從網站的資料庫中找尋相關的前人留存互助記錄訊息和影片。

3.1.2 影音伺服器系統架構

使用者在使用此平台時，使用本研究開發的客戶端程式進行連線。為了能夠看到對方的影像，我們必須建立一個伺服器專門處理客戶端發佈到伺服器的影音串流和將此影音串流轉送給其他客戶端觀看。同時，因為影音串流必定會經過伺服器，所以使用者在做影音交流時的所有過程皆可以存檔記錄在伺服器上，本論文稱這個專門處理影音轉送和存檔的伺服器為「**影音伺服器 (Media Server)**」，其中發佈影音串流的客戶端稱為「**發佈端 (Publisher)**」，接收影音串流的客戶端稱為「**接收端 (Receiver)**」。一個連線活動同時間只會有一個影像串流發佈到影音伺服器，其他多個客戶端連上來皆為接收端且

能同時觀看這個影音串流。

影音伺服器會開啟數個固定的通道 (port) 作為客戶端連線的入口，客戶端和影音伺服器之間使用 RTMP (Real Time Message Protocol) 作為通訊協定，由客戶端主動發起連線請求將影音串流發佈到伺服器端，同時在伺服器存檔寫到硬碟，並會由網頁伺服器和資料庫派發的一個專屬影片碼 (Video Code) 作為寫入檔名。另一客戶端從網頁伺服器的資料庫端取得這個影片碼，就可以在伺服器上找到目標影音檔。

影音伺服器的主要功能有：

- (1) 串流轉送：將發佈端傳來的串流轉給所有訂閱此串流的接收端。
- (2) 串流存檔：接收到發佈端傳來的串流同時寫入伺服器的硬碟。
- (3) 即時連線模式的串流狀態訊息傳送：發佈端發佈或停止發佈串流到伺服器時，伺服器會回傳狀態訊息到客戶端，訂閱的接收端能依據這些來自伺服器的狀態訊息作狀態轉換或是介面變化，如收到串流已停止發佈的訊息，接收端能在介面上通知使用者串流已被停止發佈。

3.1.3 使用者介面與設計

使用者使用本論文特別開發的 Adobe Flash 程式作為視訊連線活動的客戶端，選用 Flash 的原因主要有：

- (1) Flash 被廣泛應用在各大網站中，使用者對 Flash 也相當熟悉，不論是在網頁上玩遊戲或是炫麗的動畫效果，許多都來自 Flash。有些瀏覽器有內建 Adobe Flash Player 如 Google Chrome 5.0 以上，即使沒有內建，遇到需要載入 Flash 時也會通知使用者要安裝或更新。因此符合本平台希望能減少使用者進入門檻的目的。
- (2) Flash 對於網路影音傳輸格式——串流媒體 (Streaming media)、RTMP 通訊協定與串流媒體伺服器的連接和影音擷取裝置控制有完整的 API 支援。

3.1.3.1 開始視訊連線活動

在本平台視訊連線活動中，依有無發佈影像權力將連線身分分成兩種：

- (1) 成員(Member)：需要設定網路攝影機和麥克風的使用者，可以參與語音通話和取得影像發佈權，亦可在主畫面做標誌，讓所有房間內的使用者都看的到。另外，以 Client-Server 角度定義具有發佈權的成員稱為「**成員發佈端**」；不具發佈權的成員稱為「**成員接收端**」。
- (2) 訪客 (guest)：不具任何權限，僅能接收影像和聲音觀看。僅能加入成員建立的房間觀看。另外，以 Client-Server 角度定義訪客稱為「**訪客接收端**」

使用者進入本平台網站，進行視訊連線活動，首先會看到 flash 程式在瀏覽器中被啟動，接著檢查平台的三個伺服器運作情況如圖 3-2。

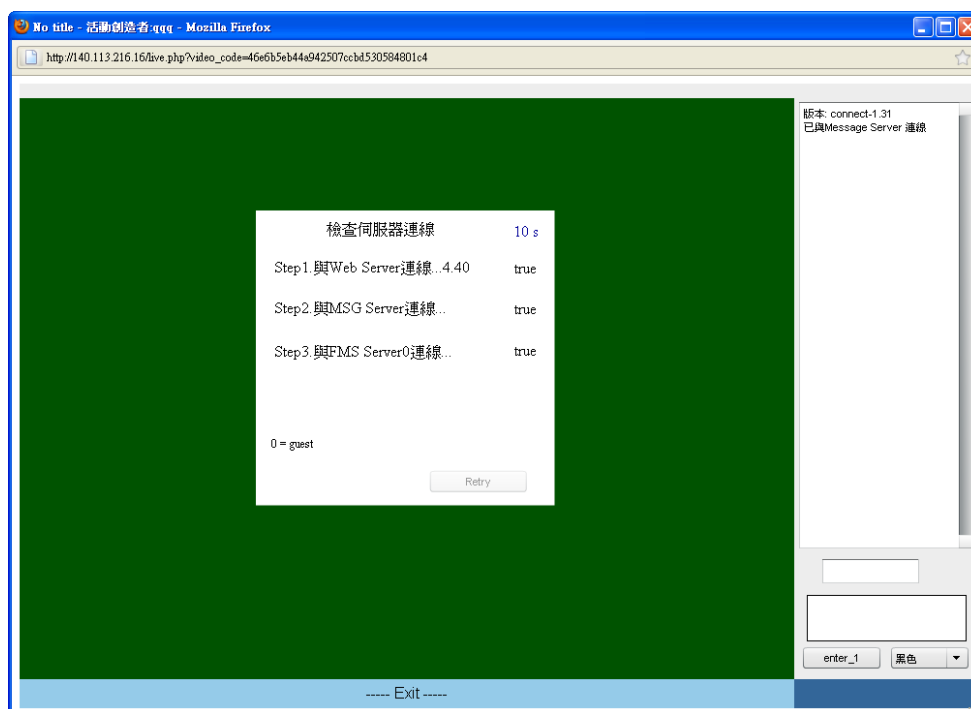


圖 3-2 客戶端 Flash 程式—三伺服器運作狀態檢查介面

當三伺服器皆正常運作，才會到下一個頁面，否則會被卡住在檢查頁面被要求重新進行連線。接著依照使用者的身分是成員或訪客而進入不同頁面，成員使用者會進入裝置設定介面如圖，可以設定使用的網路攝影機、麥克風等裝置，若是**成員發佈端**，可以選擇即將要發佈串流的攝影機（紅框位置），輸入裝置設定介面如圖 3-3，待設定完成後進入主畫面；訪客使用者則無此設定介面，而是直接進入主畫面。



圖 3-3 視訊與麥克風裝置設定介面



圖 3-4 成員發佈端的主畫面

(a) 視訊畫面區、(b) 資訊區、(c) 畫筆控制區、(d) 攝影機參數控制區、(e) 攝影機切換控制區

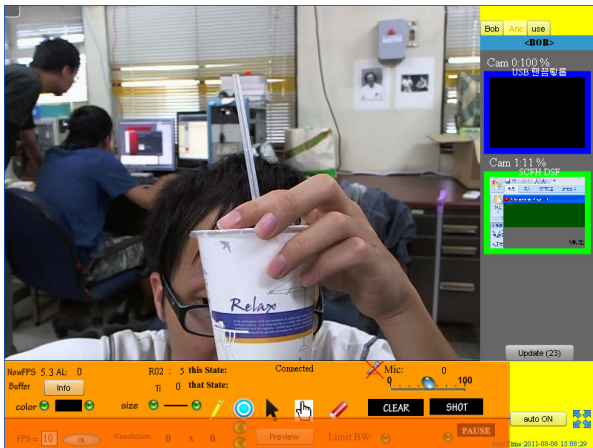


圖 3-5 成員接收端的主畫面

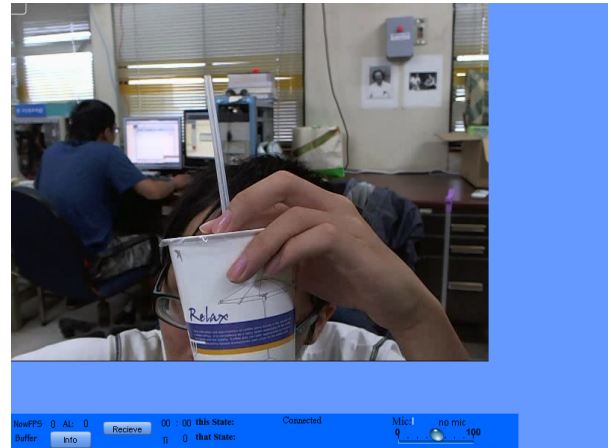


圖 3-6 訪客接收端的主畫面

不同身分的使用者看到的背景色會不同：成員發佈端為綠色（圖 3-4）、成員接收端為黃色（圖 3-5）、訪客接收端為藍色（圖 3-6），方便使用者區分現在的身分。

可以比較以上三圖發現不同身分的操作介面不盡相同：

- (1) 成員發佈端具有所有區域的功能
- (2) 成員接收端的攝影機參數控制區功能被鎖住，原因是接收端不具發佈權，不能發佈視訊影像。
- (3) 訪客接收端僅剩下視訊畫面區和資訊區，其餘區域皆無法使用。

各區域的主要功能詳細說明如下：

- (a) 視訊畫面區：所有活動內的客戶端 Flash 程式都會開到相同的視訊畫面，但是視訊來源不盡相同。發佈端的視訊畫面來自自己電腦的視訊擷取裝置，而成員接收端和訪客接收端皆來自影音伺服器的影音串流，這個影音串流是由發佈端負責發送產生的。
- (b) 資訊區：顯示目前活動內的狀態
- (c) 畫筆控制區：僅成員使用者（含發佈端和接收端）可以使用。讓使用者控制畫筆顏色粗細和切換標誌類型，在視訊畫面區進行畫筆標誌，所有活動內的成員和訪客都能看到。詳細介紹在 3.4 節。
- (d) 攝影機參數控制區：僅發佈端能使用，可以控制視訊擷取裝置（網路攝影機、虛擬

攝影機) 的設定參數，如畫面解析度、影格速率、資料速率，各參數意義會在 3.3 節詳細介紹

(e) 攝影機切換控制區：僅成員使用者可以使用，在本區域會顯示自己的視訊裝置畫面縮圖，也會顯示其他成員使用者的縮圖。各成員使用者可以依需求切換自己或是其他成員的視訊裝置，使其畫面顯示在視訊畫面區讓活動內所有成員都能觀看。詳細的切換流程會在 3.2 節介紹

表 3-1 不同使用者身分對應的客戶單程式功能差異表

	成員發佈端	成員接收端	訪客接收端
(a) 視訊畫面區	✓	✓	✓
(b) 資訊區	✓	✓	✓
(c) 畫筆控制區	✓	✓	
(d) 攝影機參數控制區	✓		
(e) 攝影機切換控制區	✓	✓	

3.1.4 雙人影音連線架構設計

3.1.4.1 建立雙人連線的流程

本小節以一個雙人連線來說明進行視訊連線活動的連線結構。

當在網站平台上找到可以互助合作的連線對象，使用者（以下稱 Client A）可對合作對象提出互助視訊連線，此即為兩個活動成員進行視訊連線活動。以下說明雙人連線的流程。

步驟一：

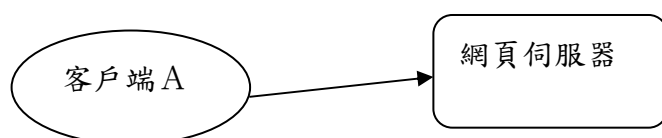


圖 3-7 雙人連線流程圖 1

客戶端 A 透過瀏覽器進入平台，在網站找到可以連線的對象。

步驟二：

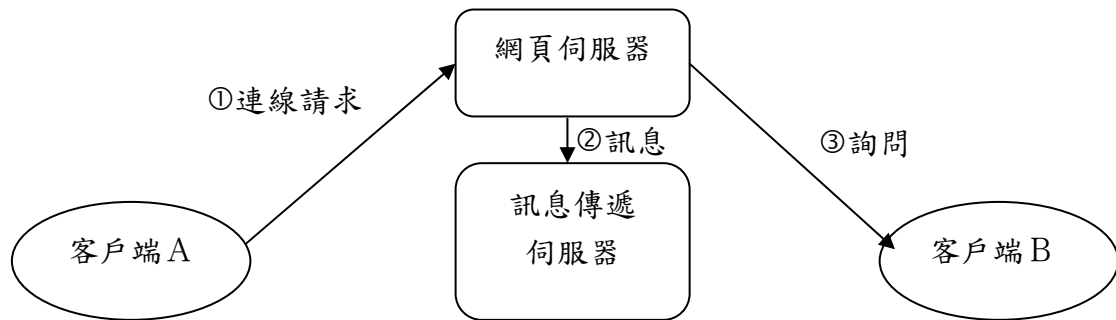


圖 3-8 雙人連線流程圖 2

①客戶端 A 透過發出連線請求。

②網頁伺服器收到指示，向訊息傳遞伺服器要求開新活動房間，若訊息傳遞伺服器(在可負載範圍內)同意，則建立房間等待連線

③向客戶端 B 詢問是否同意接受連線邀請

步驟三：

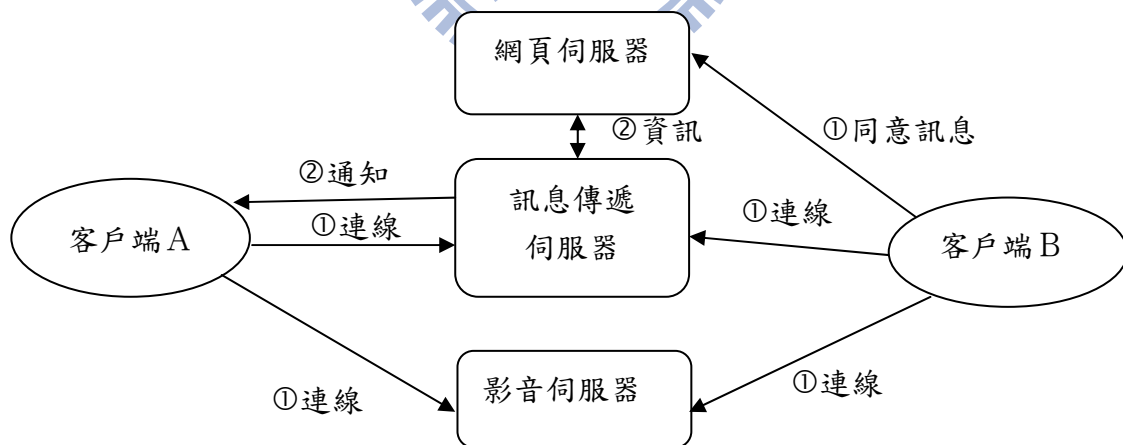


圖 3-9 雙人連線流程圖 3

①客戶端 A 連上訊息傳遞伺服器和影音伺服器，在已經建立的活動房間等待客戶端 B 進入。若客戶端 B 同意連線請求，也會連上訊息傳遞伺服器和影音伺服器到客戶端 A 所在的活動房間。

②網頁伺服器會將客戶端B同意連線的訊息通知訊息傳遞伺服器，由訊息傳遞伺服器告知客戶端A；客戶端B同意並且進入活動中。訊息伺服器

步驟四：

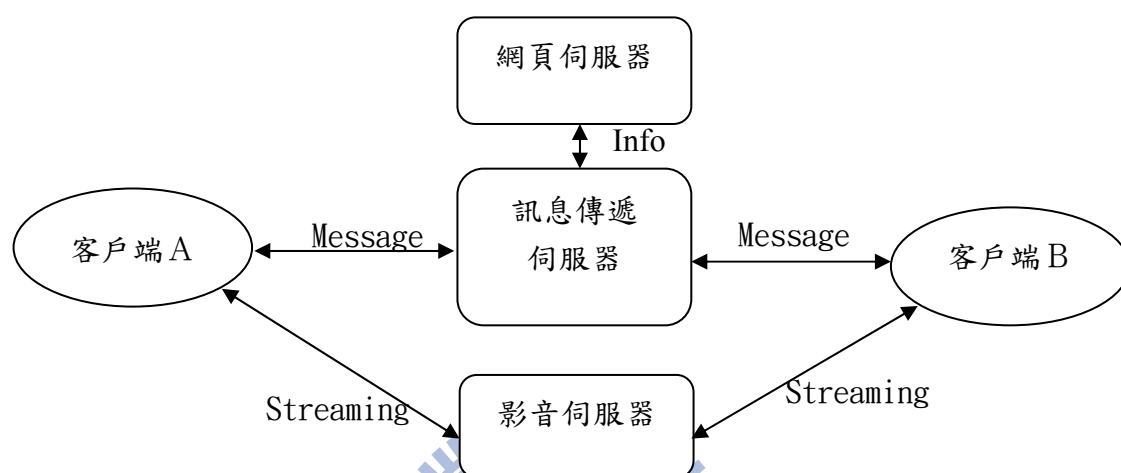


圖 3-10 雙人視訊連線進行時的系統架構圖

如圖 3-10，訊息傳遞伺服器負責雙方客戶端的文字訊息和動作指令的傳送；影音負責雙方客戶端影音串流的傳送溝通。

主要傳送的資料流：

- (1) Message：傳送文字對話訊息、畫筆標誌指令、介面操作指令、房間成員狀態
- (2) Streaming：客戶端A做為發佈端連接到影音伺服器發佈影音串流，同時發佈端也接收由客戶端B發佈的影音串流；相對的客戶端B同時接收來自客戶端A的影音串流，也發佈影音串流到影音伺服器再轉送給客戶端A。
- (3) Info；回報使用者進出狀況給網頁伺服器，同時也記載在資料庫中。

客戶端A作為發佈端的同時也會將視訊畫面也顯示在自己的視窗，客戶端B則是接收來自影音伺服器的視訊畫面，並且顯示在視窗中。雙方能在極小的時間差內觀看到相同的視訊畫面。

以上也說明了視訊連線活動時，各伺服器資料流溝通的流程。

3.1.4.2 單人連線

除了上述提到的雙人連線以外，我們也設計了單人和三人以上的多人連線。單人連線的系統設計是不需要客戶端 B 參與連線，在流程上少了通知客戶端 B 連線這個步驟，所以連線的部分只剩下客戶端 A 對網頁伺服器、訊息傳遞伺服器、影音伺服器三個伺服器的連線。影音串流同樣會傳送到影音伺服器上存檔錄影，但影音伺服器不需要將收到的串流再轉送出去。

3.1.4.3 訪客功能

在活動進行時，只要活動發起人將活動房間設定為公開，在平台上的其他使用者可以以訪客的身分進入此活動房間。訪客在連線架構中的地位，可視為圖 3-10 中將伺服器和客戶端 B 連線的雙向箭頭改為單向「伺服器→客戶端」，即為訪客接收端，僅能接收活動內的串流和訊息，無法取得發佈權、無法發出任何串流到影音伺服器，以避免訪客過多干擾到活動內的成員使用者，

3.1.5 多人影音連線架構設計

一個影音串流活動，發佈端只有一個，但是接收端可以有許多個，在 Media Server 負荷的範圍內，同時甚至可已超過百人同時觀看同一個影音串流，訪客客戶端僅能接收串流和訊息，不算是本平台中完整功能的客戶端。以下的「多人」皆泛指多位使用者受邀請加入活動的成員 (Member)，是具備有發佈影像和語音功能完整的客戶端。

本小節將探討多人視訊連線活動連線架構和其複雜度，並且討論對伺服器造成的頻寬消耗隨人數成長情況。

3.1.5.1 連線架構

建立連線的流程類似雙人連線，在雙人連線流程中客戶端 A 僅邀請一位成員客戶端 B 加入，將步驟二的圖 3-8 邀請一位改為同時邀請/詢問多位客戶端，步驟三多位被邀請的客戶端若同意，則會陸續加入活動中成為活動成員 (Member)。每一位成員客戶端加入，皆會連上訊息傳遞伺服器和影音伺服器，並由訊息傳遞伺服器通知「加入活動」訊息給已在活動內的所有客戶端。

假設目前有 N 個成員進入同一個影音活動房間，Client 1 為發佈端，剩下 $N-1$ 個使用者 Client 2 到 Client N 為接收端。Client 1 會透過 Web Server 一次發出 $N-1$ 個連線請求給這些使用者，當這些使用者同意進入房間後，其連線架構圖如圖 3-11。

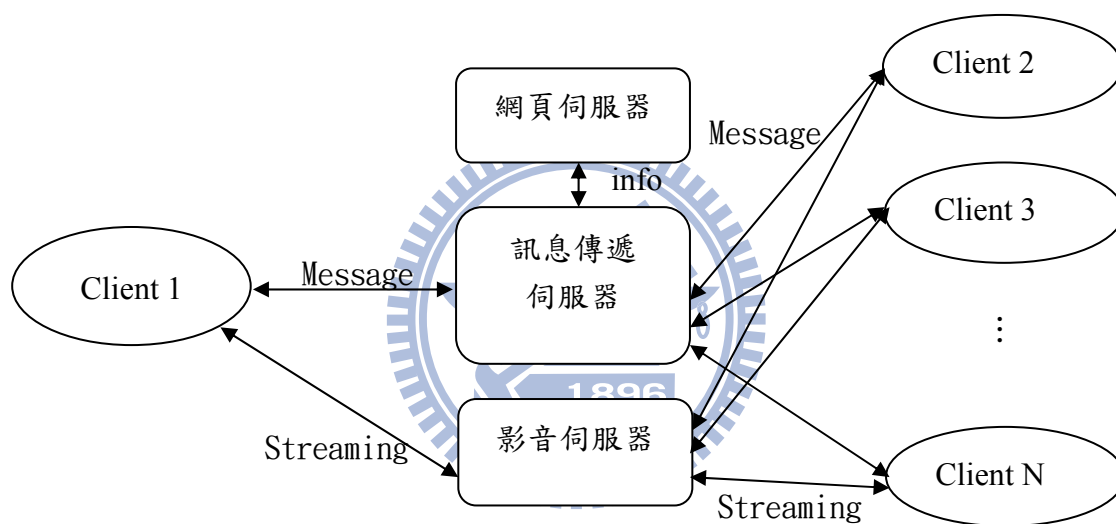


圖 3-11 多人連線系統架構圖

每個客戶端皆會與訊息傳遞伺服器有 Message 的傳輸，與影音伺服器有 Streaming 的傳輸。發佈端 Client 1 發佈有視訊和音訊的串流給 $N-1$ 個 Client 接收，而其它接收端 Client 2 到 Client N 每個都會各自發佈音訊串流給自己以外的 Client，同時每個 Client 也會接收來自 Client 1 的影音串流，形成完全圖網狀架構。

3.1.5.2 流量結構

N 個成員網路流量推算如下：

假設發佈端 Client 1 的影音串流網路流量為 B ，接收端的 Client 2~Client N 發佈的音訊串流網路流量為 S (通常 B 遠大於 S)，則影音伺服器的輸入、輸出流量為：

$$\text{輸入} = B + (N-1) \times S \quad (3.1)$$

$$\text{輸出} = B \times (N-1) + S \times (N-1) \times (N-1) \quad (3.2)$$

客戶端輸入至伺服器的流量結構如圖 3-12 左，伺服器輸出至客戶端的流量結構如圖 3-12 右。

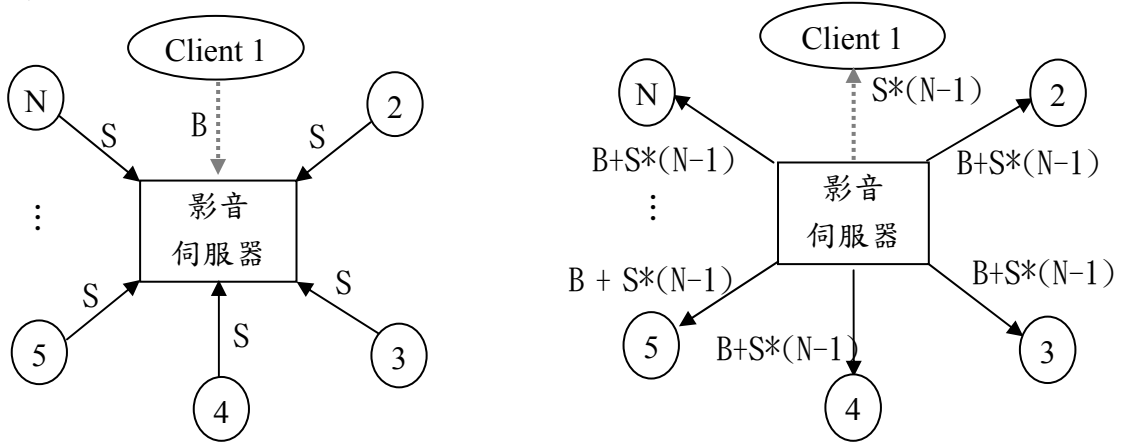


圖 3-12 左：輸入至影音伺服器流量示意圖；右：從影音伺服器輸出流量示意圖

(3.1) 式的計算是影音伺服器要收到一個影音串流和 $N-1$ 個音訊串流，(3.2) 式是影音伺服器要送出 $N-1$ 個影音串流和有 $N-1$ 個 Client 各送給其他 $N-1$ 個 Client 音訊串流，總頻寬為：

$$\text{成員頻寬} = \text{輸入} + \text{輸出} = N \times B + S \times (N-1) \times N \quad (3.3)$$

所以當成員人數增加，影音串流頻寬與人數成正比，音訊串流與人數平方成正比。

3.1.5.3 加入訪客後的流量結構

接著分析加入訪客到此活動的情況。影音伺服器會將其他成員送來的影音串流都送給已存在房間中的所有訪客。若有 M 位訪客在此房間中，則影音伺服器的輸出頻寬為 $B \times M$ ；發佈端 Client 1 的影音串流要送給 M 個訪客，其他接收端 Client 2~Client N 的成員會發佈音訊串流亦會送給這 M 個訪客，輸出頻寬為 $S \times M \times (N-1)$ ，所以 M 位訪客對於影音伺服器所造成的輸出頻寬負擔為：

$$\text{輸出} = B + S \times M \times (N-1) \quad (3.4)$$

而這 M 位訪客對影音伺服器皆不會發佈任何串流，則此 M 位訪客對於影音伺服器所造成的輸入頻寬負擔為：

$$\text{輸入} = 0 \quad (3.5)$$

故 M 位訪客對影音伺服器的總頻寬負擔為

$$M \text{ 位訪客總頻寬} = B \times M + S \times M \times (N-1) \quad (3.6)$$

考慮 N 位成員和 M 位訪客對影音伺服器所造成的總頻寬負擔為：

$$\begin{aligned} \text{影音伺服器總使用流量} &= (3.3) + (3.6) \\ &= N \times B + S \times (N-1) \times N + B \times M + S \times M \times (N-1) \\ &= B \times (M+N) + S \times (N-1) \times (M+N) \end{aligned} \quad (3.7)$$

本結果將會在之後的 3.2.1 影音合併或分流設計小節使用。

3.2 影音串流設計

使用者在本平台進行進行視訊連線活動時，會與其他使用者進行視訊、音訊的傳送，以達到雙方能夠看到相同的畫面和聽到對方的聲音的目的，其中使用者客戶端之間的傳輸媒介，即稱為串流媒體 (Streaming Media)，簡稱串流，使用的通訊協定為 RTMP，串流可以包含視訊流、音訊流和資料流三種。

由客戶端網路攝影機產生的連續畫面為視訊流，麥克風擷取到語音為音訊流，本研究將探討視訊流和音訊流合併傳送或分流的傳輸結構比較，了解資料在頻寬不足或伺服器忙碌時，客戶端丟棄資料的情形，其中音訊串流被丟棄時會很明顯被使用者查覺甚至感到不適。本研究將著力在設計方法以解決資料丟棄的問題。

3.2.1 影音合併或分流設計

一個串流可以同時傳送視訊和音訊，也可以僅傳送其中一個，合併傳送係指視訊和音訊結合成單一串流檔案發送至伺服器；分流指的視訊與音訊各分開成獨立的串流，會在伺服器端儲存兩個串流檔案。

當網路頻寬不足或是串流資料量太大發生網路擁塞現象時，會有封包在傳送前就被

客戶端丟棄的情形。一般情況下，人對聽覺會比視覺更為敏感，當音訊發生不連續的情況，會比視訊畫面的停格不連續還要令人難以忍受，確保音訊傳輸穩定成為我們主要的課題。

3.2.1.1 串流傳輸結構設計

首先分析串流合併與分流的傳輸結構

以雙人連線的架構圖為例

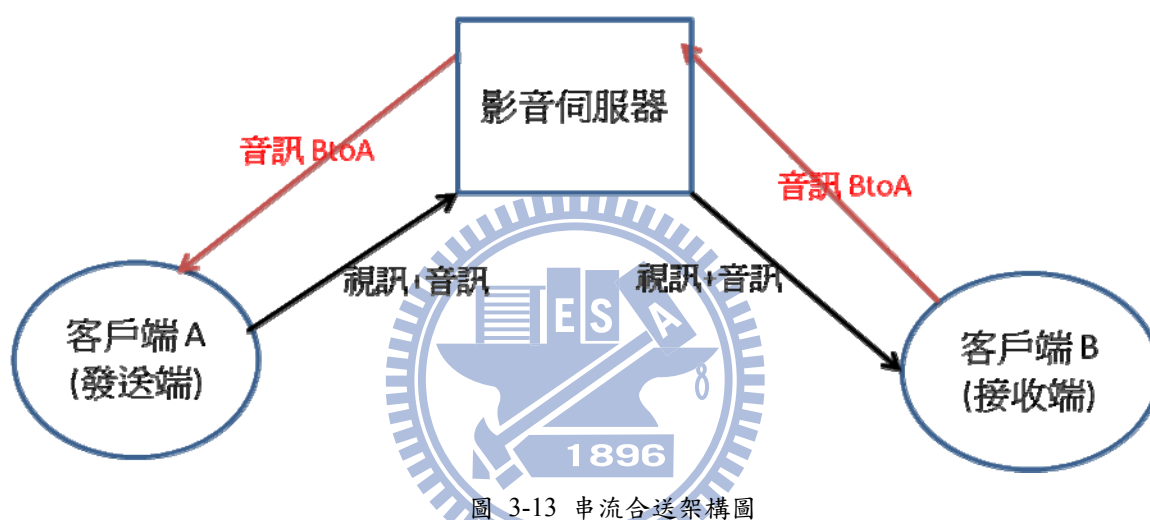


圖 3-13 串流合送架構圖

如圖 3-13，串流合併時共有二條串流：視訊+音訊、音訊 BtoA

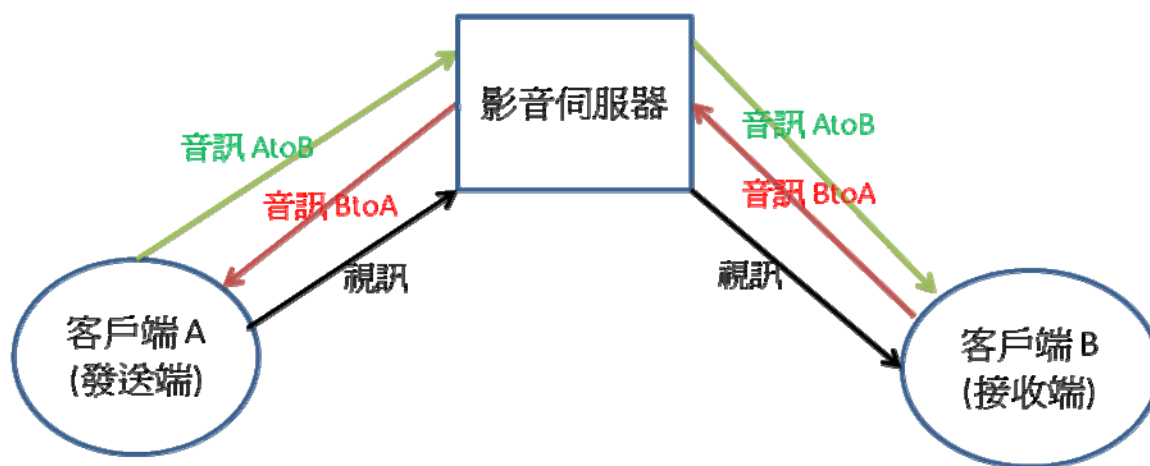


圖 3-14 串流分送架構圖

如圖 3-14，串流分流共有三條串流：音訊 AtoB、音訊 BtoA 和視訊

每個串流經過影音伺服器時會被錄製下來，當要影音重現播放這三個串流錄影檔時，需要知道各串流錄製開始時間，才能對準正確的開始播放時間，將會遭遇到各串流同步的問題（在 3.5 節有進一步介紹）。

3.2.1.2 二客戶端串流發送切換的設計

在本研究中，發佈端和接收端的角色是可以互相切換的，發佈端可以將發佈影像的權限交給其它接收端，然後自己轉為接收端。本論文稱具有發佈影像權力的為**發佈權**，具有發佈權的客戶端稱為**發佈端**。所有成員客戶端皆可以提出切換發佈。

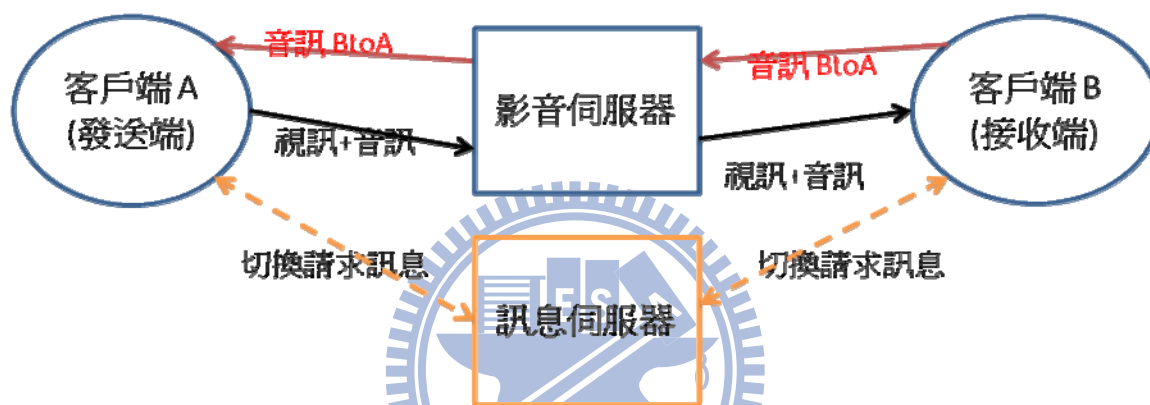


圖 3-15 客戶端提出切換請求的示意圖

所有的切換請求皆須要透過訊息傳遞伺服器傳給另一個客戶端才能進行動作，此活動內的成員皆可以提出切換發佈權請求，依據發佈權所在的客戶端不同會有不同的切換指令：

(1) SendPublishKey

具有發佈權的客戶端（發佈端）可指定要將發佈權交給哪個客戶端，則會送出

「*SendPublishKey*」訊息給目標客戶端，若目標有多個攝影機，同時可指定要顯示哪個攝影機的畫面，只要該客戶端同意就會開始進行切換流程

(2) RequestPublishKey

不具有發佈權的客戶端（接收端）可向有發佈的客戶端（發佈端）提出取得發佈權請求，則會送出「*RequestPublishKey*」訊息給發佈端，只要該發佈端同意就會開始進行切換流程。

依據串流的合併或分流傳送討論切換流程：

(1) 串流合併時的切換流程

客戶端 A 欲和客戶端 B 交換發佈權

步驟一：如圖 3-16，切斷客戶端 A 發佈的影音串流（視訊+音訊）和客戶端 B 的發佈音訊串流⁴，所以其他客戶端都會失去接收這兩個串流（從 A 發佈的視訊+音訊、音訊 B）

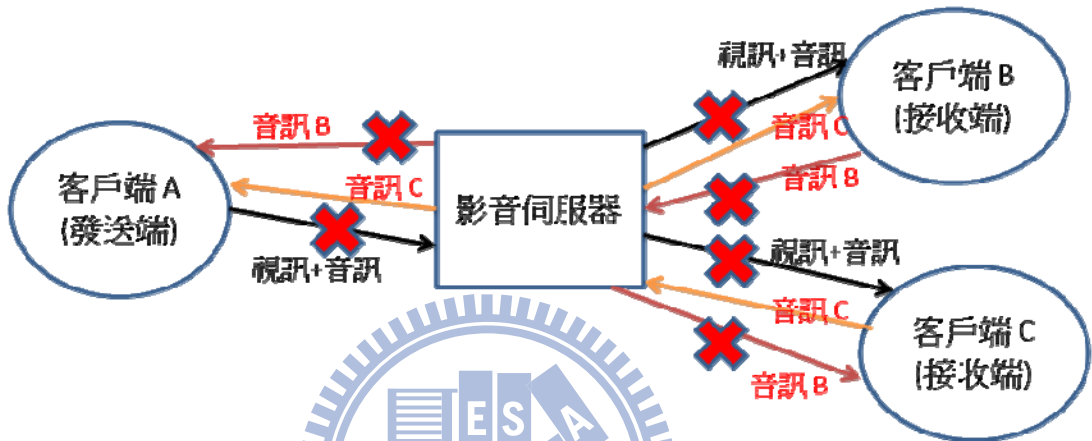


圖 3-16 合併串流切換步驟圖 1

步驟二：如圖 3-17，客戶端 B 從客戶端 A 取得發佈權，開始發佈影音串流（視訊+音訊），而客戶端 A 失去發佈權，但要重新發佈音訊串流，其他客戶端皆要重新接收這些新發佈端的串流：從客戶端 B 發佈的影音串流（視訊+音訊）、客戶端 A 發佈的音訊串流 A

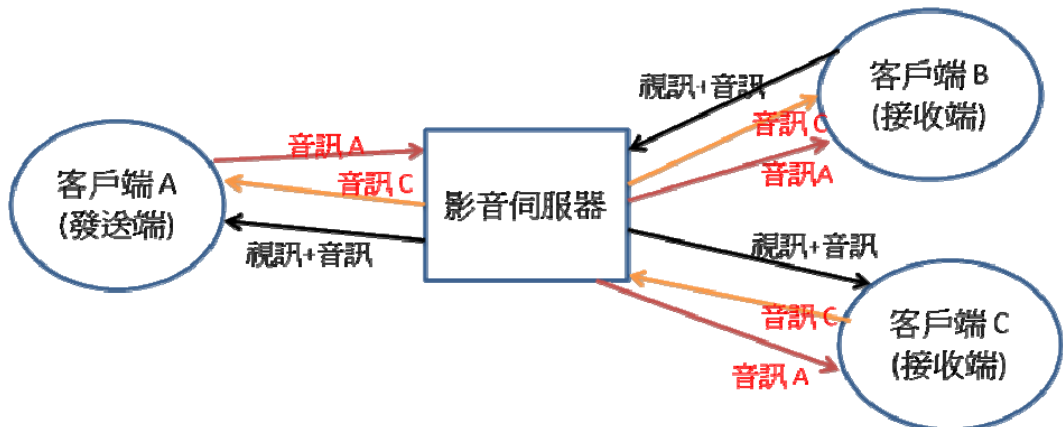


圖 3-17 合併串流切換步驟圖 2

⁴ 音訊串流、視訊串流、影音串流：一個串流中若只有音訊流，稱為音訊串流。一個串流中若只有視訊流，稱為視訊串流。一個串流中若同時有音訊流和視訊流，稱為影音串流。

(2) 串流分流的切換流程

客戶端 A 欲和客戶端 B 交換發佈權

步驟一：如圖 3-18，切斷客戶端 A 發佈的視訊，其他客戶端都會失去接收這個視訊串流。所有音訊串流皆保持傳送，不作任何變動。

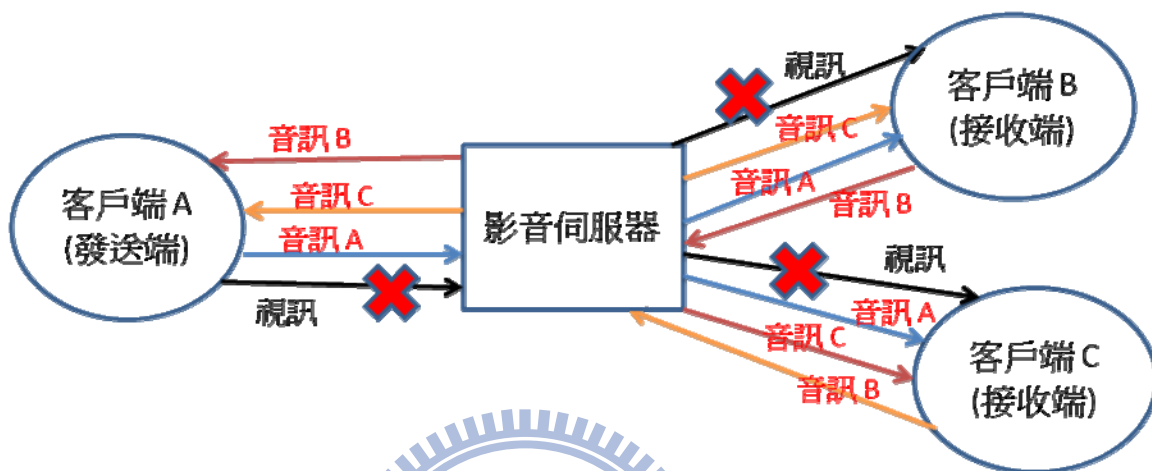


圖 3-18 分流串流切換步驟圖 1

步驟二：如圖 3-19，客戶端 B 從客戶端 A 取得發佈權，開始發佈視訊串流，而客戶端 A 失去發佈權，其他客戶端僅需重新接收新發佈端的串流：從客戶端 B 新發佈的視訊串流。

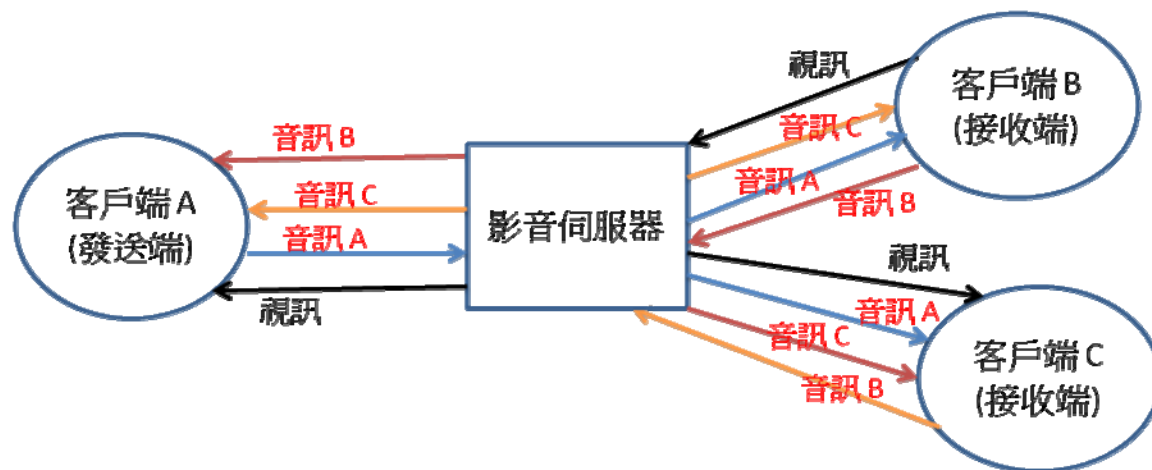


圖 3-19 分流串流切換步驟圖 2

3.2.1.3 流量結構比較

考慮 N 位成員和 M 位訪客對影音伺服器所造成的總頻寬負擔為：

(1) 發佈影音合送時：

影音伺服器收到一個影音串流（影像+聲音）流量 B ，轉送給 $M + N - 1$ 個客戶端，收到 $N-1$ 個音訊串流，每個平均流量 S ，轉送給 $M+ N- 1$ 個客戶端

(a) 影音串流流量：收到加上送出是 $B \times 1 + B \times (M+N-1) = B \times (M+N)$

(b) 音訊串流流量：收到加上送出是 $S \times (N-1) + S \times (N-1) \times (M+N-1) = S \times (N-1) \times (M+N)$

$$\text{影音伺服器總流量} = B \times (M+N) + S \times (N-1) \times (M+N) \quad (3.8)$$

(2) 發佈影音分流時：

影音伺服器收到 1 個視訊串流，轉送給 $M+ N- 1$ 個客戶端；收到 N 個音訊串流，轉送給 $M+ N- 1$ 個客戶端，

(a) 視訊串流流量：收到加上送出是 $V \times 1 + V \times (M+N-1) = V \times (M+N)$

(b) 音訊串流流量：收到加上送出是 $S \times N + S \times N \times (M+N-1) = S \times N \times (M+N)$

$$\text{影音伺服器總流量} = V \times (M+N) + S \times N \times (M+N) \quad (3.9)$$

若 $B = V+S$ ，即當單一影音串流流量 = 視訊流流量 + 音訊流流量

則可以由(3.8)式推導：

$$\begin{aligned} B \times (M+N) + S \times (N-1) \times (M+N) &= (V+S) \times (M+N) + S \times (N-1) \times (M+N) \\ &= V \times (M+N) + S \times (M+N) + S \times (N-1) \times (M+N) = V \times (M+N) + S \times N \times (M+N) = (3.9) \end{aligned}$$

則得出結論當 $B = V + S$ 時，則(3.8)=(3.9)，既當影音串流的流量 B 剛好為拆開後的視訊串流 V 和音訊串流 S 流量的總和時，串流合併和分流對影音伺服器的流量負擔是相同的。

舉一個實例：

若一個視訊流流量 20KB/s，音訊流流量 5KB/S，合送時影音串流為 20+5=25KB/s，有 5 位成員，10 位訪客，則以 $B=25$ 、 $S=5$ 、 $N=5$ 、 $M=10$ 代入合併時的流量公式，得

$$25*(10+5) + 5*(5-1)*(10+5)= 375 +300 = 675 \text{ (KB/s)}$$

以 $V=20$ 、 $S=5$ 、 $N=5$ 、 $M=10$ 代入合送時的流量公式，得分流時的流量為：

$$20*(10+5) + 5*5*(10+5)= 300 +375 = 675 \text{ (KB/s)}$$

3.2.1.4 儲存結構比較

串流合併傳送時，在影音伺服器的檔案系統中，成員客戶端發佈的影音串流（視訊＋音訊）皆會寫在同一個檔案，每次切換發佈權另一客戶端重新發佈的影音串流都會串接（append）同一檔案之後，切換發佈權只是換不同的成員客戶端來對這個檔案寫入。一個活動房間若有 N 個成員參與，同一時間會有 1 個影音串流和 $N-1$ 個音訊串流在伺服器上活動，而只要有切換發佈權的情況，原發佈端會重新發佈自己的音訊串流，則在影音伺服器的檔案系統中會有對應數量的音訊串流檔案，會有 1 個影音串流檔和 N 個音訊串流檔。

在串流分流的情況，成員發佈端發佈的視訊串流皆會寫在同一個檔案，每次切換發佈權新寫入的視訊串流都會串接（append）同一檔案之後，切換發佈權只是換不同的成員成為新的發佈端來對這個檔案寫入。而每個成員客戶端都會發佈自己的音訊串流，在影音伺服器上就會有對應數量的音訊串流檔案。可以歸納出一個活動房間若有 N 個成員參與，則在影音伺服器的檔案系統中會有一個視訊串流檔和 N 個音訊串流檔。而訪客接收端不具發佈串流到影音伺服器的權力，所以在伺服器端不會有其他串流檔案產生。

3.2.2 串流資料丟失問題與緩衝區設計

本節探討影音串流合併傳送和視訊、音訊分開傳送兩種情況，比較在頻寬不足時，資料的丟棄情況。我們從本論文 4.3.1 小節的測試結果可以得知，串流合併和分流皆會有丟棄視訊影格的情況，且隨可用頻寬減少，丟棄的視訊影格數越多。即使可用頻寬相當充裕，依舊會有丟棄影格的情形，如圖 3-20 所示。

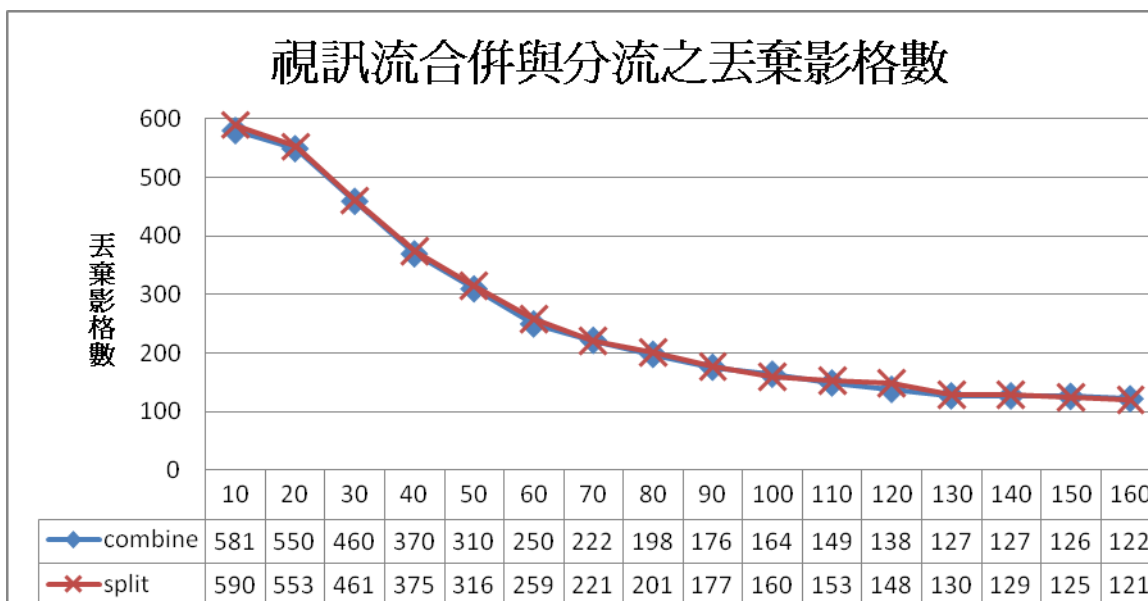


圖 3-20 測試 4.3.1 的結果：視訊流合併與分流之丟棄影格數

縱軸為 60 秒內丟棄影格總數，橫軸為限制可用頻寬 (KB/s)。表格中數據為發佈 400kbps 視訊流、44Kbps 音訊流時，combine 為影音合併傳送的視訊流、split 為分流傳送的視訊流丟棄影格數

丟棄影格數多時，不僅會造成視訊連線進行時畫面不順暢的停格現象，在影音伺服器錄影存檔時也會出現錄影檔案有畫面破損的情況，且隨著視訊活動進行時的可用頻寬不足而更加嚴重。語音資料在頻寬不足時也會被丟棄，但是目前無法從 4.3.1 測試中的丟棄影格來得知被丟棄的資料量 (Kilo Bytes)，僅能從接收端實際播出的語音內容得知資料有破損和斷斷續續的情況。(音訊流資料丟棄量的測試方法會在下一段落介紹)。

為了解決即時視訊連線活動丟棄影格的情況，本論文設計加入串流的緩衝區，用來暫存無法及時送出的資料。

3.2.2.1 緩衝區的概念

緩衝區是一個記憶體空間，當發佈端產生好的串流資料要送到網路上時，若網路擁塞，則資料會先放入緩衝區，待網路較為順暢時再送出。若網路一直擁塞，導致緩衝區的資料一直增加，直到超過緩衝區的容量限制，此時會清空緩衝區，也就是直接丟棄這

些送不出去的資料，在伺服器端收到的串流就會發生掉格的情況。

發佈串流和接收串流皆可以設置緩衝區，其功能不盡相同，首先定義：

- (1) 緩衝區容量 (bufferTime, B_{MAX}): 可用的最大緩衝區大小，使用單位為秒，是依資料量(Bytes)和資料速率 (KB/s) 去推算，可以存放多少時間長度的串流。
- (2) 緩衝區使用量 (bufferLength, B_{NOW}): 目前該串流已使用多少了緩衝區記憶體，單位與上述(1)的相同

另外在 RTMP 通訊協定下，一個串流內的視訊流和音訊流緩衝區的設立是可以各自獨立的。在發佈端或接收端設立緩衝區其功能也不盡相同，以下詳細說明：

發佈端有緩衝區會造成的現象：

在發佈端的發佈串流設立緩衝區，對發佈端、接收端和伺服器端錄影檔的影響如表 3-2



表 3-2 發佈串流有緩衝區對各種客戶端的影響

發佈端	接收端	錄影檔
因資料送不出去而存放在緩衝區，當緩衝區使用量成長，則會造成接收端暫時收不到畫面；當緩衝區使用量開始減少時，接收端則會開始收到畫面	若發佈端緩衝區使用量大於零，則接收端會依發佈端緩衝區使用量增加而呈現短暫看不到畫面的現象、畫面停格、FPS 下降，直到發佈端的緩衝區使用量減少。此時接收端收到畫面時會有畫面加速，FPS 上升的情況。	若錄影過程都無 buffer 填滿以至於丟棄影格的情況，則錄影畫面非常順暢，反之若有填滿緩衝區導至丟棄資料，則在錄影檔會出現掉格或跳格的現象。

表 3-3 發佈端緩衝區滿時會出現的現象：

發佈端	接收端	錄影檔
丟棄影格，清空 buffer，接著送出最新的影像	當發佈端緩衝區清空為零的瞬間，接收端會收到最新畫面	被丟棄的 Frame 會造成影像掉格並停頓，錄影檔會有停頓的現象

※接收端有緩衝區會造成的現象

接收端：會依緩衝區大小有初始緩衝時間 (initial time)，需要花時間載入足夠的影像長度才會開始播放，初始載入時間依照緩衝區容量 B_{MAX} 決定。當載入完成後，開始播放影片，隨著播放進行，緩衝區逐漸被清空，又依據網路速度而逐漸從伺服器端取得影片載入至緩衝區中，形成消長現象。在緩衝區未被清空前，進行的影片播放必定是順暢連續的，一旦被清空，播放就會被暫停，開始進行緩衝回填，直到填滿為止。

接收端緩衝區清空時會造成的影像

接收端：初始緩衝區必定為空，此時會依據所給定的緩衝區容量來決定初始的載入資料量。隨著緩衝區容量設定的高低，則使用者必須要等到緩衝區載入足夠的緩衝區使用量 B_{NOW} 影片才會開始播放，使用者等待時間與緩衝區容量、發佈端產生的資料量成正比，

與接收端的頻寬成反比。

因接收端的串流是從影音伺服器接收下來的，故接收端設置緩衝區對發佈端和錄影檔的串流檔案皆無影響。

對於發佈端或接收端緩衝區設立有無的分析討論：

- (1) 發佈端使用緩衝區＋影音串流合併傳送：聲音與影像使用單一串流，確保影音同步。發送端使用緩衝區可以使得在影音伺服器上錄影品質較佳，但會受到目前緩衝區使用量增減，造成影像會有停頓的現象。
- (2) 發送端使用緩衝區＋影音分流模式：聲音與影像分成各自獨立的串流，必須找到一個合適的串流資料量調控機制，避免目前緩衝量成長太高，造成接收端影音不同步、影像不順暢或接收端總是收到舊畫面等問題。
- (3) 接收端使用緩衝區＋影音串流合併傳送：適合用在單人模式時的訪客觀看接收端，訪客接收端可以看到較為完整連續的畫面，但是頻寬不足時會時常出現緩衝等待。
- (4) 接收端使用緩衝區＋影音分流模式：不適合在接收端使用緩衝區。起始緩衝等待會造成即時視訊連線（Live）一開始發佈端看到的主畫面和接收端收到的主畫面有嚴重的時間差，此時間差隨著設定的緩衝容量而增加，同時因為影音串流分開，影像接收的緩衝區較難以填滿（影像資料量較大），而聲音串流接收緩衝區較容易填滿，使得聲音和影像的初始緩衝等待長度不同，會出現影音不同步的現象，而且此現象亦會隨著雙方連線進行久了，影音不同步的時間差會更為嚴重，因影像緩衝區較容易清空而再度出現緩衝等待，聲音緩衝區則是可以一直維持接近滿的狀態（因所需頻寬小），。

找出合適的緩衝區容量的目地：

- (1) 避免發佈端的緩衝區容易被填滿，導致要執行清空緩衝區，造成畫面影格被丟棄，

錄影檔有破損情況。

- (2) 在發佈端緩衝區成長時，同時會造成接收端無新畫面而有停頓現象，此時聲音依舊繼續傳送，畫面停頓的越久，影音就越不同步，直到發佈端清空緩衝區。合適的緩衝容量可減少 Live 時接收端的影音不同步
- (3) 上述 1、2 有互相抵觸，所以要找一個介於中間安全值，首先是減少發佈端的資料量，降低填充緩衝區的速率，接著訂定一個合適的緩衝容量用以維持 Live 順暢和錄影完整。太高則 Live 影音不同步嚴重，但錄影畫面較為完整連續；太低則錄影畫面停格多，且被丟棄的影格多，但 Live 較為順暢。

3.2.2.2 丟棄資料量計算

因無法從前一節描述得知音訊串流被丟棄了多少影格，在發佈端有了緩衝區的設計之後，本論文改採用記錄發佈端產生串流的資料量 D_{CRE} (Creative Data) 和發佈端緩衝區使用量 B_{NOW} ，相減之後來計算實際從客戶端送出到伺服器端的資料。

發送至影音伺服器的資料量為

$$D_{PUB} = D_{CRE} - B_{NOW} \text{ (Bytes)} \quad (3.10)$$

留存在緩衝區未送出的資料 B_{NOW} ，對照在未使用緩衝時可將 B_{NOW} 視為丟棄，藉此計算資料丟棄量 D_{LOS} ：

$$D_{LOS} = D_{CRE} - D_{PUB} = B_{NOW} \text{ (Bytes)} \quad (3.11)$$

資料丟棄率為：

$$R_{LD} = D_{LOS} / D_{CRE} \text{ (\%)} \quad (3.12)$$

給予足夠大的緩衝區大小，排除因為緩衝區被填滿而丟棄資料的可能，分別記錄視訊、音訊流各自產生的資料量和緩衝區使用量，以此推算出資料被丟棄了多少。

本論文設計了一個測試，以可用的網路頻寬為變因，觀察可用頻寬增加時，串流合併、分流的視訊和音訊流的資料丟棄率，測試結果如圖 3-21、圖 3-22。(詳細介紹在本論文 4.3.2 小節中)

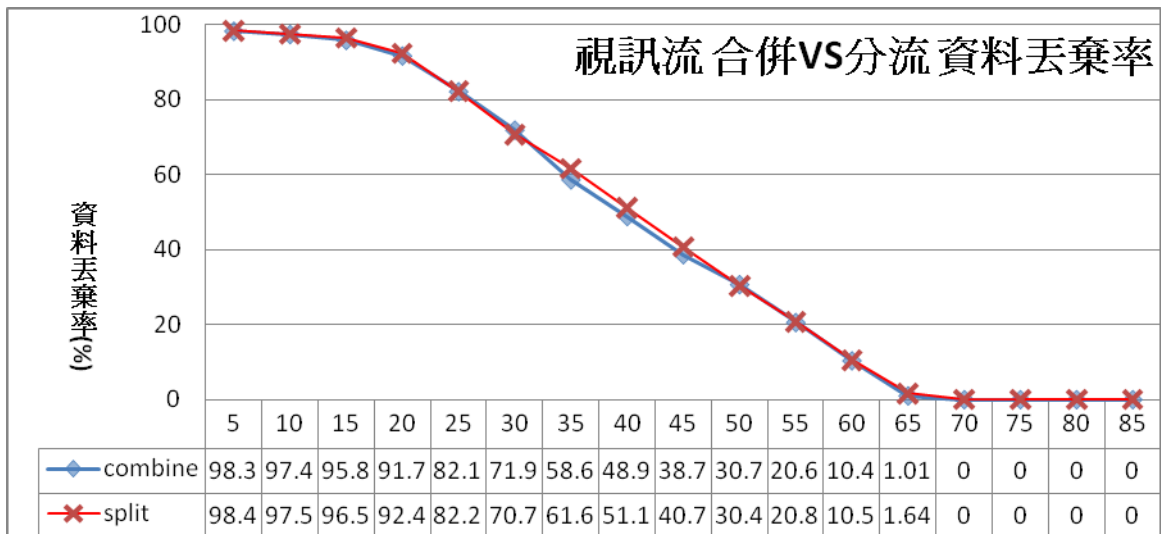


圖 3-21 測試 4.3.2 的結果：視訊流合併和分流之資料丟棄率比較圖

縱軸為資料丟棄率(%), 橫軸為頻寬限制(KB/s), 圓點 combine 表示影音合併傳送、方點 split_video 表示影音分流的視訊流資料丟棄率。表格中數據為發佈 400kbps 串流在不同頻寬限制下的資料丟棄率。

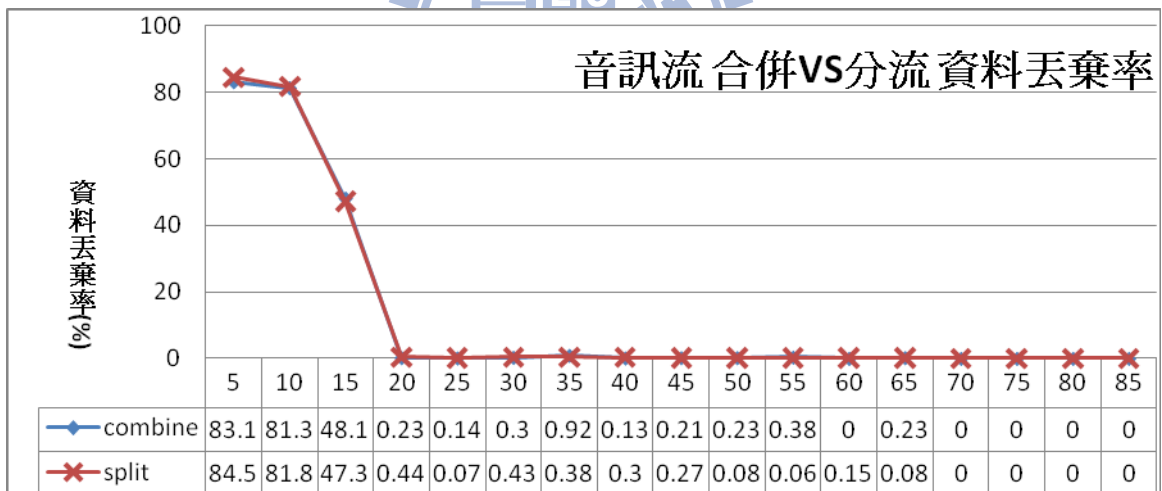


圖 3-22 測試 4.3.2 的結果：音訊流合併和分流之資料丟棄率比較圖

縱軸為資料丟棄率(%), 橫軸為頻寬限制(KB/s), 圓點 comb_video 表示影音合併傳送、方點 split_video 表示影音分流的音訊流資料丟棄率。表格中數據為發佈 400kbps 串流在不同頻寬限制下的資料丟棄率。

視訊和音訊串流合併、分流的資料丟棄率相近，串流傳輸結構不是影響資料丟棄的主因。如圖 3-21、圖 3-22，資料丟棄率顯然受到可用頻寬的影響，產生的資料量無法由目前頻寬消化出去時，在無緩衝區時，送不出的資料會在客戶端直接被丟棄，有緩衝

區時則會送入緩衝區暫存，只要頻寬稍有空缺就會先將緩衝區的資料送出。

另可觀察出在 20KB/s 的頻寬限制下，音訊流的資料丟棄率已低於 1%，但是視訊流的資料丟棄率遠大於音訊流，顯示在頻寬不足時，視訊流的資料比音訊流更容易被丟棄，而音訊流因所需頻寬遠小於視訊流，而較不會被丟棄。最根本的原因是，一般情況下，發佈端產生的視訊流資料速率(50KB/s)會遠大於音訊流(約 5KB/s)，在頻寬不足時丟棄的速率亦是視訊大於音訊流，因此產生了另一個問題——影音同步問題，將在下一小節 3.2.3 有進一步的介紹。

正常情況下，可用頻寬大多是固定在一個範圍內，變動的通常是發佈端能產生的資料量，使用者可以藉由改變網路攝影機的畫面解析度、影格速率來變動資料量。若發佈端產生資料的速度一直大於可用頻寬的消化速度，緩衝區使用量會不斷成長，造成發佈端與接收端的畫面不同步。直到緩衝區使用量超過上限，發佈端會直接清空緩衝區，緩衝區使用量歸零，影音伺服器收到的串流存檔會出現畫面破損或掉格等問題。

綜合 3.2.1 和 3.2.2 小節，本平台使用的串流設計為：

- (1) 必須加入發佈串流的緩衝區，以減少發佈串流資料被丟棄的問題。
- (2) 接收端不設立緩衝區，因初始填滿緩衝時間會造成客戶端間發出和接收的影音串流不一致。
- (3) 在設立發佈串流緩衝區時，合併與分流對視訊、音訊的資料丟棄率幾乎沒有影響。

考量影音伺服器存檔單一檔案較易管理和避免重現播放系統在播放分流的雙串流播放需對時的問題。故本平台採用：將視訊流與音訊流合併為單一串流（即影音串流）傳送，作為本平台客戶端與伺服器端的串流傳輸結構。

3.2.3 影音同步問題

為了減少視訊和音訊流資料破損或丟棄的情形，也確保伺服器串流錄影存檔的完整性，本平台須在發佈端設立串流緩衝區。

在有緩衝區的情況下，發現的影音同步問題泛指以下兩種：(1)發佈端影音同步問題、(2)客戶端間影音同步問題。

3.2.3.1 發佈端影音同步問題

發佈端發佈的視訊和音訊流無法對準同一個時間點送出，造成接收端在收到資料後，使用者感受到影音不同步的情況，稱之為「發佈端影音同步問題」

一般情況下，發佈端產生視訊流的資料量會遠大於音訊流的資料量，造成視訊流容易受到頻寬不足影響（可由測試 4.3.2 結果得知），也因此視訊流的緩衝區使用量比音訊流更容易增加。一旦視訊流與音訊流緩衝區使用量不相等，即視訊流有資料卡在發佈端送不出，但是音訊流的資料已經順利送到接收端，此時接收端收到的最新視訊流和音訊流的時間點不一致，出現影音不同步的現象。

3.2.3.2 客戶端間影音同步問題

發佈端或接收端本身對串流收發的處理時間、網路傳輸時間或伺服器處理時間等皆會造成發佈的串流在一段延遲時間之後，接收端才收到最新的串流資料，使得不同客戶端收到最新串流的時間點不一致，稱之為「客戶端間影音同步問題」

在進行一個多人視訊連線時，當發佈端產生的資料無法順利送出而暫存在緩衝區時，將緩衝區使用量換算成時間(秒)，這個時間可以直接換算成接收端需等待最新串流送達的延遲時間，稱為「緩衝延遲時間」。此時間增加，則所有接收串流的客戶端從影音伺服器收到最新串流內容的一致性就越低。緩衝延遲時間是客戶端可以控制的因素。

緩衝延遲時間只是造成客戶端影音同步問題的一部分，客戶端間影音同步問題還需

要考慮使用者處理時間、伺服器處理轉送時間和網路傳輸的延遲時間。

客戶端可控制因素 VS 不可控制因素

發佈端影音問題主要是客戶端因緩衝使用增加而產生的，屬於客戶端可控制因素；客戶端間影音同步問題則同時包含可控制和不可控制因素，前一段提到的「緩衝時間」為客戶端可控制因素，而客戶端處理時間、網路延遲時間、伺服器轉送時間等皆為客戶端不可控制因素。4.2.1 小節測試結果顯示了伺服器轉送、網路傳輸、客戶端處理時間等客戶端不可控制的因素所造成的延遲時間。

本論文僅針對在客戶端可以控制的參數去改善解決同步問題：盡可能壓低了緩衝區使用量造成的緩衝延遲時間。在 3.3 節將介紹為改善此兩個影音同步問題而設計的「串流資料量調控演算法 (Streaming Data Control Algorithm)」。

3.3 串流資料量調控演算法設計

3.3.1 設計概念

為了改善 3.2.3 小節的使用緩衝區的影音同步問題。控制發佈端視訊流和音訊流產生的資料量，並且監控各自的緩衝區使用量是接近或相等。主要的設計概念為：

(1) 控制發佈端產生資料量

緩衝區使用量增加的主因是發佈端產生的資料量無法被目前可用的頻寬所消化送出。因客戶端目前可用頻寬難以被使用者控制，故本論文採用控制發佈端產生的串流資料量。

(2) 只需監控視訊流

因音訊流對頻寬的使用量遠小於視訊流，且由圖 3-21、圖 3-22 和測試 4.3.2 結果得知，視訊流的緩衝區使用量會比音訊流的緩衝區使用量更容易成長，故僅需監控視訊流的緩衝區使用量，並依此去調整發佈端產生視訊流的資料量，使其與音訊流緩衝區使用量幾乎相等（盡可能趨近空）。

本論文設計了一套能夠依據目前緩衝區使用量、畫面影格速率來控制發佈端產生串流資料量的演算法，以符合目前的頻寬限制，使得目前緩衝區的緩衝量盡可能維持在空。同時進一步依據視訊畫面的變動程度（活動劇烈或是靜止），來決定要調控視訊流產生參數（畫面解析度、資料速率、限制影格速率）的優先順序。

由上述可知，串流資料量調控演算法是「Camera control」的型式再結合控制客戶端編碼器壓縮率以改變資料速率，更進一步的控制資料量。

3.3.2 串流資料量調控演算法

由前述 3.2 節介紹可知串流緩衝區使用量是衡量影音同步問題的依據，在本小節皆以時間(秒)為緩衝區使用量的單位。

3.3.2.1 輸入與輸出

發佈端視訊流取得的參數，輸入演算法：

- (1) 目前影格速率 (Current Frame per Second, C.FPS)：實際網路攝影機每秒擷取的畫面張數。
- (2) 緩衝區使用量 (BufferLength, B_{MOW})：緩衝區中影像資料的時間長度，若緩衝區中有 2 秒長的資料，意謂著發送端目前的畫面，至少要經過 2 秒以上(不考慮網路傳送時間)才會出現在接收端上。緩衝量會到緩衝區容量 (bufferTime, B_{MAX}) 限制，當緩衝區使用量超過緩衝區容量的兩倍時，會清空緩衝區，此時接收端就可以收到最新的資料，使得影音串流出現同步，但也意味著發佈端將丟棄資料，造成錄影資料破損。
- (3) 畫面動量 (Motion Level)：偵測視訊畫面內容的變動程度，此數值越高表示畫面變動越大。

表 3-4 演算法運作所需的輸入參數

參數名稱	範圍	使用者觀點
(1)C. FPS	0~25	越高畫面影格張數越多。
(2)BufferLength	0~10	越高影音越不同步、客戶端收到的畫面越不一致。
(3)MotionLevel	0~100	越高畫面變動程度越大。

經演算法運作後，輸出的主要參數：

- (1) 限制影格速率 (Bounded Frame per Second , B. FPS)：控制網路攝影機每秒要擷取的畫面張數，以此控制產生的資料量。程式在運作時會在硬體能力內盡可能擷取到限制內的最大張數。
- (2) 資料速率 (Data Rate)：每秒網路攝影機產生的原始視訊資料會經發佈端程式的編碼器壓縮成一個指定的資料量。單位為 Kilo Byte per Second(KB/s)。若將資料速率指定的越小，則編碼壓縮率會增加，相對的呈現畫質會越差；反之，資料速率越高，則會降低壓縮率，呈現畫質較佳。
- (3) 畫面解析度 (Resolution)：控制網路攝影機的輸出視訊畫面大小，以控制資料量大小，單位為像素 (Pixel)。

表 3-5 串流資料量調控演算法主要調控的參數項目

參數名稱	範圍	使用者觀點
(1)B. FPS	0.1~25	影響視訊畫面的流暢程度。
(2)Data Rate	20~200	影響視訊畫面的畫質。
(3)Resolution	160x120~ 960x720	影響視訊畫面的精細程度。

3.3.2.2 演算法流程

示意流程圖：

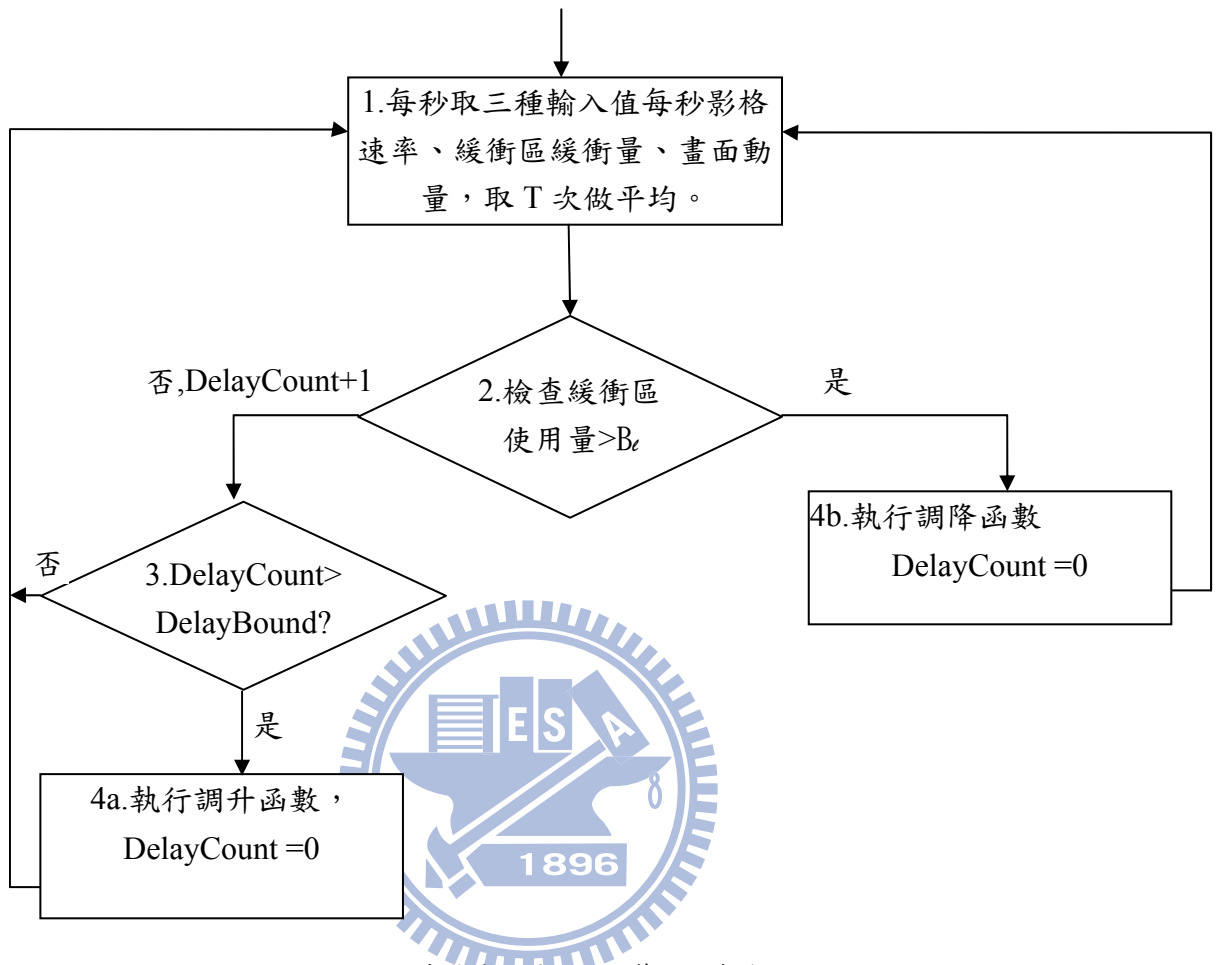


圖 3-23 串流資料量調控演算法示意流程圖

完整的演算法流程圖請見附錄 1。

表 3-6 演算法使用的主要變數

變數名稱	初始值	意義
T	5	步驟一取值的次數，會影響到演算法輸出結果的頻率。T 越低則演算法循環間隔越小，調控次數會很頻繁。
updown	False	若執行完調升後，又有調降的情形，此變數值設為 true
downup	false	若執行完調降後，又有調升的情形，此變數值設為 true
BufferLimit, B_c	0.5	緩衝限制容忍值，超過此值判定為會對使用者影音同步造成影響。此值越高，緩衝區使用量將有更多機會被填入。
DelayCount	0	延遲調升的計數值，當緩衝區使用量超過 B_c 時，此值會+1。
DelayBound	2	延遲調升的計數上限，超過此值才會進入調升函數。當穩定狀態啟動時，會提高上限
HighRate	0	當 <i>updown</i> 被設為 true 的同時，記下此時的資料速率，作為計算穩定狀態資料速率用。
LowRate	0	當 <i>downup</i> 被設為 true 的同時，記下此時的資料速率，作為計算穩定狀態資料速率用。
StableDataRate	0	由 <i>HighRate</i> 和 <i>LowRate</i> 透過公式運算出的穩定狀態資料速率，當進入穩定狀態時，資料速率會被設為 <i>StableDataRate</i> 。

演算法流程說明

每個步驟號碼可以對應到圖 3-23 流程圖中方塊內的編號。

步驟一：取值

每秒取值，取得(1) 目前每秒影格速率、(2) 緩衝區使用量、(3) 畫面動量，每秒取一次，共取了 T 次，將此 T 次的值取平均，即為 T 秒內的有效數據。接著將這三個有效數據輸入本演算法執行。

設計原因：

取 T 次做平均是為了拉長演算法執行的間隔時間，降低調升或調降的頻率。因每次執行調升或調降會去變更視訊擷取裝置的參數，有些參數如 $B.FPS$ 在設定時會受到硬體的限制而有延遲時間，同時會消耗非常多的電腦效能。因此每秒取一次值，每 T 次做平均，相當於每 T 秒才會執行一個循環，判斷要做調升、調降或不變更。

步驟二：

檢查平均緩衝區使用量是否高於 $BufferLimit B_c$ ，若超過則執行調降函數，反之則執行判斷是否為穩定狀態。

設計原因：

保持目前緩衝區使用量不超過 B_c 視為整個演算法的最優先目的，一旦緩衝區使用量成長，將會造成在即時視訊連線活動時出現雙方視訊不同步的情況，同時也會有先收到音訊後收到視訊情形，這個時間差受到發佈端的緩衝量多少來決定。

只要緩衝區使用量一超過 B_c ，馬上執行調降函數，以減少目前的緩衝區使用量。反之，若要執行調升函數，則要加上限制，因為執行完調升函數有機會造成目前緩衝區使用量的成長，而會再度執行調降函數，如此形成一個調升函數和調降函數交替執行的循環，本論文稱此循環為「擺盪現象」。

步驟三：

判斷目前 $DelayCount$ 是否超過 $DelayBound$ ，若否則回到步驟一繼續執行，若是則表示有可調升的空間，執行調升函數。 $DelayBound$ 這個值會受到目前是否為穩定狀態影響。若目前是穩定狀態，則 $DelayBound$ 的值會增加，使得要累積更高的 $DelayCount$ 才能執行調升函數。當演算法執行了調升和調降各一次之後，會推算出一個穩定值，使得系統維持在這個值不做調升。

設計原因：

增加 *DelayBound* 的限制是使提升資料速率的難度增加，因提升資料速率將有機會造成緩衝量成長。*DelayBound* 的值也受到目前是否為穩定狀態影響，當是穩定狀態時，*DelayBound* 值會增加，解除穩定狀態則會調回初始值。

為避免一直不斷執行調升或調降的擺盪現象，設計一個**穩定狀態判斷機制**。

(1) 進入穩定狀態的條件是：

當演算法曾執行「調降後馬上調升」時，*downup* 設為 true，並記下此時的資料速率為 *LowRate*；執行「調升後馬上調降」時，*updown* 設為 true，並記下此時的資料速率為 *HighRate*。執行到步驟三時，會判斷 *downup* 和 *updown* 兩個變數是否為 true，若是則判斷有擺盪現象，並將 *LowRate* 和 *HighRate* 兩者資料速率取平均作計算得到 *StableDataRate*。接著進入穩定狀態，將目前資料速率設定在 *StableDataRate*，並且 *DelayBound* 增加 10，使得執行調升資料速率變得困難，同時 *downup* 和 *updown* 皆設為 false。

(2) 解除穩定狀態的條件：

只要一執行調升或調降函數，就解除穩定狀態，將 *DelayBound* 設為初始值。

步驟四：調整與控制串流資料量

表 3-7 調升調降函數用的變數

變數名稱	初始值	意義
ActionBound	50	若 <i>Motion Level</i> 超過 <i>ActionBound</i> ，則判斷視訊畫面處於晃動狀態的依據。
MaxFPS	25	<i>B. FPS</i> 能調整到的上限值。
MinFPS	0.1	<i>B. FPS</i> 能調整到的下限值，必須大於 0 是因為當 <i>B. FPS</i> 設為 0 時，攝影機不會再擷取任何畫面。
MaxDataRate	200 KB/s	<i>Data Rate</i> 可以受演算法調整的上限。
MinDataRate	20 KB/s	<i>Data Rate</i> 可以受演算法調整的下限。

四 a. 執行調升函數

流程圖 (圖 3-24) :

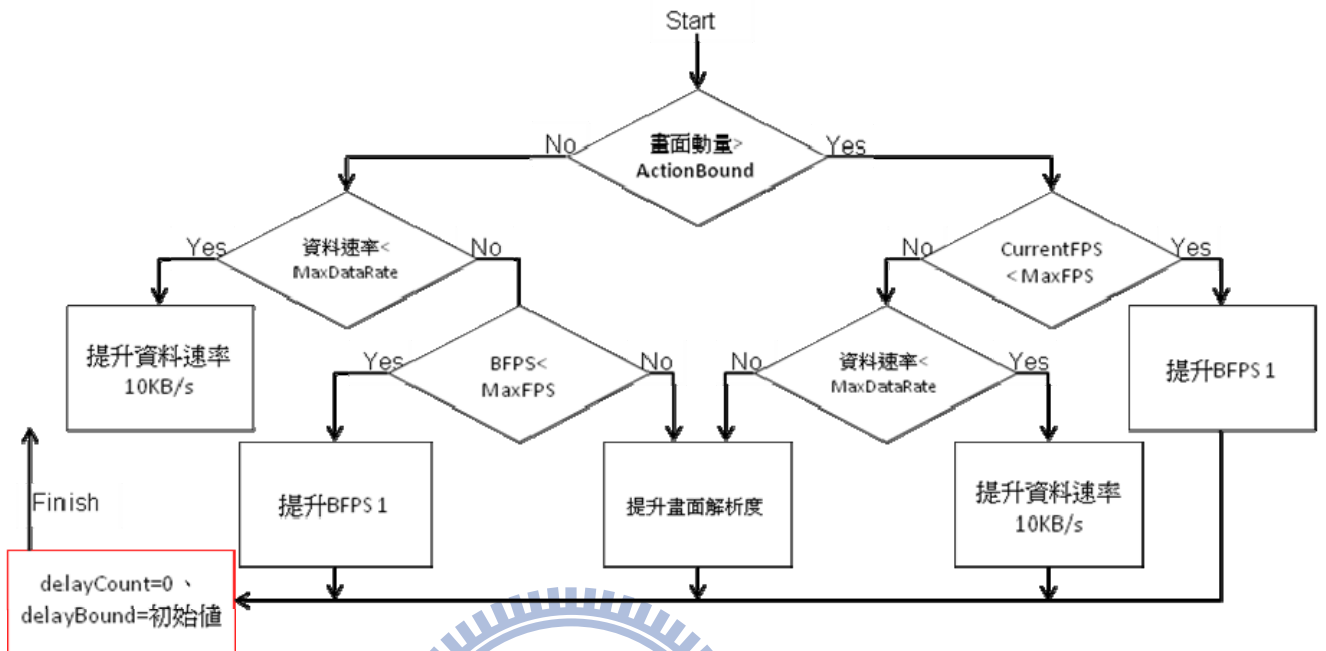


圖 3-24 調升函數運作流程圖

- (1) 由畫面動量可知目前發佈端現場的環境是否在活動中，活動量大、畫面變動多，則視訊擷取裝置的 *B. FPS* 也要相對提高，才能取得流暢的畫面。首先判斷畫面動量是否超過 *ActionBound*，若超過表示目前畫面動態需求高，則在調升時優先順序是：*B. FPS*、資料速率、畫面解析度；反之，未超過 *ActionBound* 則表示畫面動態需求低，調升時優先順序是：資料速率、*B. FPS*、畫面解析度。
- (2) 每次在做調升時都會檢查正要調升的項目是否達到最大值，若無法調升則會換下一個優先順序的項目做調升，直到成功調升或所有項目都無法調升為止。
- (3) 執行完調升函數，*DelayCount* 要歸零重新累計。同時，判斷目前不是穩定狀態，*DelayBound* 設為初始值。

四 b. 執行調降函數

流程圖 (圖 3-25):

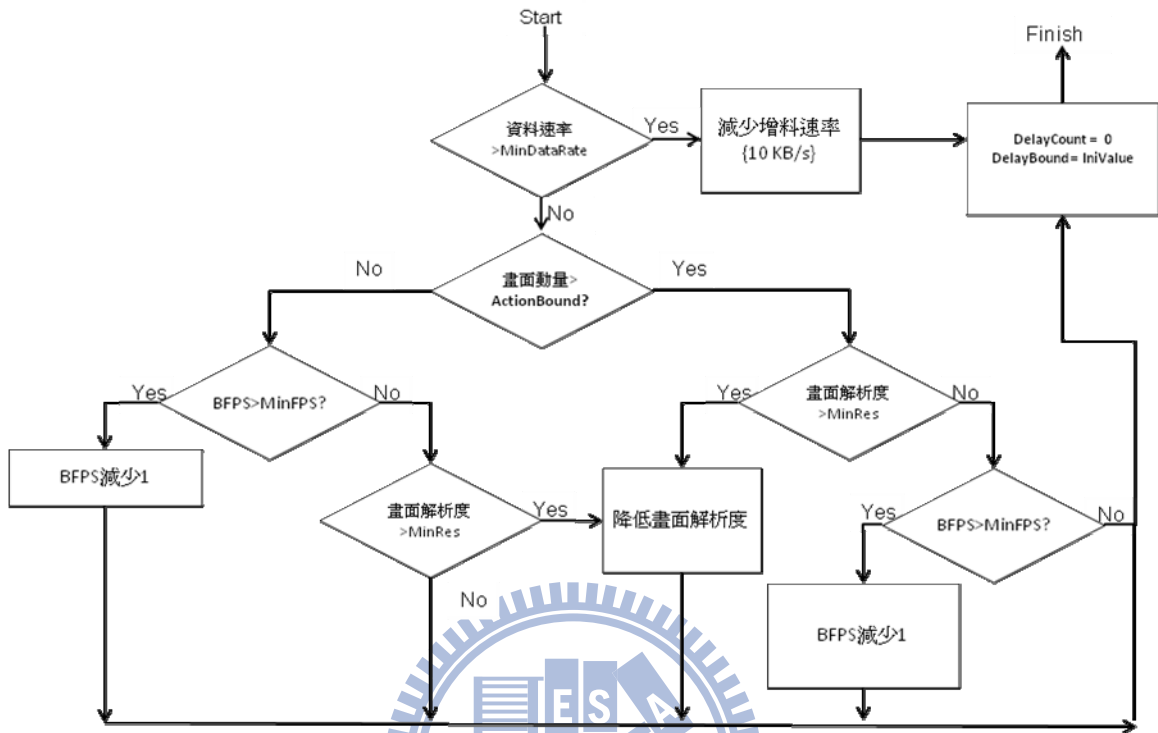


圖 3-25 調降函數運作流程圖

- (1) 當減少資料速率時，客戶端程式會提高串流壓縮比以符合減少後的資料速率，如此可以馬上減少串流資料量，故在演算法設計上會最先判斷是否還可降低資料速率，若資料速率未達最低值，則會馬上減少資料速率；反之，若已達最低值，則才會依畫面動量來決定要先調整 *B. FPS* 或是畫面解析度。
- (2) 若畫面動量超過 *ActionBound*，表示 FPS 需求高，則在做調降時，會先調降畫面解析度，最後才是 *B. FPS*；反之，若畫面動量未超過 *ActionBound*，表示目前較需要對畫面品質要求較高，則會先調整 *B. FPS*，最後才是畫面解析度。
- (3) 執行完調降函數，*DelayCount* 要歸零重新累計。同時，判斷目前不是穩定狀態，*DelayBound* 設為初始值。

3.3.3 設計實驗與討論

3.3.3.1 不同網路環境之演算法運行測試

測試目的：

在不同的網路環境下，比較使用或未使用串流調控演算法的資料丟棄率，藉此觀察改善資料丟棄和影音同步問題。

測試方法：

使用 NetLimiter 控制網路環境，每次設定不同的可用網路頻寬，發佈端使用足夠大的緩衝區以便能確實記錄填入緩衝區的資料量，發佈資料速率為 400kbps 的視訊流和 44kbps 的音訊流合併傳送到影音伺服器，記錄 60 秒後發佈端產生的資料量和緩衝區的資料量。用前述 3.2.2.2 小節介紹的公式(3.8)、(3.9)、(3.10)計算視訊流資料丟棄率。最後與圖 3-21 未使用演算法的影音合送視訊流資料丟棄率作比較和討論。

測試結果與討論：

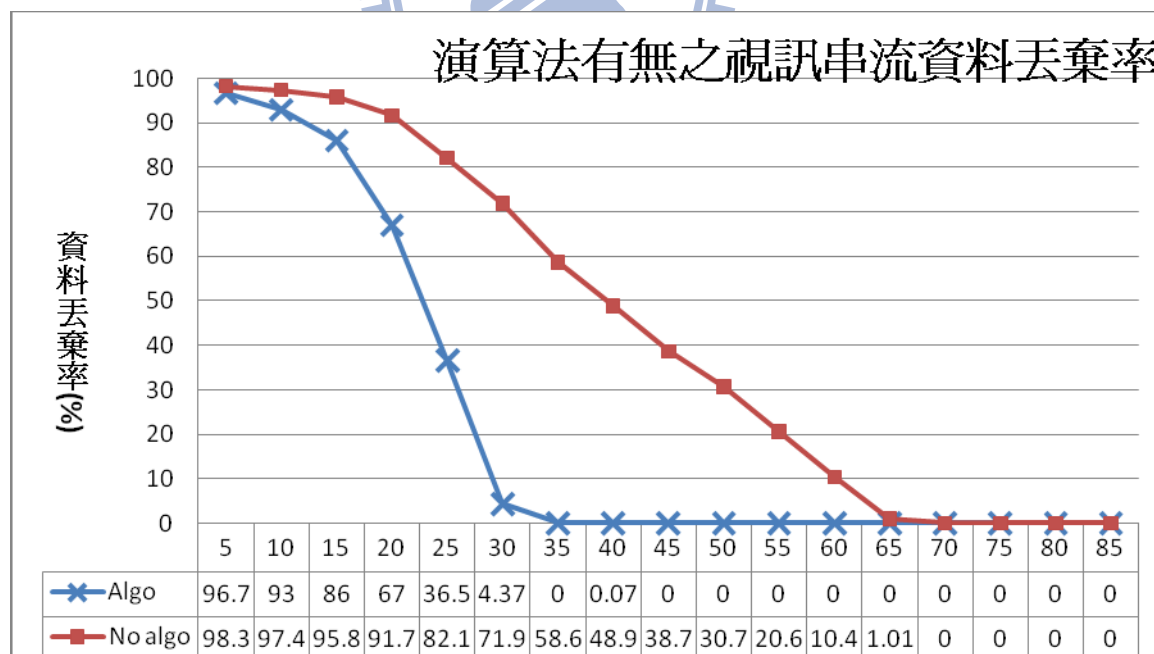


圖 3-26 演算法有無之視訊流資料丟棄率比較圖

■ 為未使用演算的資料丟棄率，✕ 為使用演算法的資料丟棄率，橫軸為可用的頻寬 (KB/s)，縱軸為資料丟棄率 (%)

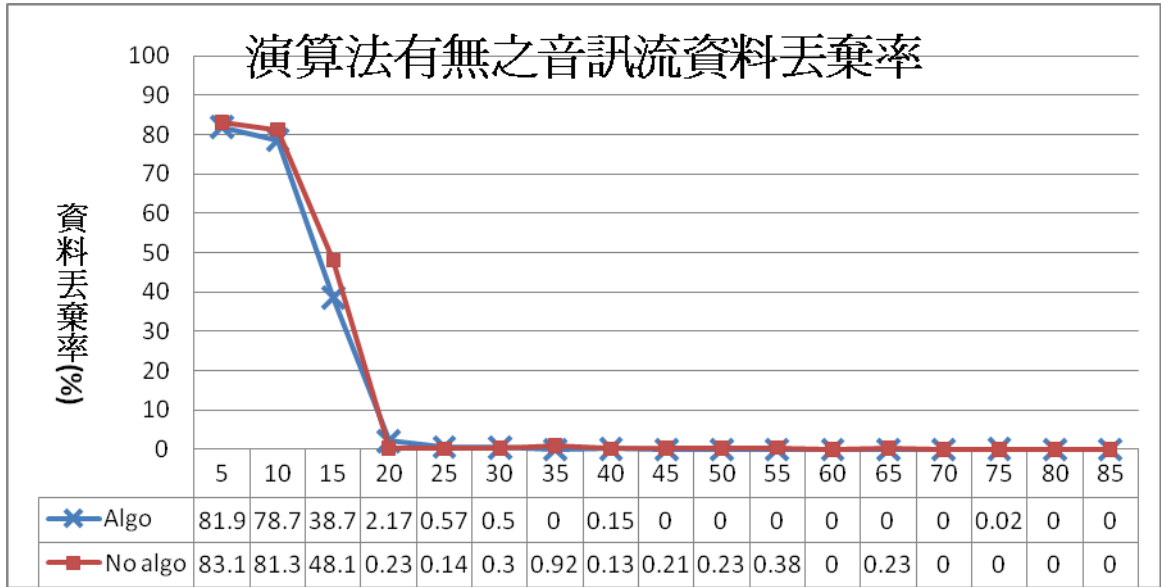


圖 3-27 演算法有無之音訊流資料丟棄率比較圖

■ 為未使用演算的資料丟棄率，✕ 為使用演算法的資料丟棄率，橫軸為可用的頻寬 (KB/s)，縱軸為資料丟棄率 (%)

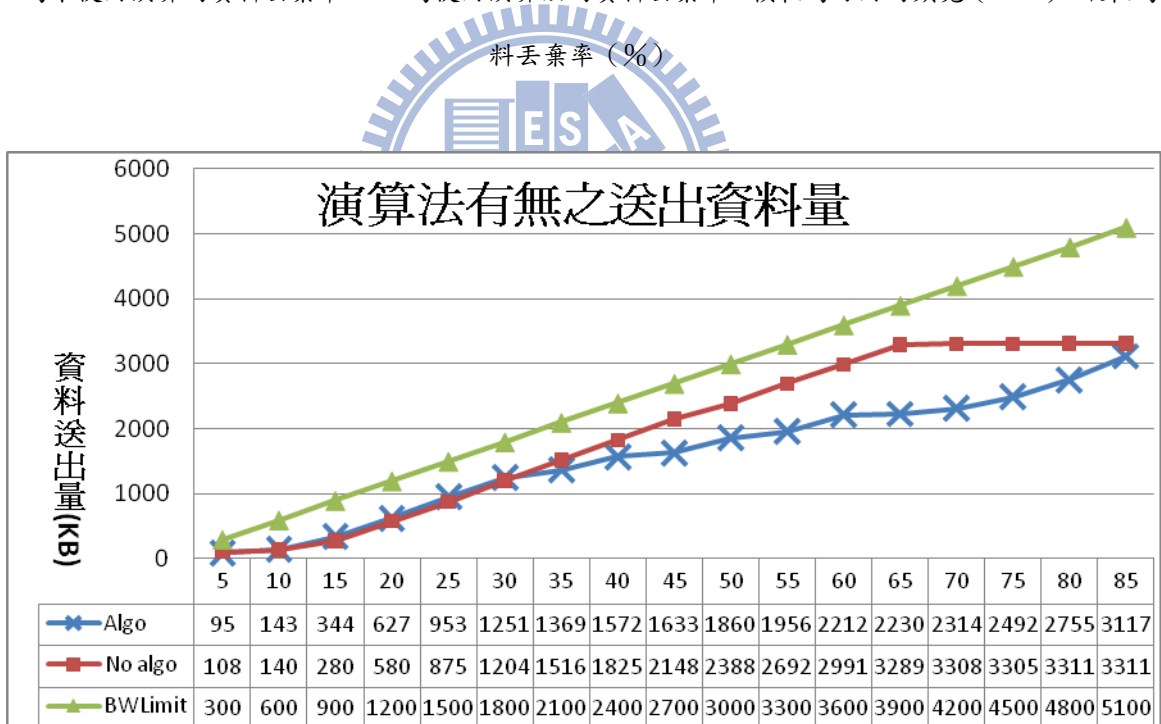


圖 3-28 演算法有無之視訊流送出資料量比較圖

■ 為未使用演算的資料丟棄率，✕ 為使用演算法的資料丟棄率，▲ 為可用頻寬限制參考線，橫軸為可用的頻寬 (KB/s)，縱軸為成功送至影音伺服器的資料量 (KB)

由圖 3-26 可看出使用演算法後，資料丟棄率大幅下滑，在 35KB/s 的限制頻寬下已經降為零，但是伴隨的代價可從圖 3-28 看出送出資料量亦大幅下滑。原因是本演算法

是調整攝影機的參數以減少產生的資料量，來降低視訊流的資料丟棄率，但也造成畫面品質變差或 FPS 降低，好處是資料在客戶端被「丟棄」的情況大幅降低，且可使緩衝區使用量大幅減少甚至一直維持在空的狀態，發佈端影音同步問題和客戶端間同步問題都得以改善。

資料丟棄率降低伴隨而來的好處是：影音伺服器儲存視訊流記錄時，檔案損壞減少，在重播串流檔案時，較不會有因檔案受損而出現畫面破格或是無法播出的情形。

3.4 畫筆標誌系統

畫筆標誌系統是本論文針對此互助平台而設計的一個新概念，不同於一般視訊會議的白板功能是針對靜態圖像，本論文開發的畫筆標誌系統是針對**動態的視訊畫面**做標誌，並且同一視訊連線活動所有使用者會看到相同的標誌。

3.4.1 設計方法

設計概念：

在視訊連線活動進行中，發佈端會看到自己攝影機擷取出的視訊同時將視訊發佈到影音伺服器，而其他使用者則是透過影音伺服器去接收發佈端的視訊，則所有進入此活動的使用者皆會看到相同的視訊畫面顯示在主畫面上，達到「視訊共有」的目的，為此設計畫筆標示系統的功能有：

- (1) 所有參與活動的成員皆可以在主畫面上做標誌，並且同時顯示在其他使用者的畫面上
- (2) 所有成員皆可以對其他成員的標誌進行個別物件拖曳或刪除。
- (3) 將所有標誌皆記錄下來，並在之後使用重現系統(本研究特別設計的)中播放出來。

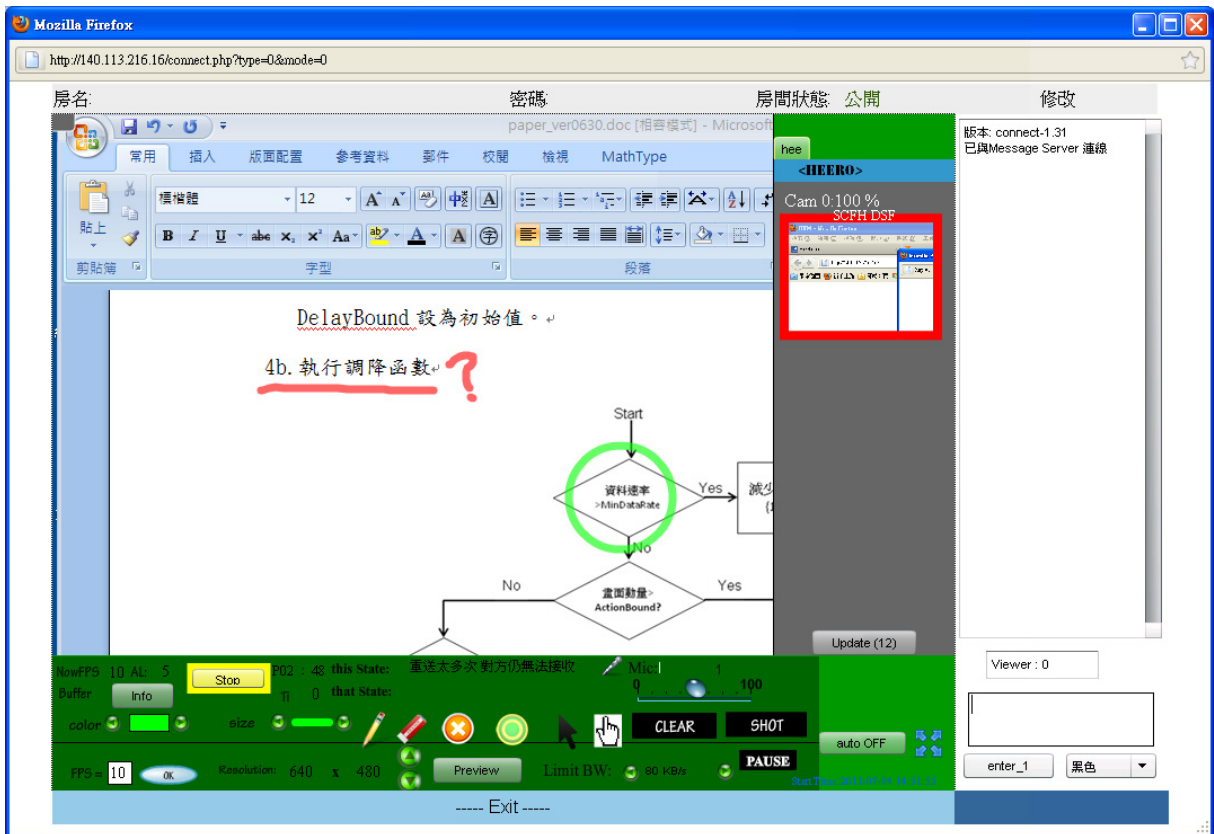


圖 3-29 畫筆標誌系統

成員使用者可在視訊畫面區做標誌傳給活動內所有使用者

設計方法：

畫筆標誌系統的操作介面主要有畫筆顏色、畫筆粗細、鉛筆類型、圓圈類型、箭頭類型、物件拖曳、物件刪除、全部清除等幾個按鈕，如圖 3-30，所有標誌的結果都能同步顯示在其他客戶端的畫面上。



圖 3-30 使用者操作畫筆的介面

物件化設計：

使用者在標誌時，拖曳滑鼠直到放開之後就會形成一個物件。如此，方便在多人連線時，各使用者都能對這些物件做操作，如拖曳、刪除等。

3.4.2 多客戶端間的畫筆同步

標誌訊息的同步機制是透過訊息傳遞伺服器做為傳送的管道。其傳送訊息內容主要有：標誌類型、畫筆樣式（顏色、粗細）、位置、編號。

結構圖：

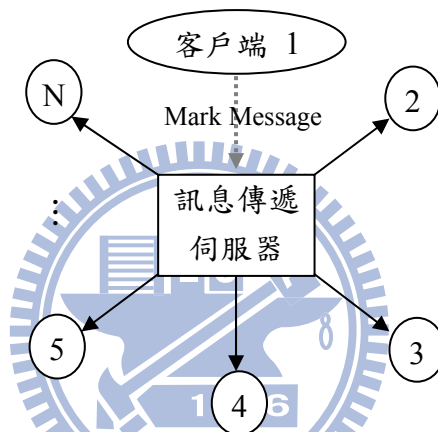


圖 3-31 標誌訊息傳送的架構圖

如圖 3-31，客戶端 1 在主畫面標誌放開滑鼠時，同時發送標誌訊息（Mark Message）送到訊息傳遞伺服器，由訊息傳遞伺服器轉送給其他客戶端。其他客戶端依收到的訊息內容，將標誌畫出。其他客戶端也皆可以發送相同的標誌訊息。

共用物件編號：

在同步控制上，為了能讓所有使用者皆能互相控制他人畫出的標誌物件，則這些客戶端必須要有共用的計數器去對這些在主畫面上的物件做編號。做法是，在發送標誌訊息的同時，也傳送該物件的編號，使得所有客戶端皆使用這唯一的編號將其資料存放在陣列中，如此可以多個客戶端皆可以操作拖曳或刪除物件。

標誌訊息格式：

標誌訊息必須盡量設計的精簡短小，主要原因有：

- (1) 在固定的字串長度下傳送更多內容

(2) 降低訊息接收時的解讀複雜度

(3) 減少訊息傳遞伺服器的負擔

訊息格式是由標誌類型加上參數組成：

「指令類型 參數 1, 參數 2...」

指令類型有：鉛筆、圓圈、箭頭、移動、刪除、清空，每個指令有的參數個數都不盡相同，如表 3-8。

表 3-8 各種畫筆標誌類型的標誌訊息結構

畫筆種類	訊息結構
 鉛筆	「起點 座標 x, 座標 y, 物件 ID」+「折點 座標 x, 座標 y, 物件 ID」
 鉛筆(長)	「起點 座標 x, 座標 y, 物件 ID」+「折點 座標 x, 座標 y, 物件 ID」 +「折點 座標 x, 座標 y, 物件 ID」+ ... +「折點 座標 x, 座標 y, 物件 ID」
 箭頭	「箭頭 座標 x, 座標 y, 向量 x, 向量 y, 物件 ID」
 圓	「圓圈 座標 x, 座標 y, 半徑 r, 物件 ID」
 刪除	「刪除 座標 x, 座標 y, 物件 ID」
 移動	「移動 座標 x, 座標 y, 物件 ID」

3.4.3 標誌自動清除

標誌自動清除功能是依據不同情境下的畫面動量變化而延伸的功能。

3.4.3.1 設計概念

當畫面有變動時，標誌往往會與畫面內容不符合而需要清除。本研究設計一個智慧型的自動清除功能，我們訂定一個清除門檻 (Clear Threshold)，當畫面動量高過清除

門檻，判定有畫面變動，此時會執行畫面清除；反之若低於門檻，則不做任何變更。而清除門檻的制定是依據現場環境背景擷取到的畫面動量來制定，若環境背景一直處於變動的情況下，如在播放影片的情況下，則清除門檻值就會相對較高，需要更高的畫面動量才能觸發畫面清除；反之，若背景環境大多處於靜止的情況，如在播放投影片，則清除門檻相對較低，畫面稍有變動就會高過此門檻而觸發畫面清除。

有執行標誌自動清除功能的只有擁有發佈影音串流權力的發佈端，每次觸發畫面清除皆會清除所有客戶端畫面上的所有標誌。若有客戶端間發佈權切換的情形，標誌自動清除功能會在新的發佈端運作。

3.4.3.2 清除門檻的制定

動態清除系統會依據當下擷取到的畫面動量去調整。

作法是每秒取一次目前的 *MotionLevel*，取五次平均加上一個容忍值做為一個清除門檻 (Clear Threshold)，其公式為：

$$\text{Clear threshold} = \text{Average MotionLevel} + \text{adjust value}。$$

Adjust value 是一個常數值，表示在一個 *Average MotionLevel* 之下，需要比 *Average MotionLevel* 在更高的 *MotionLevel* 才會觸發清除畫面動作。

舉個例子：

在畫面高動量時，比如在播放一步動作電影，得到的 *Average MotionLevel* 會提高，計算出來的清除門檻也會提高，所以儘管畫面一直在變動，標誌也不會被清除，只有在有更快速的動作使畫面產生更高的 *MotionLevel*，才會超過清除門檻而觸發清除畫面動作。

在畫面低動量時，比如在上課時播放投影片，*Average MotionLevel* 會較低，計算出的清除門檻也較低，在切換投影片時 *MotionLevel* 會瞬間提升超過清除門檻而觸發清除畫面動作。

3.5 互助歷程之記錄與重現系統

為了將所有互助合作過程的所有影像、記錄、標誌做為有用的資料能在之後觀看，我們將這些資料皆依時間儲存下來，並設計一套能同時處理這些資料並播放出來的系統。每個互助合作的儲存記錄皆可以納入平台的資料庫，供使用者調閱觀看，並將這些資料做有效的重複使用。將儲存記錄和重現播放分成兩個小節介紹。

3.5.1 記錄系統設計

設計方法：

一個視訊連線活動有多位成員使用者參與時，每個成員使用者皆會有自己的資料（串流、標誌、對話、擷圖）送到伺服器上，所以一個活動的記錄是多個使用者的綜合，記錄的方式是依照共同的伺服器時間順序排列，並記錄每個來源使用者的身分識別碼（ID）。

首先，本系統平台的網站、影音、訊息伺服器皆會統一對時，使用共用的標準時間，稱此時間為「系統標準時間」，所有記錄儲存時都是使用這個系統標準時間軸作記錄。

一個互助合作的過程會有四項資料需要記錄，每個資料的記錄方式皆不盡相同：

(1) 串流檔案（包含所有成員的影音串流、音訊串流）

當串流發佈到影音伺服器的同時，會將串流檔案存放在影音伺服器的硬碟中。所有客戶端會發佈一份自己專屬的音訊串流在伺服器存檔，而一個活動只會有一個共用的影音串流由發佈端發佈，若有發佈端切換，則取得發佈權的新發佈端會將發佈的影音串流重新串接到此共用的檔案。

(2) 畫筆標誌

發佈標誌訊息到訊息傳遞伺服器的同時，亦會在伺服器上存檔。每位成員使用者皆

能發送標誌訊息，但是在記錄儲存時皆存在同一個文件檔案中，依照時間順序排列。

(3) 文字對話

同畫筆標誌的做法，會在訊息傳遞伺服器上儲存在一個文件檔案中。

(4) 畫面擷圖

使用者在客戶端程式觸發擷圖動作，將擷取的內容送到網站伺服器由 PHP 寫入。各使用者擷取的圖片會存放在其網路空間中，能在之後瀏覽網站時看到這些擷圖，並能依自己喜好決定是否要公開這些擷圖給其他使用者觀看。

3.5.2 重現系統設計

3.5.2.1 設計概念與結構

為完整重現這多個記錄檔案，本論文設計一個能夠同時與網站伺服器、影音伺服器、訊息傳遞伺服器溝通取得記錄檔的記錄重現系統。

結構圖：

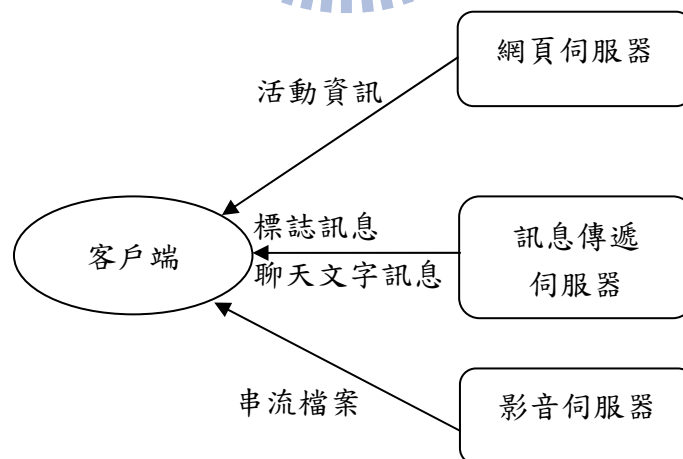


圖 3-32 與各伺服器溝通取得記錄的結構圖

客戶端播放程式會與網站伺服器中的資料庫取得一個活動的必要資訊，如識別活動名稱的 video code、活動參與成員的 id 列表、各記錄檔存放路徑等，播放程式取得這些資訊後會分別與訊息傳遞伺服器和影音伺服器連線。

與訊息傳遞伺服器連線取得活動的文字對話記錄、標誌記錄，將其載入客戶端的記憶體中，進行字串拆解和儲存在陣列中；與影音伺服器連線，接收主影音串流和所有成員的音訊串流進行播放。因為要同時播放不同伺服器上的檔案，或者同一伺服器中會有多個檔案要播放，所以我們需要處理多個記錄檔案播放的時間同步問題。

3.5.2.2 介面介紹

使用者可在本平台網頁進入重現系統的播放器，選擇想觀看的活動紀錄，即可開始播放，如圖 3-33，介面配置與網路上的播放器差異不大，上方區域為視訊播放區，下方橫條區域為播放控制列。

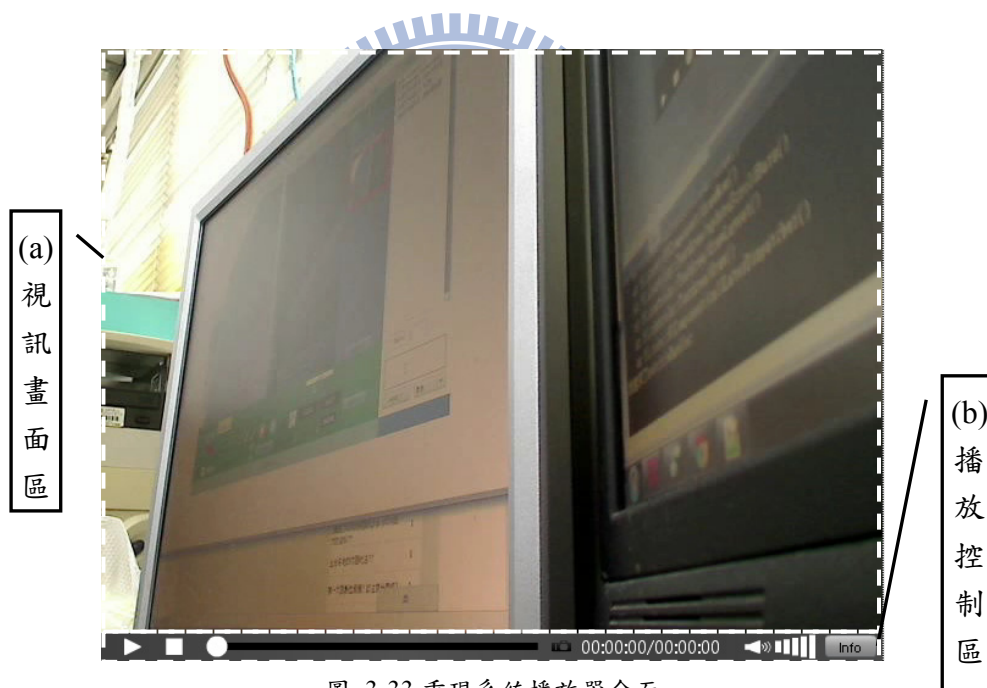


圖 3-33 重現系統播放器介面

若此活動記錄有畫筆標誌、文字對話記錄，則在播放過程中會依原活動記錄的時間、位置、內容準確的顯示在視訊畫面區，如下圖 3-34

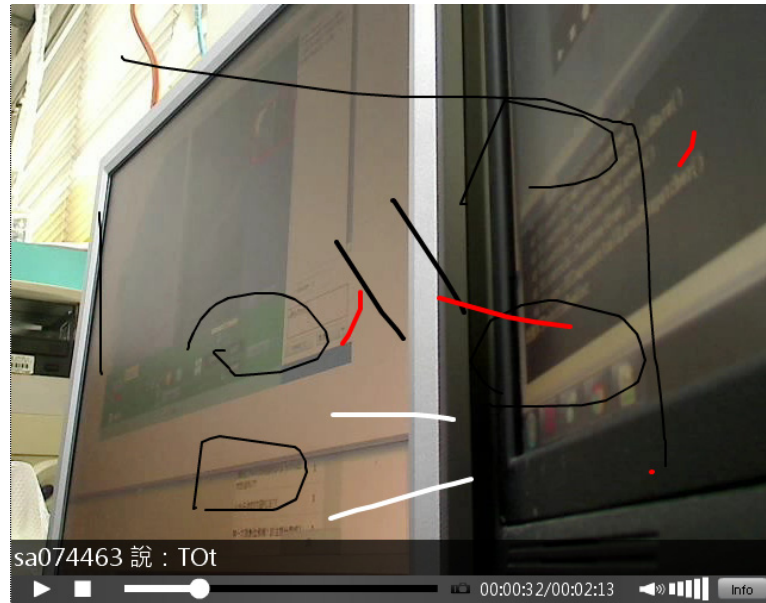


圖 3-34 播放畫筆記錄與文字對話的示意圖

3.5.3 重現系統之同步問題

3.5.3.1 系統標準時間軸

在多人視訊連線的時候，每個客戶端進入活動將音訊串流發送到伺服器上，皆會有一個獨立的存檔時間，畫筆標誌動作或文字對話亦會在訊息傳遞伺服器存檔並記錄時間。在重現播放的設計上，我們無法確保是先有串流檔案在影音伺服器上產生或是文字、標誌記錄在訊息傳遞伺服器上產生。所以在播放器啟動進行初始化載入資料時，需要比對所有存檔（包含串流、標誌訊息、文字對話）最早的時間記錄，做為此活動開始運作的「啟動時間」。

有了啟動時間，接著將各種記錄依時間順序在系統標準時間軸上開始播放，如下圖為例：

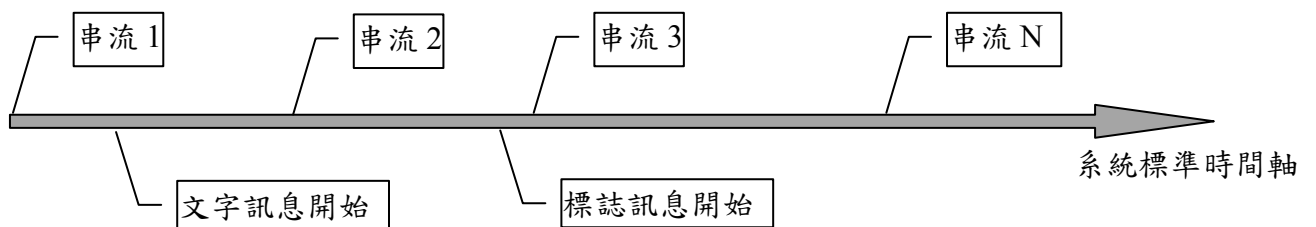


圖 3-35 播放各記錄時間示意圖圖

每個串流檔案的播放間隔會受到使用者成員加入活動和開始發佈串流的時間影響，所以播放時間間隔差異較大，而文字訊息和標誌訊息則是依活動內任一成員有在聊天室窗打

字或是在主畫面標誌的第一筆記錄為準開始播放。但只要確保所有儲存記錄皆是依照系統標準時間軸依序播放，就能解決同步問題。

3.5.3.2 多串流播放

在進行一個多人連線活動時，會在影音伺服器儲存一個影音串流和依照活動成員人數的多個音訊串流。之後要使用本平台的重現系統播放這些串流記錄時，需與影音伺服器同時連接這多個串流播放。

當同時播放串流數越多，會造成影音伺服器處理延遲時間更長，使得應同時播放的串流會因伺服器處理速度問題，各串流播放開始時間有些處差異，無法都對準系統標準時間軸，可參考第四章「4.5.1 多人連線記錄之同步播放測試」的測試結果。

解決此問題的方法是在多位使用者受邀進入一個活動時，盡可能避免同時發佈音訊串流，以錯開伺服器端音訊串流檔案儲存的建立時間。



第4章 系統測試與評估

第4章介紹第3章描述內容的相關測試，以佐證或驗證內容的正確性和合理性，使用的測試環境如下：

(1)網路環境：區域網路，使用 1Gbps 的網路頻寬。

(2)伺服器端硬體規格：如表 4-1

表 4-1 平台中的網站、訊息、影音伺服器硬體規格

Hardware Configuration	
Processors	Intel® Pentium® Processor E5300 @ 2.60 GHz (LGA 775)
Motherboards	ASUS P5KPL-CM (G31/ICH7) LGA 775
Memory	4 GB (2GBx2) Kingston KVR800D2N6/2G DDR2-800 6-6-6-18 @ 1.8 V
Graphics	Integrated Intel Graphics Media Accelerator (Intel® GMA 3100)
Network	Atheros AR8121/AR8113/AR8114 PCI-E Ethernet Controller (L1e)
Storage	Seagate Barracuda 7200.10 160 GB

(3)客戶端硬體規格：如表 4-2

表 4-2 測試用客戶端的硬體規格

Hardware Configuration	
Processors	Intel® Core™ i5 CPU 750@ 2.67 GHz 2.68GHz
Motherboards	ASUS P7P55D-E
Memory	DDR3 4GB RAM
Graphics	NVIDIA GeForce 9600 GT (600MHz,900MHz 256bit 1024MB)
Network	Realtek® 8112L Gigabit LAN controller
Storage	Hitachi 7200 1TB

4.1 影音伺服器負載測試

測試目的：

測試本系統平台的影音伺服器(Red5)單一串流被多客戶端接收時，在可提供順暢畫面給客戶端的情況下，最多可支持多少數量的客戶端。並藉由所記錄的網路流量、CPU使用率和記憶體使用量數據分析來找出瓶頸所在。

測試方法：

本測試為一個串流被多個客戶端所接收。使用的串流檔案資料速率為 400kbps 串流。定義一個客戶端進行連線活動時，會去接收此串流。

啟動 Smaxe 公司開發的「JUV RTMP LoadTester (Lite) 0.9」模擬軟體以設定不同之客戶端數量連上影音伺服器來增加伺服器的負載。接著增加一個新的**測試客戶端**去接收並播放此串流檔案。

每次測試會記下四項數值：

- (1) 伺服器的網路流量：測試時影音伺服器的輸出或輸入總流量。觀察伺服器所能提供的最大輸出及輸入流量。
- (2) CPU 使用率：用以評估是否因 CPU 能力不足，造成伺服器瓶頸，以至於無法處理更多使用者。
- (3) 記憶體使用率：用以評估是否因記憶體不足，造成伺服器瓶頸，以至於無法處理更多使用者。
- (4) 反應時間：當測試客戶端要求接收串流動作時記下時間 T1，伺服器接收到客戶端的要求，將串流傳送至此客戶端，客戶端接收到串流時記下時間 T2。若忽略網路傳輸的時間，T2 減 T1 的時間差即為伺服器的反應時間。

其中數據(1)、(2)、(3)為評量伺服器瓶頸的要素，數據(4)作為 4.2.1 節訊息傳遞發佈與接收反應測試中，做為本平台訂定使用者可忍受延遲時間的參考資料。



隨後每次測試會逐漸提高設定的**客戶端數量**。藉由提高接收客戶端總數，觀察伺服器服務能力上限及負載瓶頸所在。

測試結果與討論：

下列四圖為測試結果的四項數據，橫軸皆為客戶端數量，縱軸依序為(1)伺服器輸出流量 (MB/s)、(2)CPU 使用率(%)、(3)記憶體使用率(%)、(4)伺服器反應時間 (ms)。

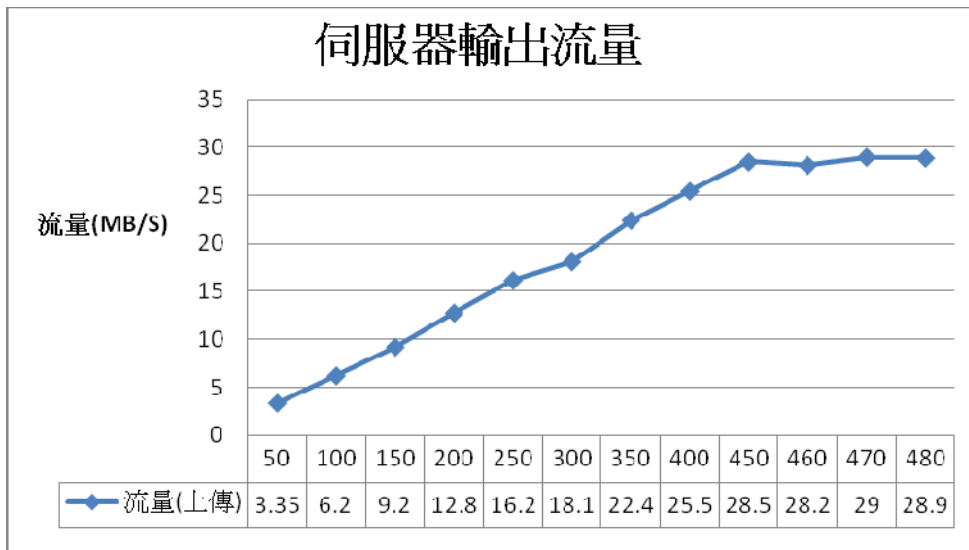


圖 4-1 影音伺服器負載測試之伺服器輸出流量圖

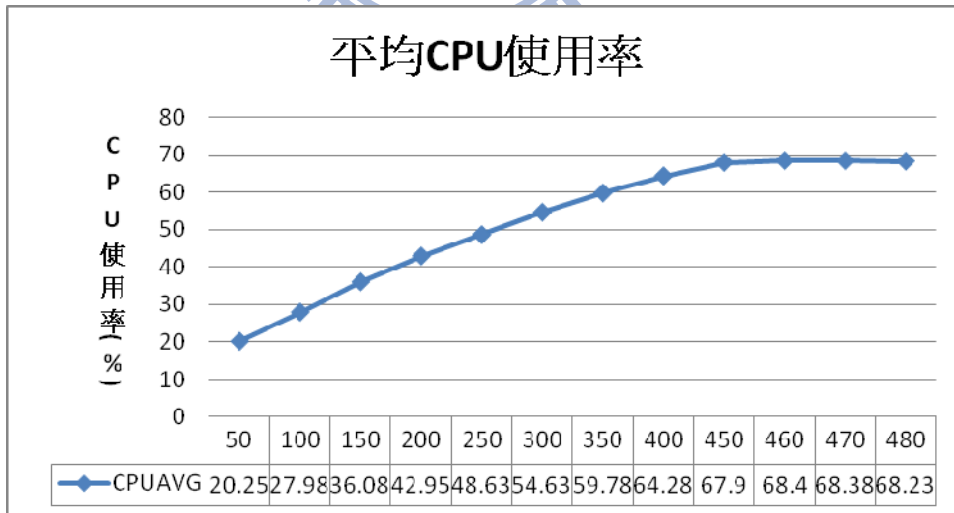


圖 4-2 伺服器平均 CPU 使用率圖

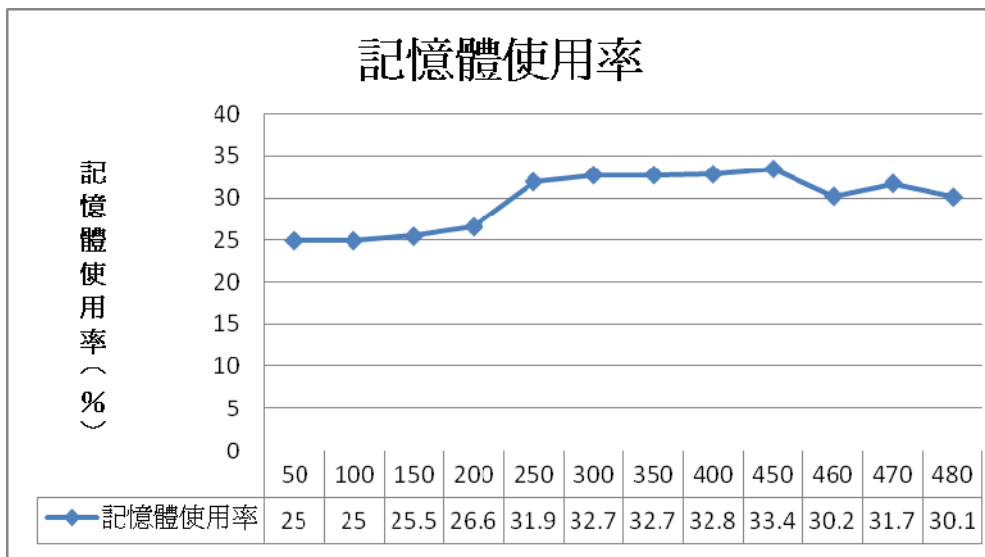


圖 4-3 伺服器平均記憶體使用率圖，總量為 3.7GB

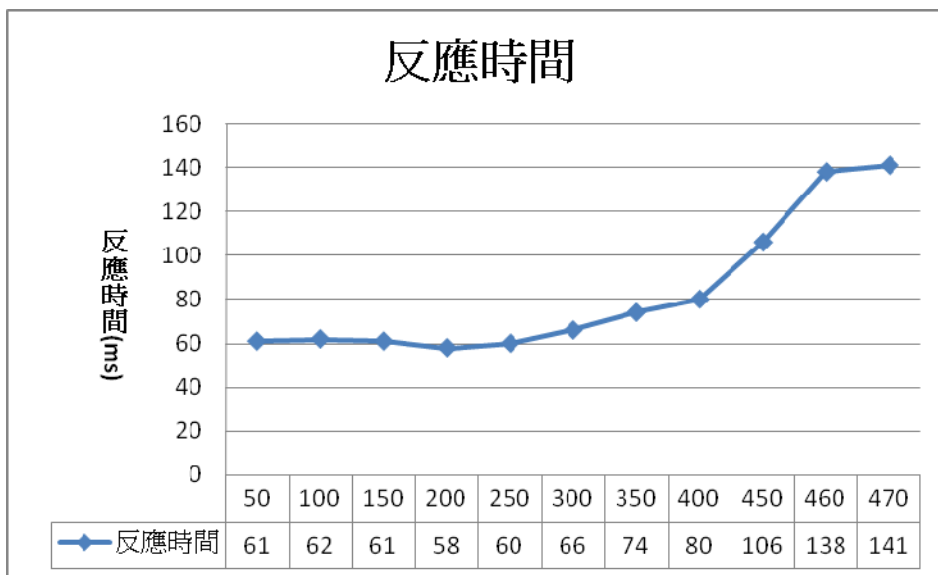


圖 4-4 伺服器處理客戶端接收串流之反應時間

伺服器輸出流量、CPU 使用率隨著客戶端數量增加成線性成長，直到接近 450 到 480 之間時趨於平緩，且當測試數量為 480 時，使用測試客戶端連接影音伺服器已毫無反應，不被伺服器所接收，顯示本平台可連接的客戶端數量約在 470 到 480 之間。因使用的網路頻寬比流量還大，再觀察的 CPU 使用率，發現無明顯成長，可以推測伺服器瓶頸為 CPU。

因伺服器反應時間由圖 4-4 觀察出：在伺服器接近滿載時，即客戶端數量是 450 以上時，反應時間有明顯增加；而記憶體使用率由圖 4-3 得知記憶體使用率僅從 25% 小幅成長到 30% 左右，無明顯改變。

另外，觀察測試客戶端播放的視訊畫面，即使到 470 個客戶端數量，依舊不會出現任何停頓情形，可判斷本論文的影音伺服器 Red5，高負載並不會影響已經連線的客戶端。當超過伺服器負載時，伺服器不會再接受客戶端連線。

4.2 多人連線結構之效能測試

4.2.1 客戶端間訊息傳遞與接收串流反應時間測試

測試目的：

當一個客戶端取得發佈權時成為新發佈端時，需要通知原發佈端接收串流，這個過程會受到影音伺服器和訊息傳遞伺服器負載不同而有延遲時間。測試兩個伺服器在不同人數負載情況下，通知串流接收到接收完成的延遲時間。

測試方法：

訊息傳遞路徑為：發佈端 Flash A 向訊息傳遞伺服器發送訊息，訊息內容為：“要求客戶端向影音伺服器接收串流”，訊息傳遞伺服器轉送訊息給接收端 Flash B，接收端收到訊息後向影音伺服器送出接收串流的要求，取得影音串流。如圖 4-5 所示應用實例：發佈端傳送訊息要求新加入活動的使用者接收影音串流。

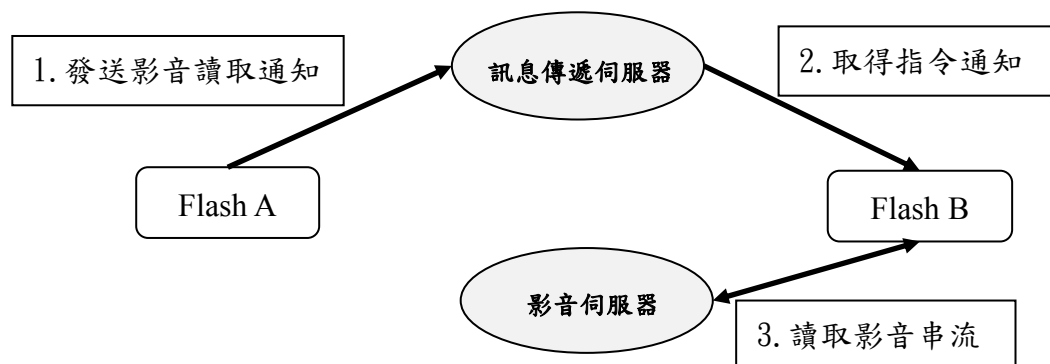


圖 4-5 客戶端 Flash 程式要求其他客戶端接收影音串流的通知路徑

本測試使用如同 4.1 節的測試軟體 JUV RTMP LoadTester (Lite) 0.9 作為模擬大量使用者，先對影音伺服器進行預先設定數量客戶端接收串流連線，以模擬相同數量的使用者對影音伺服器造成的 CPU 使用率、記憶體使用、硬碟效能與網路等之負載情形。另外以平台已完成之「訊息傳遞伺服器測試軟體」[5] 模擬使用者對伺服器發送訊息的行為，使用同樣的設定使用者人數對伺服器增加負載。

在訊息傳遞伺服器與影音伺服器皆有負載的情況下，測試客戶端 Flash A 持續發佈串流資料到影音伺服器，串流速率為 400Kbps，此為本平台一般使用者採用之資料速率。

在不同的使用者人數設定下，測試客戶端 Flash B 新加入視訊連線活動，Flash A 會收到 Flash B 加入活動的通知訊息，此時 Flash A 會發送一則訊息給訊息傳遞伺服器轉送至 Flash B，訊息內容為：“要求客戶端向影音伺服器接收串流”，發送訊息的同時會在 Flash A 記下時間 $T1$ 。Flash B 收到訊息傳遞伺服器轉送來的訊息時記下時間 $T2$ ，Flash B 收到訊息加以解讀、解讀完成到執行訊息內容完成時記下 $T3$ ，最後 Flash B 會向影音伺服器要求 Flash A 發佈之串流。Flash B 成功接收串流時，記下時間 $T4$ 。

測試結果與討論：

測試數據與使用者人數之關係如圖 4-7

$T2 - T1$ 的時間差為 Flash A 發出訊息，到 Flash B 收到此訊息所耗費的時間。此值僅與訊息傳遞伺服器轉送訊息的處理時間有關

$T3 - T2$ 的時間差為 Flash B 解讀訊息到執行完成的時間，此值與 Flash B 硬體能力有關，可視為一個常數值。

$T4 - T3$ 的時間差為影音伺服器收到客戶端串流接收要求至客戶端接收到串流的時間。

$T4 - T1$ 的時間差為一個新的使用者進入活動時，由發佈端通知此新使用者接收串流到此使用者實際收到串流的時間，恰好為前三個實驗數據相加。

圖 4-6 為使用真實電腦配備進行本測試之測試結果。因硬體配備與人力有限，更進一步以軟體模擬本測試的使用者人數。

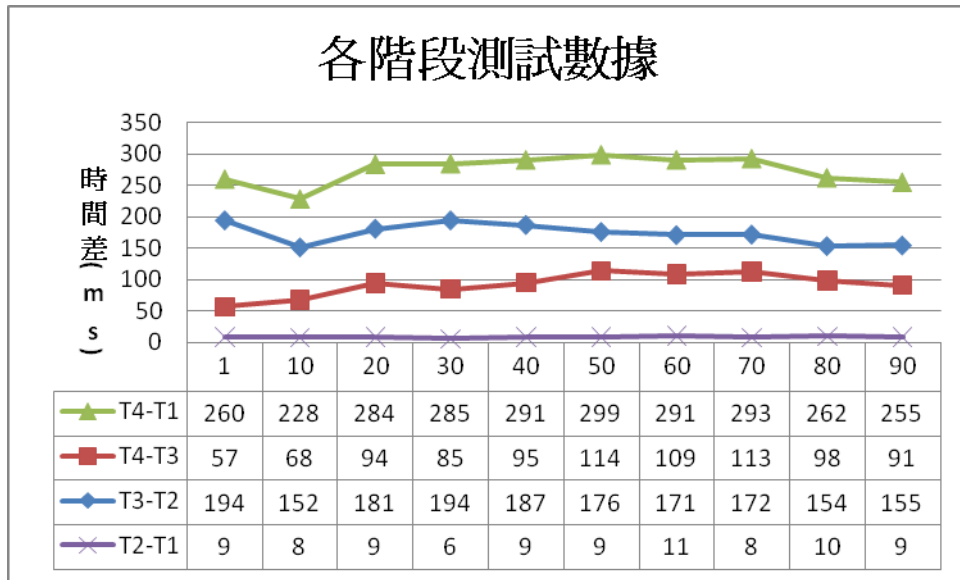


圖 4-6 使用實機之測試結果

橫軸為影音伺服器及訊息傳遞伺服器負載的使用者人數，縱軸為各階段測試時間差。

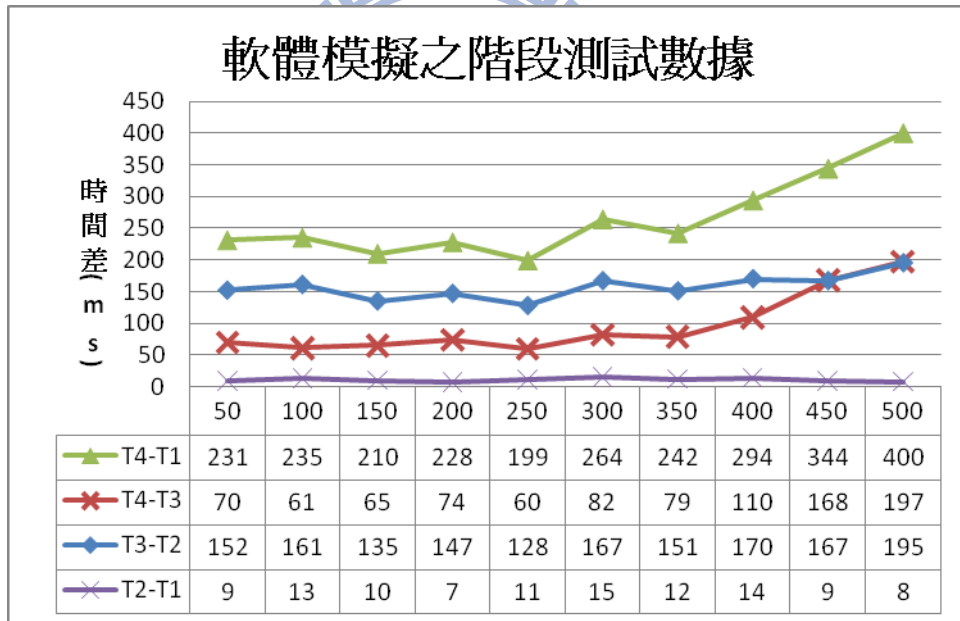


圖 4-7 改以軟體模擬使用者之測試結果

橫軸為影音伺服器及訊息傳遞伺服器負載的使用者人數，縱軸為各階段測試時間差。

當設定的使用者人數到達 450 人時， $T4-T3$ 開始有明顯的增加，而 $T3-T1$ 依舊無太大起伏，再參考 4.1 影音伺服器的負載測試結果可得知，450 人以上已經接近影音伺服器的滿載人數，則整個訊息傳遞流程的時間差 $T4-T1$ 增加，主要是因 $T4-T3$ 影音伺服器的負載接近上限而回應遲緩影響。

本次測試得知在影音伺服器及訊息傳遞伺服器同樣服務相同人數時，會先受限影音

伺服器的負載能力而無法再繼續增加服務人數。

4.2.2 多人連線發佈與接收串流之效能測試

測試目的：

在一個多人視訊連線活動中，發佈端發佈串流給活動內所有成員使用者，測試所有成員成功接收串流的延遲時間與活動內成員人數的影響。

測試方法：

一位成員使用者邀請 N 位其他使用者加入多人視訊連線活動。活動建立並且其他使用者加入完成後，使用者 A 擁有發佈權成為成員發佈端，其他使用者為成員接收端。發佈端發佈一個 400Kbps 串流到影音伺服器，同時記下時間 T_{start} ，各接收端成功接收串流各自記下時間 T_{R1} 、 T_{R2} 、 \dots 、 T_{RN} ，則最大發佈與接收的延遲時間差為 $Max \{T_{Ri} - T_{start}\}$ ， $i = 1, 2, \dots, N$ 。

測試數據：

圖 4-8 為一次發佈串流給多位使用者時，比較發佈端和所有接收端的時間差取最大者，越高代表該使用者越晚才會接收到串流。

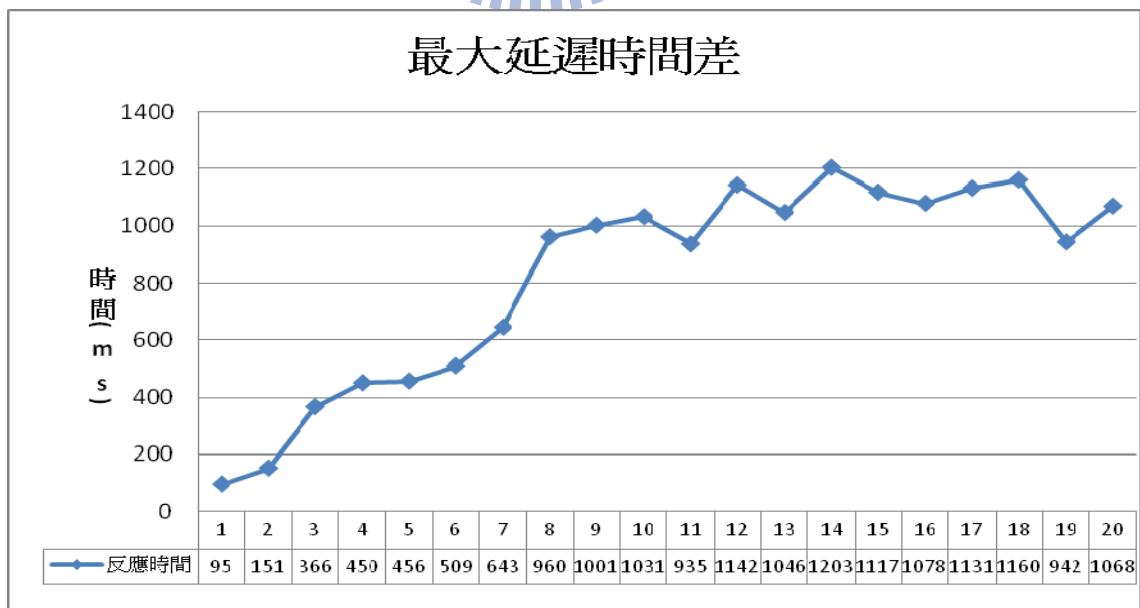


圖 4-8 發佈開始到所有接收端完成接收之接收端最大延遲時間差圖
橫軸活動中成員接收端的數量，縱軸為延遲時間差 (ms)

因硬體資源與人力限制，僅測試 20 位使用者，如圖 4-8 所示，最大延遲時間差隨著客戶端數量增加，直到十人之後維持在一秒左右，接著最大反應時間不再有明顯成長。

此一測試結果說明發佈與接收之間或多或少會有時間差，此為 3.2.3.2 小節描述的客户端間同步問題的影響元素之一，時間差受到網路環境、客戶端硬體設備、伺服器處理時間、串流本身的資料速率等許多可控制和不可控制的因素影響。本論文就針對緩衝區使用量，盡量減少產生客戶端間同步問題的可控制因素。

4.3 多人影音同步問題

4.3.1 影音分流或合送之影格丟失測試

測試目的：

探討多客戶端之間在傳輸串流，當網路環境變更造成頻寬不足時，比較串流影音合併與分流的丟棄影格數。

測試方法：

以頻寬控制軟體 NetLimiter 限制目前可用的頻寬，一個發佈端使用網路攝影機以 FPS10 的影格速率擷取畫面成為串流，不使用緩衝區，固定發佈的視訊流為 400Kbps 和音訊流為 44Kbps，持續發佈 60 秒鐘到影音伺服器。討論不同的串流傳送模式，在 60 秒內串流丟棄影格數。記錄以下二項數據：(1)串流合送時，發佈串流的丟棄影格數、(2)串流分送時，發佈視訊流的丟棄影格數，接著再變更 NetLimiter 限制的可用頻寬，觀察丟棄影格數的變化。

測試結果討論：

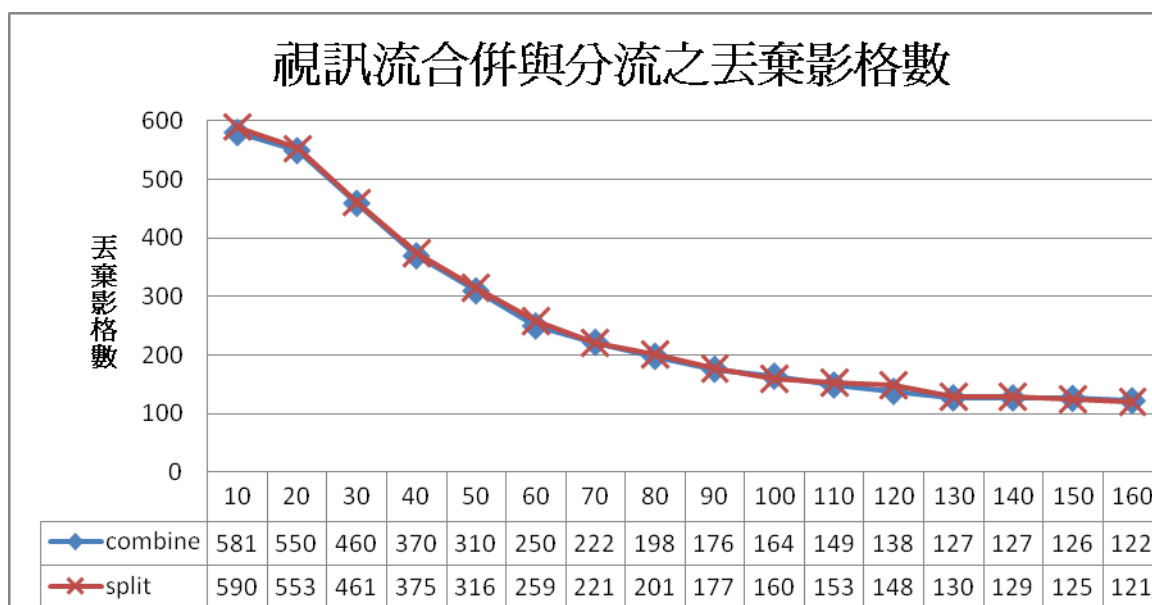


圖 4-9 測試 4.3.1 的結果：視訊流合送與分流之丟棄影格數

縱軸為 60 秒內丟棄影格總數，橫軸為限制可用頻寬 (KB/s)。表格中數據為發佈 400kbps 影音串流時，combine 為合併傳送、split 為分流傳送的視訊流丟棄影格數

控制頻寬限制為變因，測試不同傳輸結構的視訊流丟棄影格數，直到控制頻寬限制無法造成丟棄影格數明顯變化為止。如圖，可發現在限制頻寬為 130 到 160KB/s 時，丟棄影格數已無明顯變化，顯示可用頻寬增加，在無使用緩衝區的情況下還是會丟棄視訊流的影格。

另外，從圖 4-9 的合併與分流的丟棄影格數測試結果得知，兩者的趨勢線幾乎沒有差別，顯示串流合送與分流並不會影響丟棄影格數。

4.3.2 不同頻寬限制下的串流合併或分送資料丟棄率測試

測試目的：

在不同的頻寬限制下，比較影音串流合併和分送時，因產生資料量無法被頻寬所消化的資料丟棄率。

測試方法：

以頻寬控制軟體 NetLimiter 限制目前可用的頻寬，一發佈端發佈 400Kbps 的視訊

流和 44Kbps 的音訊流持續 60 秒鐘到影音伺服器，串流使用足夠大的緩衝區，排除因為緩衝區被填滿而丟棄資料的可能。每次測試控制不同的可用頻寬，記錄發佈 60 秒後，分別記錄視訊流、音訊流各自產生的資料量 D_{CRE} (Creative Data) 和緩衝區使用量 B_{NOW} ，相減之後來計算實際從發佈端送出到伺服器端的資料 (觀察伺服器實際儲存的串流檔案大小，恰好與相減之後的結果相近)。

發送至影音伺服器的資料量為

$$D_{PUB} = D_{CRE} - B_{NOW} \text{ (Bytes)}$$

留存在緩衝區未送出的資料 B_{NOW} ，對照在未使用緩衝時可將 B_{NOW} 視為丟棄，藉此計算資料丟棄量 D_{LOS} ：

$$D_{LOS} = D_{CRE} - D_{PUB} = B_{NOW} \text{ (Bytes)}$$

資料丟棄率為：

$$R_{LD} = D_{LOS} / D_{CRE} \text{ (\%)}$$

每次測試分別記錄下列四種資料送出丟棄率 R_{LD} ：(1)合併時視訊流、(2) 合併時音訊流、(3)分送時視訊流、(4)分送時音訊流。另外也記下：(5)合併時資料發送量、(6)分流時資料發送量，做為 3.3.1 小節串流資料量調控演算法使用的數據。

測試結果討論：

將測試數據繪製成以下三圖，橫軸皆為橫軸為頻寬限制(KB/s)，縱軸分別為表示串流合併與分流傳送的視訊流資料丟棄率(%)、音訊流資料丟棄率(%)、資料發佈量(KB)。

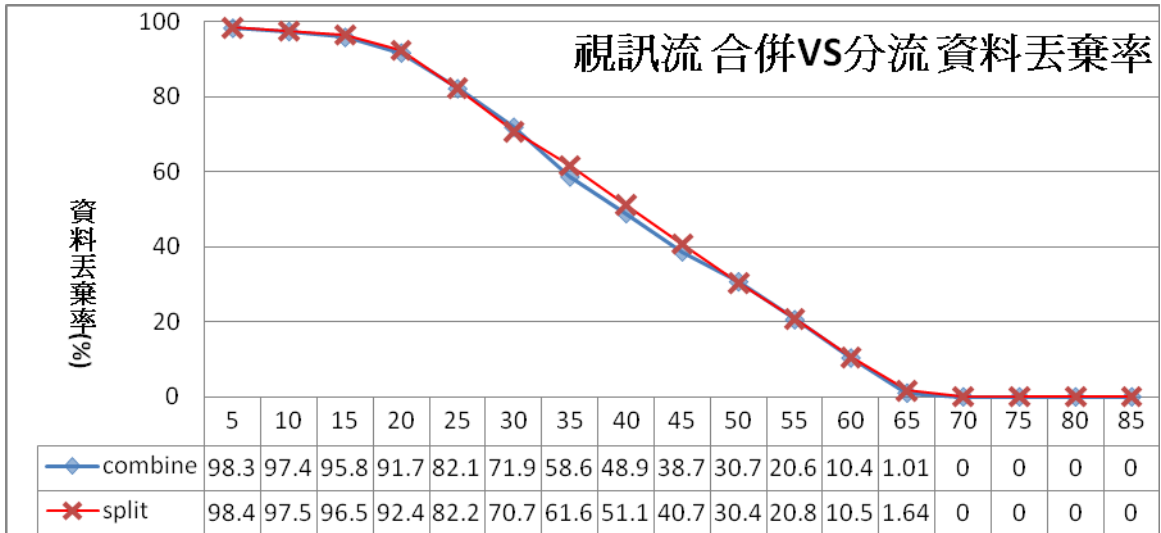


圖 4-10 測試 4.3.2 結果：視訊流合併和分流之資料丟棄率比較圖

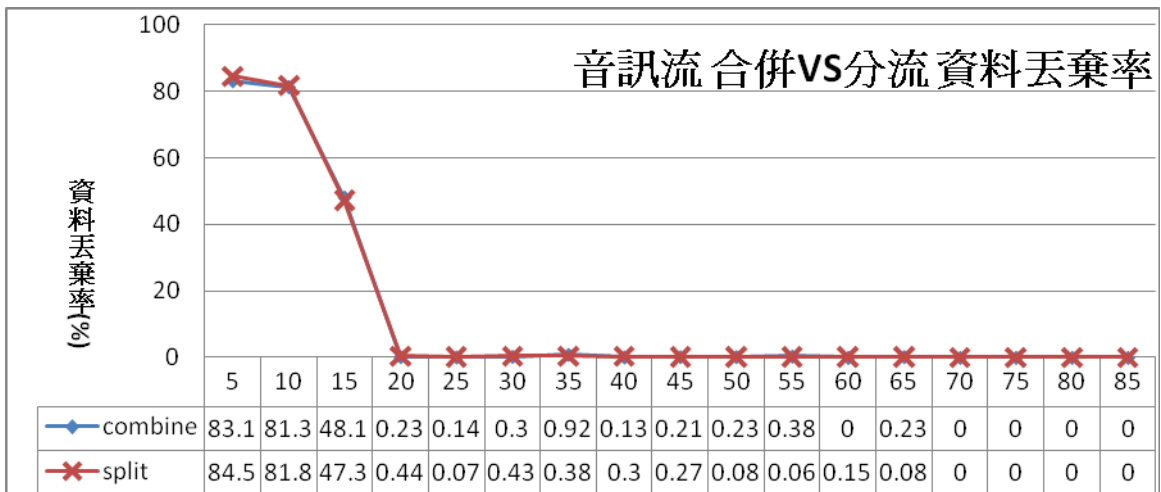


圖 4-11 測試 4.3.2 結果：音訊流合併和分流之資料丟棄率比較圖

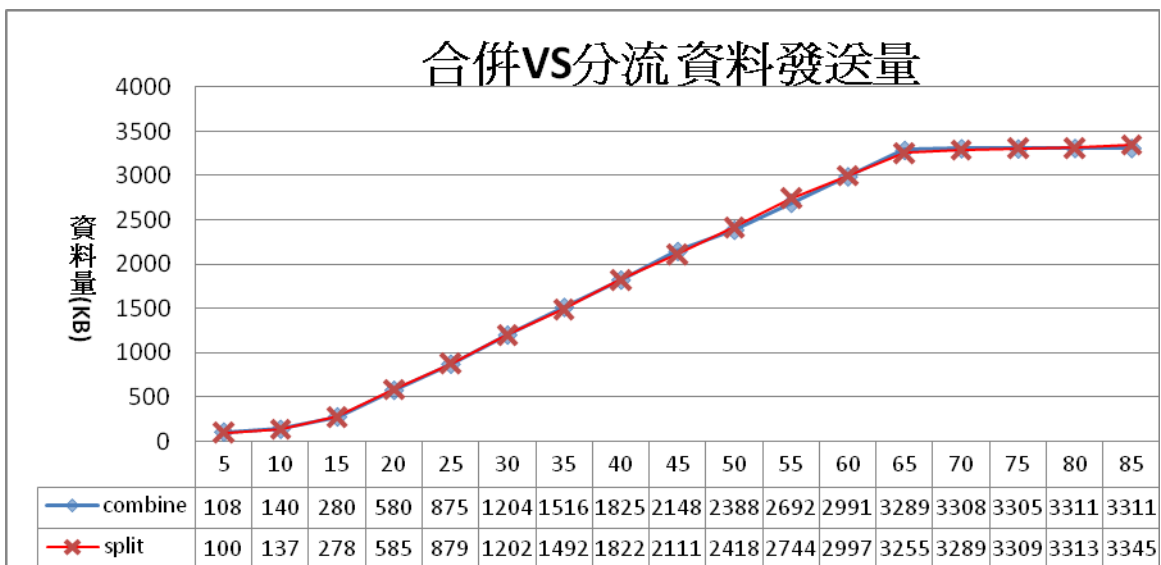


圖 4-12 測試 4.3.2 結果：串流合併和分流之資料發送量比較圖

由上述三圖可知，視訊和音訊流合併、分流的資料丟棄率相近，發送到伺服器的資料量趨勢線幾乎是穩合。顯然地，串流傳輸結構不是影響視訊、音訊流資料丟棄率的主因。

資料丟棄率顯然受到可用頻寬的影響，觀察圖 4-10 的視訊流資料丟棄率下降趨勢，以頻寬限制 20KB/s 為分界線到 65KB/s 為止，此段視訊流頻寬限制與資料丟棄率的關係線幾乎是線性關係，並且可以推得預測趨勢線方程式為：

$$y = -10.137x + 101.95 \quad (4.1)$$

可用頻寬超過 65KB/s 後，資料丟棄率已幾乎為零，頻寬足夠使用。可用頻寬低於 20KB/s，視訊流的丟棄率無法呈現線性，直到音訊流資料丟棄率為零（以本測試為例是在可用頻寬 20KB/s 之後），視訊流的資料丟棄率才與可用頻寬呈現線性關係。這個測試結果可得知傳送音訊流會優先於視訊流，音訊流較不會受到頻寬不足而被丟棄資料，以保持較完整的語音通話內容，滿足本平台期望能維持視訊連線活動品質的考量。

4.3.3 畫筆標誌透過訊息傳遞伺服器傳送的基本完成時間測試

測試目的：

定義「遠端畫筆標誌基本完成時間」為：在兩個客戶端情況下，一客戶端送出畫筆訊息到另一客戶端接收並且繪製完成的時間。以此了解在使用者與人互動時，雙方看到相同的畫筆標誌出現在視訊畫面區時，所需的最少延遲時間。







測試方法：

建立一個雙人視訊連線活動，一個客戶端 A 透過訊息傳遞伺服器將畫筆標誌訊息傳的給客戶端 B。測試本平台所有的畫筆標誌類型，客戶端 A 送出標誌訊息時記下時間 $T41$ ，客戶端 B 收到標誌訊息開始解析處理直到標誌在主畫面完成記下時間 $T42$ ，觀察各種畫筆標誌類型在客戶端間傳遞標誌的基本完成時間 ($T42-T41$)。

測試結果與討論：

表 4-3 為各種畫筆標誌類型進行 20 次測試所得的平均完成時間

表 4-3 客戶單間標誌訊息送出到另一客戶端標誌完成的基本完成時間
鉛筆僅畫單一直線，鉛筆(長)指的是畫到系統可用的最大訊息長度 1400 字元

畫筆種類	完成時間 T42-T41 (ms)
 鉛筆	156.5
 鉛筆(長)	909.9
 箭頭	64
 圓	29.2
 刪除	30.8
 移動	37.3

將本次測試結果與表 3-8 標誌訊息結構對照比較，鉛筆標誌訊息是由一個起點加上多個折點串接而成，故折點越多，鉛筆標誌訊息就會越長，而除了鉛筆以外的標誌訊息僅需一行標誌訊息即可完成。顯然地，畫筆標誌基本完成時間受到不同類型的標誌訊息長度影響。

4.4 記錄與重現系統測試

4.4.1 多人連線記錄之同步播放測試

測試目的：

測試多人連線的串流錄影，重現記錄時需要同時播放多個串流，觀察播放時的延遲時間。

測試流程：

設定不同的活動成員數，活動內的發佈端發佈影音串流，其他成員發佈音訊串流，

共儲存了一個影音串流和不同數量的音訊串流數。使用本系統的記錄重現播放器為客戶端，與影音伺服器連線，連結這些串流播放。在客戶端發出播放指令時，記下時間 $T51$ ，所有串流皆向影音伺服器要求連結成功並且開始播放時，記下 $T52$ 。每次測試皆在相同的網路環境下，且確定尚未達到網路頻寬限制，觀察 $T52$ 和 $T51$ 的時間差，忽略網路傳輸時間，即為影音伺服器對多串流同時播放的反應時間。

測試結果討論：

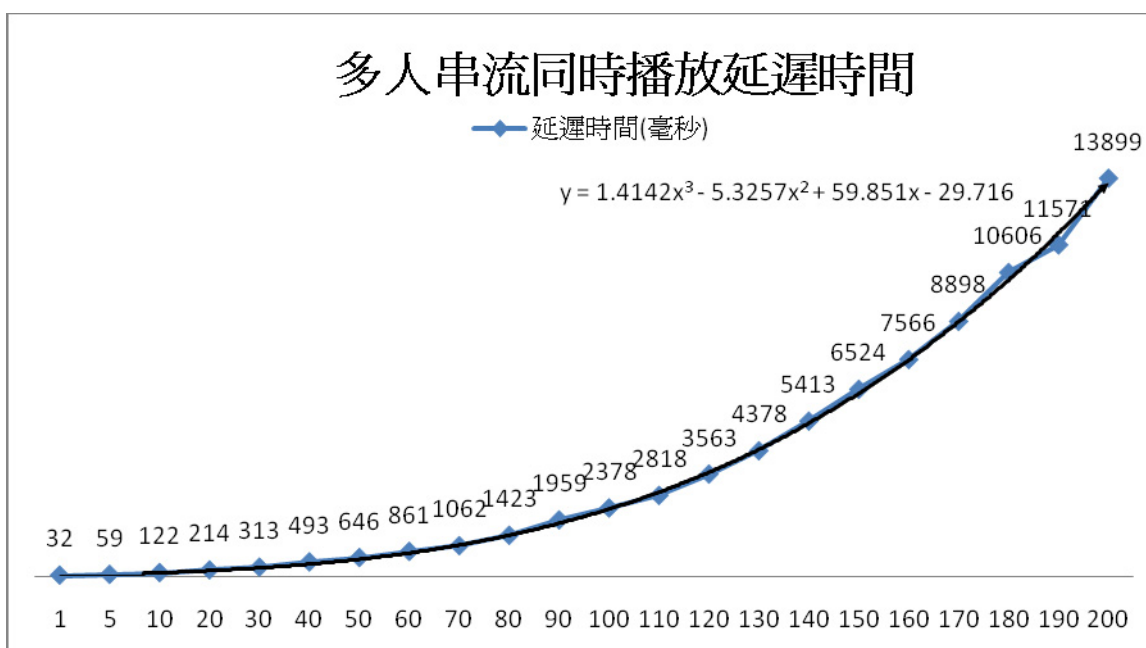


圖 4-13 多人串流同時播放延遲時間趨勢圖

橫軸為同播放音訊串流數，縱軸為延遲時間 ($T52 - T51$ ，單位是毫秒)，每個點上方的數字為測試數據。如圖 4-13，伺服器反應時間會隨著同時播放串流數增加，經由曲線迴歸預測分析得此曲線近似一個三次函數曲線。令 x 為同時播放串流數， y 為延遲時間，得到一個 y 對 x 的三次函數為：

$$y = 1.4142x^3 - 5.3257x^2 + 59.851x - 29.716 \quad (4.2)$$

當同時播放串流數超過一百時，延遲時間已經超過兩秒，即同一時間錄製的音訊串流在播出時，最早和最晚播出的音訊串流相差超過兩秒，使用者聽到此二音訊串流的對話已經對不起來了，會有強烈的不協調感。此結果也說明多串流同時播放會造成重現系統的同步問題。解決方法是：應避免多個成員客戶端同時發佈音訊串流到影音伺服器上儲存。

第5章 結論與建議

5.1 結論

本論文提出的多人畫筆標誌系統是一個協助遠端互動的新概念，雖功能簡單，但是市面上對於動態視訊的標誌功能還不多見，讓使用者能更容易描述問題和現場狀況。畫面動量、緩衝區使用量的偵測與應用也是本論文一個評估現場環境的嶄新要素。以畫面動量衡量互助活動使用者的行為動靜狀態，以緩衝區使用量衡量使用者的硬體能力、網路頻寬，加強在串流資料量產生的控制，使得客戶端程式更能輔助使用者使用其網路攝影機和可用的頻寬。

透過本論文研究之客戶端程式輔助功能，讓使用者能夠克服各種網路環境和提高資料的完整性，達成隨時隨地進入雲端服務視訊互助平台獲得協助的目的。



5.2 未來展望

5.2.1 公式化的進階串流資料量調控演算法

目前 3.3 節提到的串流資料量調控演算法還可以更進一步的強化效率。調控參數以改變資料量的速度不是理想，目前作法是以「逐漸」調高或調低來觸碰門檻值，這過程就已經先丟棄許多資料了。因不同使用者的電腦硬體配備和網路環境皆不盡相同，目前尚缺乏一個合適的公式直接計算出門檻值，使得演算法在短時間將參數調到門檻值。能在短時間偵測到的資訊如 $C.FPS$ 、 B_{now} 等，經公式計算得到一個門檻值是未來改進的方向。

5.2.2 視訊連線客戶端應用於行動裝置平台

現今社會手機普及率已遠高於個人電腦、筆電，智慧型手機也大多具備攝影機和麥克風的功能，符合本平台進行視訊連線活動所需的基本輸入裝置。若能將本平台功能設計一

個專屬應用程式安裝在手機上，搭配手機無線上網功能，更可以隨時隨地取得現場環境的問題且能進入平台尋求協助。但將會面臨的問題是，需要有更佳的串流資料量控制演算法以符合嚴苛的無線網路環境，手機 CPU 對即時視訊壓縮的運算能力會是最先遇到的問題。

5.2.3 標誌智慧型追蹤系統

本論文 3.4 節提到的畫筆標誌系統可以再進一步強化，不僅僅是使用者主動進行標誌，更可以加入物件辨識功能，以系統 AI 自動判斷來追蹤在變動中視訊畫面的物件。

5.2.4 記錄重現系統進階功能

本論文 3.5 節的紀錄重現播放器，目前僅有播放功能，未來可以結合畫筆標誌系統，在播放器播放影像時也讓使用者可進行畫筆標誌加以編輯、加上文字註解等，使得每位使用者可以對一個活動記錄有個人化的畫筆標誌，強化對互助記錄的描述，豐富本平台的互助活動記錄資料庫。



第6章 參考文獻

[1] 串流媒體，Wiki，

<http://zh.wikipedia.org/wiki/%E6%B5%81%E5%AA%92%E4%BD%93>

[2] RTMP 介紹，wiki，

http://en.wikipedia.org/wiki/Real_Time_Messaging_Protocol

[3] 視訊會議介紹，Wiki，

<http://zh.wikipedia.org/wiki/%E8%A6%96%E8%A8%8A%E6%9C%83%E8%AD%B0>

[4] 戴永彰，「雲端服務視訊互助平台之網站架構研究」，新竹交通大學，碩士論文，2011 年。

[5] 陳冠宏，「雲端服務視訊互助平台之訊息傳遞伺服器研究」，新竹交通大學，碩士論文，2011 年。

[6] 孫超，「流媒體服務器 Red5 的擴展設計、測試與優化」，上海交通大學，碩士論文，2009 年。



附件一：串流資料量調控演算法完整流程圖

