

附錄 A、專題研究計畫書

基於 SIP 的對等式網路架構即時視訊群播串流服務

A SIP-Based P2P Network Architecture For Real-Time Multimedia Streaming Service

專題學生：林育慈

指導老師：陳旻秀

(一)摘要

^[1]串流服務是利用網路分段傳送的方式，將經過壓縮的多媒體資料像流水般傳送，達到不需將整個檔案下載下來即可播放的目的，更而達到提供現場直播的服務。而在手持行動裝置與行動上網服務的迅速發展之下，將串流服務與行動應用軟體搭配，更可讓使用者隨時隨地針對任何狀況或景象與朋友即時分享畫面與資料。

然而大量運用即時轉播服務卻會產生大量的資料傳輸，會對轉播端的網路頻寬產生壅塞的影響，雖可藉由群播（multicasting）技術方式改善轉播端的負載，但卻會造成伺服器端大量的重複資料傳輸。為了解決這個問題，本計畫將以對等式（Peer-to-Peer，P2P）網路架構為主，搭配並修改 SIP 的協定，開發一套提供使用者方便使用且具有跨平台執行能力的即時視訊串流群播軟體。

(二)研究動機與研究問題

網路視訊服務係為現今相當蓬勃發展的一項服務。常見的隨選視訊（Video on Demand）服務——以 Youtube 為例，不單只是一般使用者會將其拍攝或喜歡的影片放上網路視訊平台與朋友分享，亦有媒體相關產業將這些網路視訊平台視為一個非常重要的宣傳媒介，更有使用者或團體藉由網路視訊平台打響知名度，作為其事業開端。除隨選視訊服務外，即時視訊（Real Time Video）串流服務的運用也相當蓬勃。^[1]串流技術能將一連串的媒體資料壓縮後，利用網路分段傳送，達到即時的現場直播服務。而即時串流服務搭配網路群播（multicasting）服務，可成為一簡易轉播平台，讓更多人取得一般傳播媒體無法取得之訊息，且更快掌握最新且最即時的資訊。尤其手持行動裝置的普遍運用，搭配行動上網的頻寬與穩定性的提昇，使得網路視訊串流服務能與行動裝置結合，讓社會大眾隨時隨地針對任何狀況或景象與朋友即時分享畫面與資料。在近年來國內外許多社會運動皆可看到此類即時群播視訊串流服務，如國內的太陽花運動在立法院內的即時轉播使得社會大眾對於活動關注有著推波助瀾的效果。

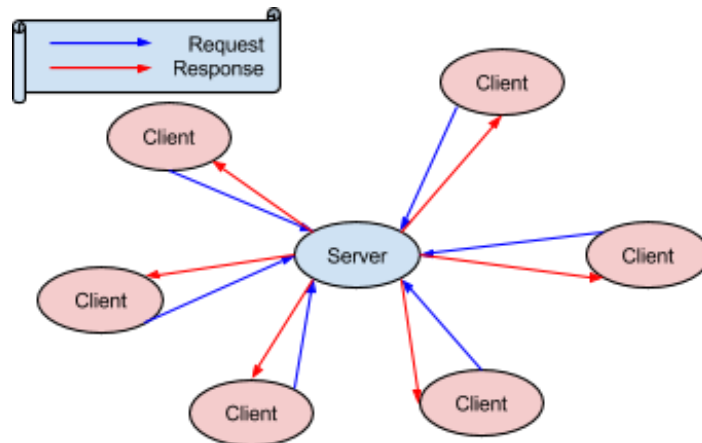
然而大量運用即時轉播服務所產生大量的資料傳輸，會對轉播端網路頻寬造成嚴重的影響，因此搭配了網路群播技術有助於將資料傳輸量進行有效的縮減。在應用層方面，採用群播技術可分為兩大類，分別是主從式（Client-Server，C/S）架構與對等式（Peer-to-Peer，P2P）架構，目前多採用主從式架構來改善大量資料傳輸問題。但儘管可透過群播技術改善轉播端對有限網路頻寬產生壅塞的影響，但對收看端來說，大量點閱還是會對伺服器造成大量的重複資料傳輸，進而對伺服器端的網路傳輸產生影響。因此比起會對伺服器端造成壅塞壓力的主從式架構，本計畫將以對等式網路架構為主，開發一套提供使用者方便使用且具有跨平台執行能力的即時視訊串流群播軟體。

(三)文獻回顧與探討

1. 群播網路架構

1.1. 主從式 (Client-Server, C/S) 網路架構

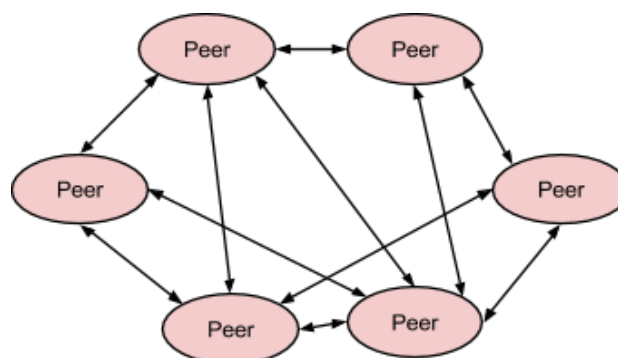
主從式架構如圖一，是將客戶端 (client) 與伺服器 (Server) 區分開來的一種網路架構。伺服器，意即服務的提供者，會管理其資源並為客戶提供服務的電腦軟體，它會被動的等待來自客戶端的請求 (request)，進行處理後給予客戶端回應 (response)。而客戶端便是主動的向伺服器提出請求以獲得服務。純粹請求、回應的相互關係相當簡單明瞭，由伺服器掌握所有客戶端連線資訊所形成的星狀拓撲也相當單純。然而當提出服務請求的客戶端數量龐大時，伺服器本身的處理速度就會造成服務提供的瓶頸，甚至直接使得服務中斷，因此在傳輸資訊量大與交互關係頻繁的視訊串流服務當中，主從式架構的伺服器需要較高的計算處理能力。



圖一：主從式網路架構

1.2. ^[2]對等式 (Peer-to-Peer, P2P) 網路架構

對等式網路如圖二，是由一個個分散且對等的節點 (Peer) 連結所組成，它不依靠中心伺服器，而是靠節點交換資訊的網際網路體系。P2P 架構網路中的每個節點都同時擔任伺服器與客戶端的角色，不只下載自己所需的資料，同時也提供其他客戶端下載的服務。此種方式將伺服器的負擔分擔至整個網路中的每個參與者，就算網路中的節點再多，整個網路也還是能夠正常運作，比起傳統的主從式架構能容納更多的使用者加入。然而沒有中心伺服器提供連線資訊，需要依靠用戶群之間的交流才能夠找到其他節點。



圖二：對等式網路架構

集中式對等式系統（Centralized P2P System）架構，會有一個中心伺服器來維護檔案列表及其擁有者節點連線資訊，若有節點欲進行搜索與下載時，會向中心伺服器發送請求以獲得節點資訊，再直接與目標節點建立連線以取得檔案。而節點一連上此系類系統或是檔案列表有所變更，便會將自身檔案列表與連線資訊傳送至中心伺服器。雖此架構簡單且僅讓伺服器負擔檔案查找之計算成本，但其效率與擴展性仍會受到中心伺服器影響而產生瓶頸，且若中心伺服器因受攻擊或硬體毀損等各種因素導致離線，會使得整個網路失效。P2P 熱潮蔓延之始祖，1999 年由 Shawn Fanning 所創造之 Napster 係屬此種架構。

分散/無結構式對等式系統（Decentralized/Unstructured P2P System）架構，最大的特點就是沒有中心伺服器與明確的網路拓樸（network topology）。由於沒有設定中心伺服器來做連結，因此節點對整個網路拓樸可說是一無所知，因此連線建立時僅需要知道任一節點之連線資訊後即可加入該 P2P 網路，並透過相鄰節點的資訊交換來了解整個 P2P 網路。而當需要搜尋及下載檔案時，會將查詢請求送給相鄰節點，若未找到其需求之檔案，便會再繼續將請求發送給相鄰節點，若有找到則回傳其連線資訊，使得提出搜尋請求的節點能夠直接連線後取得檔案。此種架構起始容易、無需維護中心伺服器，亦不因中心伺服器而產生瓶頸或問題，然而沒有結構性的網路路由卻會使得查找的效率下降，甚至造成網路傳輸量的大量增加。

分散/結構式對等式系統（Decentralized/Structured P2P System）架構，是利用分散式雜湊表（Distributed Hash Table, DHT）來建立網路拓樸。在系統中，每個節點都有其識別編號，並會維護相鄰節點之識別編號與其位址資訊。當有新的節點要加入此類 P2P 網路時，會透過雜湊表來將位址資訊轉換成識別編號並通知其相鄰節點進行資訊更新，而其所擁有之檔案亦會透過雜湊表轉換成關鍵值（key）建立關鍵值列表。當有節點欲進行檔案的查找與下載時，會先將其查找關鍵字轉換為關鍵值，再傳送查詢封包到關鍵值相同的節點位址取得其所維護的關鍵值列表以取得擁有該檔案之節點連線資訊。

2. 串流的群播服務

串流的群播服務，多數是以主從式架構為主，在採用主從式架構方面常見的串流群播服務有 Google Handout、Ustream 及 Wirecast。^[3]Google Handout 可當作視訊通話軟體，也可作為線上直播平台，它可支援圖片、影片、文字或是動畫等傳播方式，由於其不同平台上之運作方式皆相同，因此無論使用者是藉由電腦或是手持行動裝置皆可簡單的加入，視訊通話的部份還可當作線上視訊會議的應用，最多可同時容納 10 位使用者做雙向溝通，直播的部份則可以直接將影片在使用者個人的 Google+ 和 Youtube 平台中播放，就算結束後還是可以在 Youtube 做重複的觀賞。^[4]Ustream 提供使用者利用有錄影功能的設備與網路連線，將影片分享至 Ustream 自有平台或其他影音平台做分享，並可做即時的實況轉播，在手機部份更提供了行動裝置使用的應用軟體，讓使用者能夠隨時隨地直接做直播的服務，而且 Ustream 還具備了取得觀賞者電子信箱地址的功能，使得平台能夠提供分析圖表來讓使用者了解其觀眾族群，並適時的提出調整，更可與觀賞者利用訊息做互動。^[5]而 Wirecast 可自行開台做直播的服務，使用者能夠使用任何有錄影功能的設備進行影片的錄製，例

如手機、錄影機、網路攝影機、相機等，錄製完畢後利用 Wirecast 將之分享至線上影音平台。

而採用 P2P 架構之串流群播服務方面，^[6]SopCast 是一套基於 P2P 的網路直播軟體，其核心是由 SopCast 自行定義開發的 SOP 通訊協定來運作。其運作架構中會有一台伺服器 SopServer 作為一個頻道的 P2P 種子提供給數個觀看者（peer），而有興趣的使用者再向 SopServer 取得種子後根據內容取得視訊數據加入觀看，又會將自己的數據資訊放入種子中分享給其他的觀看者。依其加入順序變成第一級、第二級、第三級等階層組成的網狀 P2P 網路，任何一個觀看者都有一些邏輯上的相鄰節點，並與之保持雙向的連接、共享數據關係。

3. ^{[7]-[14]}會談發起協定（Session Initial Protocol，SIP）

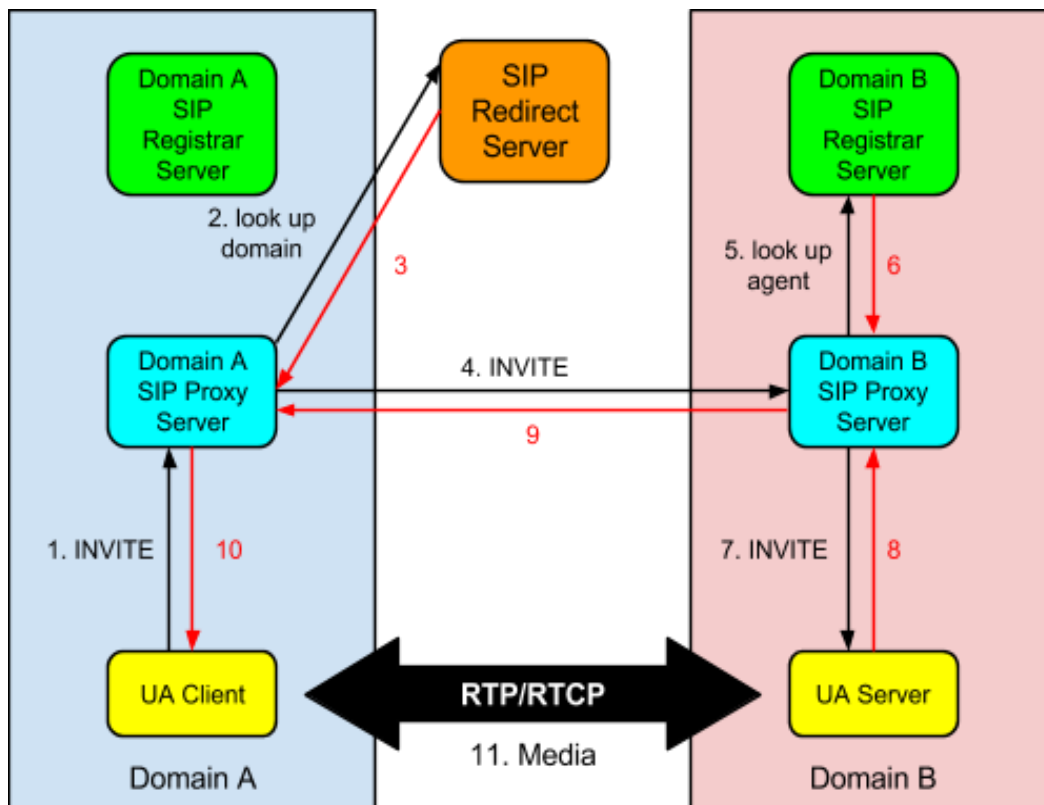
SIP 是個簡單、方便使用且泛用的信號控制通訊協定，可以用來開啟連線服務、結束正在使用的連線服務與管理設定既有的連線服務要求。SIP 能夠在異質的傳輸層協定上運作，並可用於開啟多類型的多媒體視訊串流服務，包含影像、聲音與文字等。簡言之，SIP 是一個使用於應用層的信號控制協定，可以被應用於開啟或管理任何型態的連線服務。

SIP 採用類似於電子郵件地址的方式來辨別使用者，稱之為 SIP URI，每個使用者都有唯一的帳號，其格式為 SIP 帳號@host，其中帳號是使用者自定義之名稱，可能是暱稱、名字縮寫或其他有意義的字元所組成，host 則是這個帳號註冊主機的域名（domain）。

SIP 訊息（message）的格式與通訊模式近似於 HTTP，用起始行（start Line）、訊息表頭（message header）以及訊息本體（message body）三部份組成一個完整的 SIP 報文。起始行會隨著請求（request）和回應（response）而呈現不同的格式。當 SIP 報文屬於請求時，起始行會以一個大寫英文單字開頭，儲存所要進行的方法（method）、目的端的 SIP URI 以及 SIP 版本。方法指的是希望目的主機做的動作，例如 RFC3261 中提及了六個基本 SIP 方法，包含 INVITE、ACK、BYE、CANCEL、REGISTER、INFO、OPTION，利用這六個基本的 SIP 方法，可以滿足目前大部分的 SIP 應用。目的端 SIP URI 則指出了要進行此動作的主機，SIP 版本則是說明此訊息所使用的版本。而當目的端主機收到 SIP 請求訊息後，會回傳 SIP 回應訊息。此時的起始行則是由 SIP 版本、狀態碼（Status Code）及原因短語（Reason Phrase）所組成。SIP 版本一樣是代表著訊息所使用的版本，狀態碼則是以三碼十進位的數字表示對於請求的處理狀況，狀況分為六大類，例如 1XX 代表一些進行中狀態的說明、2XX 表示成功、3XX 代表轉向的狀況、4XX 是指請求端的錯誤、5XX 表示主機端的錯誤、6XX 代表一些全域性的錯誤，而原因短語則是對於處理狀況有更進一步的敘述。訊息表頭主要是用來傳送額外的資料，訊息本體則是 SIP 訊息的文本（text），主要是用來傳遞會談描述協定（Session Description Protocol，SDP）的訊息。

在 RFC3261 中，SIP 被定義了三項網路元件，分別是代理伺服器（Proxy Server）、註冊伺服器（Registrar Server）、重導伺服器（Redirect Server）及使用者代理（User Agent，UA），可以用來呈現完整的 SIP 通訊協定行為。當 UA 要和另一

個 UA 建立連線時，會先將目的端 UA 的 SIP URI 送給代理伺服器。若目的端 UA 在同一網域（domain）中，代理伺服器則會向註冊伺服器詢問目前目的端 UA 的位址資訊並直接傳送；反之，代理伺服器則會向重導伺服器查詢另一個網域資訊，再轉送給另外一個網域的代理伺服器，由另一個網域的代理伺服器來負責傳送，連線建立過程如圖三所示。



圖三：來源端與目的端 UA 的連線建立過程

4. ^[15]會談描述協定（Session Description Protocol）

SDP 係為一會談（session）描述的格式，用來定義在網路上該如何描述會談的協定，而會談泛指為包含所有往來於發送端（sender）與接收端（receiver）的數據流（data stream），數據流包含資料、控制資訊等。由於它是一個描述會談的協定，所以只被使用於不同但與會談相關的傳輸協定當中，包含 SIP、即時串流協定（Real Time Streaming Protocol，RTSP）等。

SDP 是一種基於文本（text）的協定，描述是由許多文本行所組成，格式為 `<type>=<value>`，`<type>` 皆為一個英文字母，而 `<value>` 的部份則是結構化的文本，其格式依 `<type>` 而定。必須存在的 `<type>` 有 v、o、s、t、m 五項：v 代表的是 SDP 的版本，目前的版本號為 0；o 代表的是會談來源端的描述，格式為 `o=<User Name> <Session ID> <Session Version> <Network Type> <Address Type> <Destination Address>`；s 指的是此會談的名稱，會以通用字元集（Universal Character Set，UCS）來表示；t 表示此會談開始與結束的時間，使用網路時間協定（Network Time Protocol，NTP）來表示；而 m 則是媒體描述，媒體分為音訊、視訊、應用、數據以及控制五種。而常見的 `<type>` 還有 a 用來描述屬性、c 用來表示連線資訊、u 表示 URI、b 描述送出會談所用的頻寬等。

5. ^[16]即時傳輸協定 (Real-time Transport Protocol, RTP) 與即時傳輸控制協定 (Real-time Transport Control Protocol, RTCP)

RTP 詳細說明了在網際網路上傳遞音訊和視訊的標準封包格式，RTCP 則是為 RTP 提供信道外 (out-of-band) 控制，進行服務品質 (Quality of Service, QoS) 的監視與回饋、媒體間的同步以及群播成員標識。在 RTP 與 RTCP 的協作之下，能夠將多媒體資料進行打包與傳輸。

RTP 基本上是一個建立在用戶數據報協定 (User Datagram Protocol, UDP) 之上的協定，但也可以指定使用傳輸控制協定 (Transport Control Protocol, TCP) 來進行，不過 RTP 必須使用偶數的通訊埠 (port)。多媒體資料經過壓縮編碼處理後，會交由 RTP 封裝後進行分組，再依序將封包傳輸出去。然而由於 UDP 不提供 QoS 或傳輸可靠性的保證，因此 RTP 只能利用時間戳記 (timestamp) 及序號 (sequence number) 來達成同步功能及確定封包是否遺失。RTP 封包的表頭主要為承載資料類別 (payload type)、序號、時間戳記與來源識別碼 (source identifier)：承載資料類別指定了多媒體資料的編碼方式，方便接收端進行解碼；序號欄位則是在依序送出 RTP 封包時，每送一個出去數值就會加一，使得接收端能夠進行封包重組與遺失偵測；時間戳記能夠讓接收端進行時間同步；而來源識別碼能夠讓接收端識別 RTP 串流的來源端。

RTCP 是設計和 RTP 一起使用，用來進行流量、壅塞控制的協定。當應用程式開始一個 RTP 會談時，RTCP 會使用 RTP 的下一個通訊埠號。RTCP 的運行並沒有和 RTP 同步，而是週期性的發送 RTCP 封包給接收端，用以監聽服務品質以及交換會談使用者資訊。RTCP 封包的報文被定義了五種，分別是發送端報文 (Sender Report, SR)、接收端報文 (Receiver Report, RR)、來源描述 (SDES)、離開通知 (BYE) 以及應用程式自定義 (APP)。發送端報文與接收端報文就是 RTP 封包的發送端與接收端所發送的報文；來源描述的主要功能是作為會談成員有關標識資訊的載體，例如使用者名稱、地址、號碼等，也可用來作為傳遞會談控制資訊；離開通知是用來指示自己將退出會談；應用程式自定義報文則是由應用程式自行定義，使得 RTCP 的擴展性十分充裕，提供了相當的靈活性。

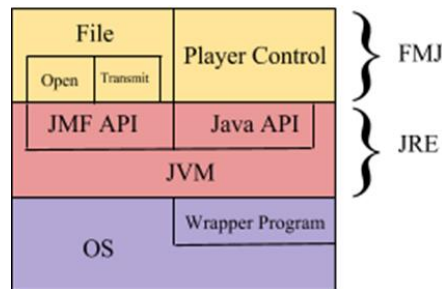
(四)研究方法及步驟

本計畫目標是開發一套即時視訊串流群播軟體，且此軟體會以 P2P 網路架構去進行串流群播服務。對即時串流提供者而言，此軟體能夠讓提供者製作種子，並將種子利用討論區或任何方式分享出去；對觀看者而言，只要取得種子，此軟體就要能夠進入該 P2P 網路，找尋適合的節點以接收即時串流。且此軟體應具備跨平台之執行能力。為了達成目標，將進行以下研究步驟：

1. 軟體架構設計及相關文件研讀

在進行軟體的開發與實驗前，需先進行 SIP、SDP、RTP、RTCP 等相關協定標準的研讀，理解其運行原則與規格，以進行後續實作之 SIP 機制修改及群播軟體開發。軟體開發部份預計以 Java 語言進行，以簡化軟體跨平台開發的複雜度，並預計使用 Freedom for Media in Java (FMJ) 這套開放原始碼的框架 (Framework) 作為開

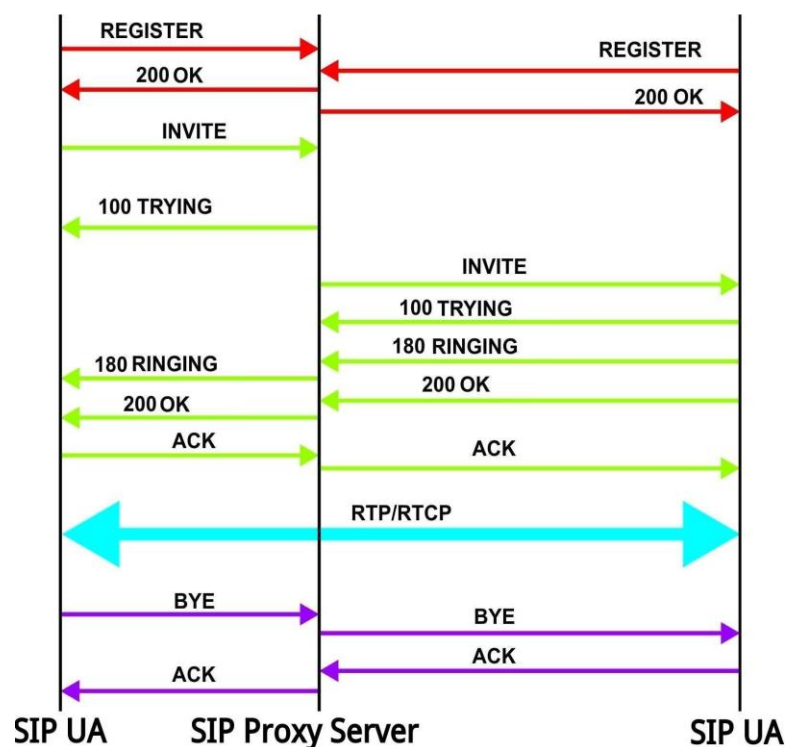
發的基礎。^[17]圖四為 FMJ 運作組架構圖。FMJ 必須要運作在 Java 執行環境（Java Runtime Environment，JRE）上，裡面有檔案（File）與播放控制（Player Control）兩大模組。播放控制模組主要負責播放器軟體的使用者介面功能操作。檔案模組則是負責影片的資料讀取、傳送等功能。而在多媒體檔案的編解碼、封裝等方面的支援，部份是使用 Java Media Framework（JMF）的 API，大部分則是搭配作業系統所使用的外覆程式（Wrapper Program）來進行，如 Windows 系統搭配 Direct Show、Mac OS X 搭配 Quicktime for Java 等，或是跨作業系統可安裝 Gstreamer 或是 FFMPEG。



圖四：FMJ 運作模組架構圖

2. SIP UA 軟體實作

標準的 SIP 是為一對一視訊串流服務提供連線建立與管理的通訊協定。在此步驟所需完成的即是 SIP UA 連線建立、串流檔案傳播等所需之軟體，SIP 僅僅只是涉及通訊會談的控制指令部份，輔以 SDP 來描述會談資訊，而涉及串流檔案傳播則需要協同使用 RTP 及 RTCP。圖五為 SIP UA 間之交互關係。通訊會談的控制訊息（control message）是以 SIP 進行，而 SIP 訊息本體會有 SDP 訊息，^[15]當連線建立完成後，多媒體串流的傳輸則是運用^[16]RTP 及 RTCP 來完成。本計畫在 SIP UA 軟體實作方面，會依照上述標準來進行。



圖五：SIP UA 間之交互關係

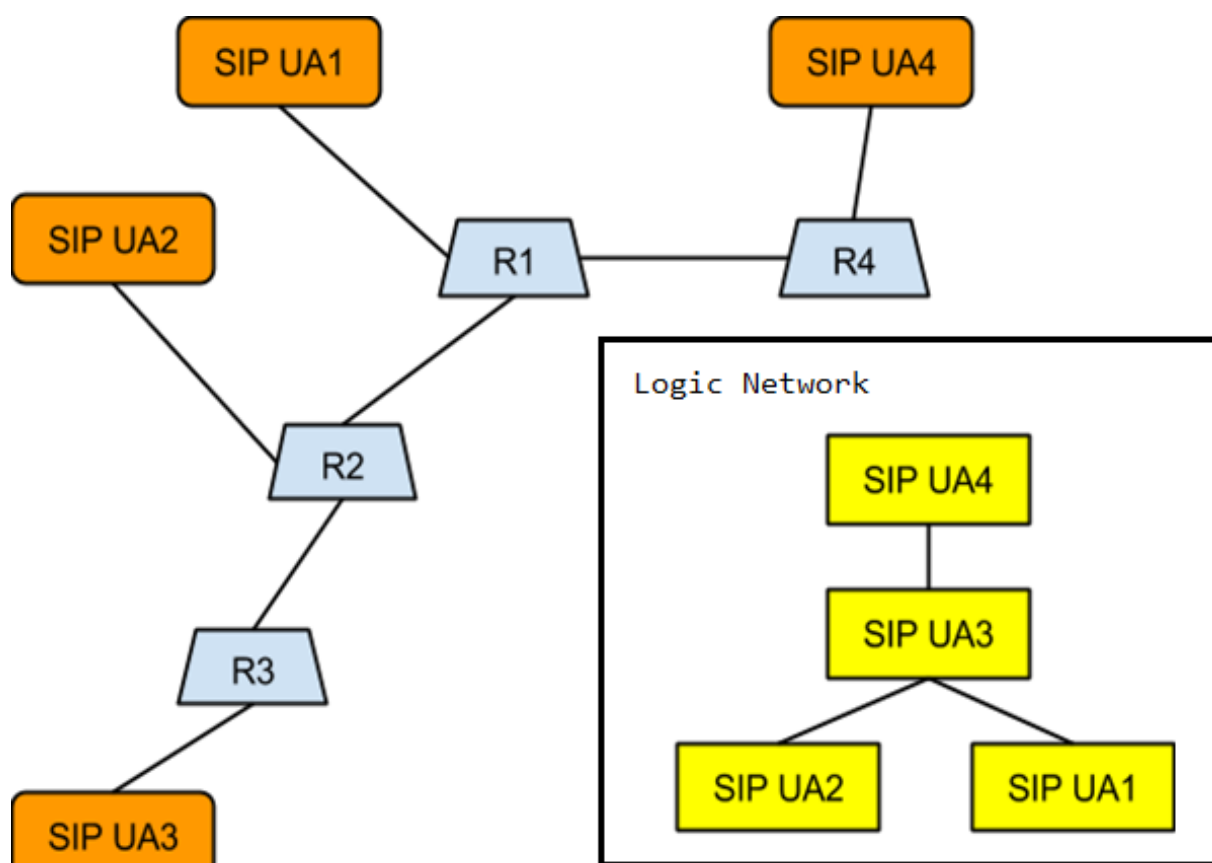
3. 群播服務提供

3.1. 作種

為使得觀看者能夠加入該提供者之串流網路，此步驟會讓提供者能夠製作 P2P 種子，裡面包含了轉播資訊、SIP URI 等。

3.2. P2P 網路的加入

本計畫預定的所採用的架構如圖六所示。SIP UA4 為轉播端，將包含其轉播資訊、SIP URI 的種子以任何方式分享出去。SIP UA3 為第一個與 SIP UA4 產生連接的收看端，因此 SIP UA4 會與 SIP UA3 產生直接的連線。而 SIP UA1 與 SIP UA2 為後續有興趣而提出連線請求的收看端，當它們提出加入請求時，透過轉播端取得可連線之節點資訊後建立連線。此架構將允許每個轉播端或轉送端設定可分享的連線數量，在圖六的例子中，SIP UA4 僅允許一個收看端，而 SIP UA3 則允許多個收看端，因此 SIP UA1 與 SIP UA2 將提出請求而取得轉送端的資訊，接著分別向 SIP UA3 提出連線請求而產生資料通道的直接連線，如此一來 SIP UA3 收到 SIP UA4 的任何視訊串流封包將轉送給 SIP UA1 與 SIP UA2。



圖六：P2P 群播網路架構

4. 實作以 SIP 為基礎的 P2P 即時視訊串流群播軟體

完成前述之研究步驟後，最終便是將所有環節整合在一起，完成計畫目標所提出之以 SIP 為基礎之 P2P 即時視訊串流群播軟體，並進行使用者介面的優化，使得使用者能夠方便使用之。

(五)預期結果

本計畫之目標為開發一套以 SIP 為基礎之 P2P 即時視訊串流群播軟體，預期此軟體能夠達到以下結果：

1. 此軟體能夠方便讓即時串流提供者製作種子，以便分享給其他有興趣觀看者。
2. 此軟體能夠在使用者取得種子之後，順利進入 P2P 網路，找尋到適合進行串流傳輸的節點以接收即時串流，並在成功建立連線後亦成為可進行傳輸之節點。
3. 此軟體能夠對欲傳送串流之節點建立連線，並接收即時串流。
4. 此軟體能夠將接收到之即時串流播放出來，並傳輸給其子節點。

(六)參考文獻

- [1] Vangie Beal. “streaming“. <http://www.webopedia.com/TERM/S/streaming.html>
- [2] S. Saroiu, P. K. Gummadi, and S. D. Gribble. “A Measurement Study of Peer-to-Peer File Sharing Systems”. In Proceedings of Multimedia Computing & Networking 2002 (MMCN '02), January 2002.
- [3] Google Handout, <https://plus.google.com/hangouts>
- [4] Ustream, <http://www.ustream.tv/>
- [5] Wirecast, <http://www.telestream.net/wirecast/overview.htm>
- [6] SopCast, <http://www.sopcast.org/cn/docs/>
- [7] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler. “SIP: Session Initiation Protocol”. RFC 3261, IETF, Jun 2002.
- [8] H. Schulzrinne and J. Rosenberg. “The session initiation protocol: Internet-centric signaling”. IEEE Communications Magazine, Vol. 38, Oct. 2000.
- [9] C. Perkins. “IP Mobility Support”. Internet RFC 2002, October 1996.
- [10] X.W. Zhang, J.G. Castellanos, A.T. Campbell. “P-MIP : Paging Extensions for Mobile IP”. ACM Transactions on Networking, pp. 127-141, July 2002.
- [11] A. T. Campbell, J. Gomez, C-Y Wan. “Cellular IP”. Internet draft, draft-valko-cellularip-01.txt, October 1999.
- [12] A. T. Campbell, J. Gomez, S. Kim, B. Paul, T. Sawada, C-Y Wan. “A Cellular IP Testbed Demonstrator”. Proc. IEEE International Workshop on Mobile Multimedia Communications, pp. 145-148, 1999.
- [13] A. T. Campbell, J. Gomez, S. Kim, A.G. Valko, C.Y. Wan, Z.R. Turanyi. “Design, Implementation, and Evaluation of Cellular IP”. IEEE Personal Communications, pp. 42-49, August 2000.
- [14] P. De Silva, H. Sirisena. “A Mobility Management Protocol for IP-Based Cellular

Networks”. IEEE Personal Communications, pp. 31 –37, June 2002.

[15] M. Handley, V. Jacobson, C. Perkins. “SDP: Session Description Protocol”. RFC 4566, IETF, July 2006.

[16] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. “RTP: A Transport Protocol for Real-Time Applications”. RFC 3550, IETF, July 2003.

[17] Freedom for Media in Java, <http://fmj-sf.net/>