

多媒體串流應用：產品比較與實例分析

論文領域：視訊通訊

涂國祥 林盈達

國立交通大學資訊科學系

新竹市大學路 1001 號

TEL：(03) 5712121 EXT. 56667

E-Mail：tgs@app.geo.ncu.edu.tw，ydlin@cis.nctu.edu.tw

主要聯絡人：涂國祥 TEL：(03) 5712121 EXT.58554

摘要

多媒體串流在網際網路上的應用受限於頻寬而在過往幾年呈現緩慢成長的趨勢，由於目前寬頻網路的加速建置及 ADSL 的快速普及，相信未來幾年將是多媒體串流邁向成熟運用的時期。在此我們將為大家介紹目前的串流技術發展，包括 RTP、RTSP、SDP、SMIL 等相關通訊協定，以及相關的商業及自由軟體產品介紹，同時以蘋果電腦(Apple Computer)所提出的開放原始碼解決方案(Darwin Streaming Server)為例，探討目前串流媒體應用程式的實作技術，其中包括串流資料檔案格式、RTP 封包流量控制、RTP 封包切割、RTSP 在傳送 RTP 封包時流程控制以及伺服器端對客戶端緩衝區溢出控制等。

關鍵字：串流、Streaming、RTP、RTSP、Darwin Streaming Server(DSS)、Streaming Media

一、簡介

隨著網路傳輸技術的發展，網際網路的應用已不再侷限於文字或圖片的傳輸，結合聲音、影像的互動式多媒體網路是人類追求的必然趨勢。自 1995 年 RealNetworks.com 推出 RealAudio 1.0 以來，大量的多媒體資訊藉由稱為串流技術的新傳輸科技在 Internet 中迅速擴張。

串流是一種具有即時(Real-time)特性的多媒體傳輸技術，相較於傳統存放在網際網路上的多媒體資訊必須全部下載到本機端(Local Host)後才可開始播放的方法，串流技術讓客戶端只需預先下載極小一部份資料即可開始播放，大量節省了使用者在下載時的等待時間；再者串流技術的另一特色是並不將多媒體資料實際拷貝一份存放在本機端，而是利用客戶端緩衝記憶體(Buffer)的概念，使資料不經實體儲存而直接由緩衝記憶體讀取播放後丟棄，可節省本機端磁碟儲存空間且具有一定程度的智慧財產權的保護功能。正因為串流技術的這些特質，使得其在網際網路上的利用極為廣闊，舉凡線上教學、視訊會議(Video Conference)、線上廣告、隨選視訊(VoD)、即時線上轉播以及網路電台(NetRadio)等各種網路多媒體應用都可運用串流技術來達成。

時至今日，網際網路上存在著超過 2000 個各式的網路電台及超過 100 個線上電視節目，每週更有超過 350,000 小時的線上內容產生。根據麥肯錫季刊[1](2001 年第 3 季)的預測，未來幾年內(美國)由於寬頻網路的建構完成將使得串流媒體的市場由 1400 萬美元(2001 年)以等比級數成長至 75 億美元(2005 年)，尤其是線上多媒體廣告及隨選視訊等具革命性的應用更將對媒體產業產生革命性的衝擊。同時來自 DFC Intelligence 的分析師更將串流媒體廣告類比為有線電視廣告市場[2]，美國有線電視廣告市場由 1980 年的 5000 萬美元成長至 2001 年的 131 億美元，而全美定時使用串流媒體服務者也將達到 8400 萬人。

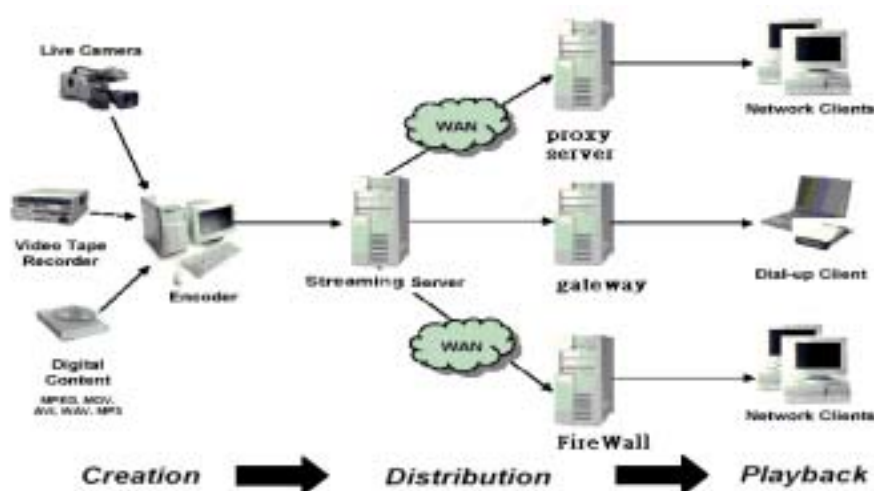
然而，串流媒體的未來雖美，卻仍然面臨著一些急需解決的問題。首先是流暢度(Smoothly)的問題，由於網際網路的特性使然，網路傳輸延遲的時間並不是固定的，以致客戶端每秒所能接收的資料量並不固定。然而多媒體資料的播放必須保證在一定的傳輸速度否則就會產生畫面延遲或不連續的問題，而多媒體播放的平順與否是多媒體內容最重要的關鍵；再者是網路頻寬使用公

平性的問題，最新的串流技術多使用 UDP/IP 為傳輸協定。UDP 因為沒有擁塞控制的機制，所以相較於 TCP 有較好的頻寬佔有能力。但大量的 UDP 封包產生必將導致對 TCP 封包的排擠，從而影響到網際網路目前現存的大部份以 TCP 為基礎的資料傳輸，因此尋求一個能夠讓 TCP 與 UDP 共榮共存的方法實是當務之急；當然，尋找一種具高品質及低資料傳輸速度的視訊壓縮/解壓縮方法(CODEC)一直是多媒體領域的重要課題；另外伴隨著串流媒體應用而來的其他問題，例如隨選視訊及線上音樂的計費機制、智慧財產權的保護等都是還需要更進一步研究的課題。

接下來的文章將為大家介紹目前常用的串流傳輸技術，以及目前商業及自由軟體串流媒體相關產品，同時藉由檢視 Apple Corp.所提出的 Darwin Streaming Server 來了解整個 Streaming Media 的實際運作流程。

二、串流技術及其通訊協定

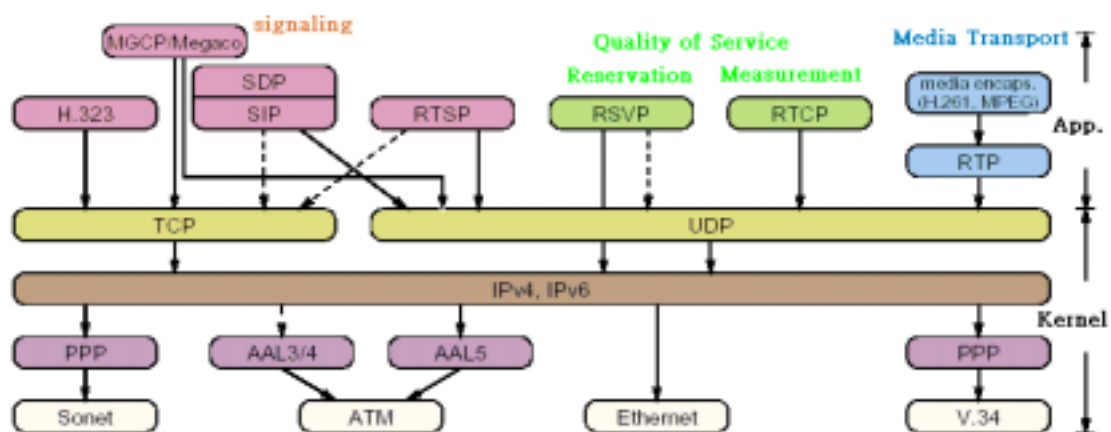
串流媒體是建構在網際網路上的應用，其大致組成元件及流程如圖一所示，首先由媒體擷取裝置取得數位化的媒體資訊，經過編碼過程產生可以支援串流的媒體格式，再經由串流伺服器透過網際網路將串流媒體傳送到客戶端，客戶端經過解碼程序後即可播放。



圖一：串流媒體的組成元件

至於串流媒體在網際網路上的傳輸方式目前主要有兩種方式，一種是以 HTTP/TCP 為基礎，另一種是以 RTP/UDP 為基礎。因為以 HTTP 為基礎的 WWW 是目前網際網路上使用最廣的服務之一，利用 HTTP 協定可以讓串流媒體得以穿過防火牆的阻隔，且將串流媒體放在 WWW 上可以讓使用者方便的取得這些內容，然其缺點在於 TCP 通訊協定將會導致傳輸速度的減緩，進而增加串流媒體所需要的頻寬，且串流資料必須事先放置於網頁上而無法達到真正即時的效果。使用 RTP/UDP 的優點在於 UDP 可以提供較好的傳輸效率，且可以達到真正即時的效果並提供群播等有效率的傳輸方式，缺點則是必須架設另外的串流伺服器，同時如果有防火牆的問題則需要在防火牆兩端加裝 translator 的設備以便穿過防火牆。

在目前網際網路有關多媒體的眾多協定中，各通訊協定在整個協定架構中的所在位置大致如圖二所示，在此僅就與串流媒體直接相關的 RTP、RTSP、SDP、SMIL 等協定作一簡介。



圖二：網際網路多媒體協定堆疊[3]

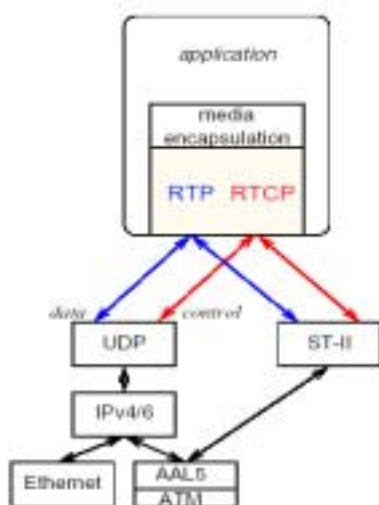
2.1 RTP(Real-time Transfer Protocol , RFC1889)[4]：

RTP 是為了處理具由即時特性的資料而訂定的一個屬於應用層的端對端(End-to-End)傳輸通訊協定，主要分為兩大部分：RTP 及 RTCP(RTP Control protocol)。圖三為其架構，其中 RTP 負責

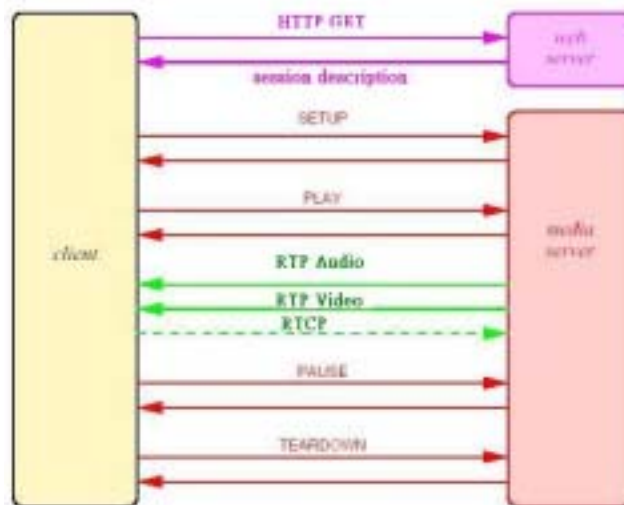
Data 的傳輸，而 RTCP 則負責對該 RTP session 的參與者提供傳輸品質等相關的資訊以達到一部份的服務品質(QoS)的控制。RTP 的連線埠號(Port)必須為偶數號碼，而 RTCP 的連線埠號必須為該偶數號碼的下一個數字。RTP/RTCP 並不提供服務品質或傳輸可靠性的保證，只提供時間戳記(Timestamp，用來達成同步等能力)、序號(Sequence Number，用來決定封包是否遺失等資訊)等一些具有即時特性資料的基本功能。因此 RTP 必須需要其他的輔助性資訊來完成串流媒體的實作，例如：RTP 封包負載資料(Payload)的格式等。

2.2 RTSP(Real-time Streaming Protocol , RFC2326)[5] :

RTSP 可以稱為一個對多媒體服務做遠端控制的通訊協定，RTSP 在語法上與 HTTP 非常相近，皆是屬於純文字的通訊協定，其請求(Request)訊息格式為被請求者的 URL 位址及協定版本，接著是標頭及訊息本文。不同於 HTTP 的單向溝通，RTSP 的通訊以客戶/伺服器雙向的請求/回覆來完成溝通。訊息本文可能包含一“會議描述(session description)” 描述該會議中串流媒體的訊息，此一會議描述的格式並未限定，然目前大部分皆以 SDP[6]為其訊息格式。運用 RTSP 協定中所定義的各種方法(method)，可以達成對串流媒體的各種控制。



圖三：RTP/RTCP 概觀



圖四：RTSP 通訊流程範例

2.3 SDP(Session description Protocol , RFC2327) :

SDP 是一種以文字格式描述多媒體會議(session)的通訊協定，嚴格說 SDP 並不算是一種通訊協定，而在內涵上比較傾向於類似 HTML 的標記語言。SDP 可以被包含在 SIP、RTSP、SAP、HTTP 甚至 E-Mail 等各種協定中。SDP 協定中可以包含各種有關 session 的描述：媒體種類 (Video/audio/shared application 等)、媒體傳輸協定(RTP/UDP/IP 等)、媒體格式(H.261 video/MPEG video/G.723.1 audio 等)、還有一些有關安全方面的訊息(如該媒體會議的加密鍵值)等資訊。下圖五是一個用 SDP 描述 RTSP DESCRIBE RESPONSE 的範例,其中“O=”定義該 session description 的來源及其版本；“S=”提供關於內容的資訊；“m=”定義媒體種類、埠號、傳輸協定及支援的編碼方法；“a=rtpmap”則列出 RTP Payload 的種類、編碼方法之名稱、取樣頻率、編碼參數等資訊；“a=control”則列出 RTSP request URL。

```
o=- 2890844526 2890842807 IN IP4 192.16.24.202
s=Twister PG-13 (c) Warner Bros.
m=audio 0 RTP/AVP 98
a=rtpmap:98 L16/16000/2
a=control:rtsp://audio.example.com/twister/audio.en
m=video 0 RTP/AVP 31
a=rtpmap:31 MPV
a=control:rtsp://video.example.com/twister/video
```

圖五：用 SDP 描述一個 RTSP Multimedia session 的範例

2.4 SMIL(Synchronized Multimedia Integration Language)[7] :

SMIL 是 W3C 所提出一個具有擴充性的標準，SML 文件屬於 XML 文件的一種。利用 SMIL 文件可以達成讓數個獨立的媒體物件整合為一個具有同步化能力的多媒體呈現。簡言之，一份 SMIL 文件是用來描述數個媒體物件中哪一個物件應該在何時、何地以何種方式呈現出來。媒體物件在 SMIL 文件中是以 URL 的方式指定，可能是一個文字檔案、JPEG 圖片、影片或是串流檔案。由於 SMIL 是以 XML 為基礎，所以具有比 HTML 更為嚴謹的結構。但是與 HTML 一樣具

有<tag>與</tag>語法結構。需注意的是 SMIL 的所有 tags 都必須是小寫的形式，同時每個 tag 都必須<tag>與</tag>成對或是以<tag parameters />的形式表示。

```
<smil>
  <head>
    <layout>
      <!--layout tages -->
    </layout>
  </head>
  <body>
    <!-- body tags -->
  </body>
</smil>
```

圖六：SMIL 文件範例

經由 RTP/UDP(TCP)組成的資料傳輸方式，及 RTSP/SDP 所構成的串流媒體控制機制，再加上由 SMIL 及 HTTP 等輔助機制，一份串流媒體資料就可以從伺服器端經由網際網路傳輸到客戶端

三、串流產品綜覽

在目前的商業市場中，由於串流媒體伺服器端的規格並未有一個權威性標準訂立，以至於各家廠商紛紛推出以自己的標準或特有通訊協定所開發的串流媒體伺服器程式，導致了一套串流伺服器只能對應一套編碼/解碼格式及一套串流播放軟體的現象。綜觀市場，目前串流媒體的三大主流分別是 RealNetworks.com 的 RealSystemR[8]、Microsoft 的 Media ServicesR[9]以及蘋果電腦主推的 Quick Time ServerR[10]。RealNetworks.com 由於較早進入串流市場，目前居於串流媒體廠商領導者的地位。Microsoft 近來挾著 Windows 作業系統的威力強力搶進串流市場，其所推出的 Windows Media Player 因內建於 Windows 系統中，加上其方便的使用介面使得聲勢逐步逼近 RealSystem。至於 Apple 則以其在影像處理上的特長推廣其 Quick Time 格式。由於 Microsoft 的壓力，目前 RealNetworks 及 Apple 的串流產品開始進行整合的工作，以期能藉著彼此串流格式的支援擴大使用者的目的，Apple 為了推廣其 QuickTime 格式更進一步開放其 QuickTime Streaming Server 的原始

碼，企圖以 Open Source 的力量增進其影響力。另一方面，Philips 則是推出以 MPEG-4 為基礎的串流解決方案(WebCine)，然其在目前市場上的佔有率可說微乎其微。以下就三大主流串流媒體作一比較。

	Real System	Windows Media	QuickTime
Client	Real Player	Media player	QuickTime Player Real Player
Primary Protocol	RTSP+(UDP & RTP)	MMS+(UDP or TCP)	RTSP+(UDP & RTP)
Support HTTP delivery?	Yes	Yes	Yes
SMIL support	Yes	Yes	Yes
Multicast support	Yes	Yes	Yes
Web-based administration	Yes	Yes	Yes
Digital Rights Management	Yes	Yes	Yes
Remote Encoder Control	No	Yes	Yes
OS support	Windows/Linux Solaris/ HP/UX IBM/AXIs	Windows	MAC OS X
Variable Bit Rate encoding	Yes	Yes	Yes
File Format	AU、MP3、AIFF、 Mov(H.261,PCMU)	ASF、MP3	MOV(Hinted)

表一：三大商用串流軟體功能比較表

根據 2000 年 11 月 Nielsen//NetRatings.com 的一份調查報告顯示，使用 windows media format 的 Internet streaming user 正在迅速增加中，雖然 RealNetworks 的領先仍然明顯，不過差距正在快速縮小中，RealNetworks 在與 Microsoft 在串流媒體市場的戰爭結局會不會是另一個網景？值得我們關注。

	2000 年 7 月	2000 年 11 月	總增加人數	使用者成長率
Windows Media	10.9 million	13.2 million	2.3 million	20.7%
RealNetworks	27.1 million	27.7 million	0.6 million	2.3%
Apple QuickTime	8.1 million	7.9 million	-2 million	-2.5%

表二：三大串流媒體格式使用率(來源：Nielsen//NetRatings, November 2000)

在 Open Source 方面，由於受限於編碼解碼器(Codec)及串流媒體的標準未定，open Source 的資源顯得非常紊亂，且大部分在進行中的發展套件多為支援 Shoutcast 等播放 MP3 的計劃(例如 Icecast[11]、Fluid 等)。值得一提的是 Apple 依據 APSL(Apple Public Source License)授權所釋出的 Darwin Streaming Server(DSS)[12]，這是一套以 QuickTime Streaming Server 核心程式為基礎的開放程式碼，DSS 的釋出為 open source 領域帶來新的契機，目前已有計劃開始利用 DSS 核心來進行 MPEG-4 的 open source streaming 整合計劃(MPEG4IP[13])，相信不久之後，開放原始碼領域必將產生重量級的串流媒體伺服器程式。下面對目前開放原始碼領域較多人使用的串流程式作一個簡單的比較。

	Darwin Streaming Server	Icecast	Obsequiem[14]	MPEG4IP
Support protocol	RTP、RTSP HTTP	HTTP(ShoutCast)	HTTP(ShoutCast) , RTP(Multicast)	RTP、RTSP HTTP
File Format	Mov	MP3	MP3	MPEG-4
OS support	Linux、Windows	Linux、	Linux、	Linux
Client	QuickTime	FreeAmp、xmms	FreeAMP、xmms	GMP4Player
Multicast support	Yes	No	Yes	Yes
License	APSL	GPL	GPL	APSL、LGPL、GPL

表三：開放原始碼領域串流軟體功能比較

綜觀整個串流媒體領域，因為串流規格的標準未定，以至於產生目前多家爭鳴的現象，商業產品因為有雄厚的財力及人力資助，而有遠較開放原始碼套件完整的整體解決方案。雖然在編碼 / 解碼格式方面各家皆有所好，但可喜的是在網路傳輸協定上因為標準協定的公佈而有趨向一致的趨勢(雖然 Microsoft 推出 MMS 通訊協定，實是 RTP 的 Microsoft 版本)，且 RealNetworks 與 Apple

也開始相互支援。在 Open source 領域因為 DSS 的出現有了新的活力，同時多個正在進行中的計劃相信將會為 open source 增添新的力量。

四、實例探討：Darwin Streaming Server(DSS)

雖然我們已經了解了目前串流媒體大概的處理流程，然而並無法了解其資料速度的控制、緩衝記憶體配置、封包切割等運作機制，因此經由實際追蹤串流媒體的原始程式來了解其內部運作流程是有其必要性的。DSS 是一個依照 IETF 標準實做的一個具有強固性的串流伺服器程式，所以我們選擇這個伺服器程式來作為我們深入了解的對象。

4.1 Darwin Streaming Server (DSS)簡介

DSS 伺服器程式是蘋果電腦的開放程式原始碼計劃之一，可以讓我們透過網際網路傳送串流媒體檔案，其核心程式為 QuickTime Streaming Server，目前最新的版本是 version3.0.1。DSS 伺服器程式中實作了 RTP、RTSP、SDP 等 IETF 的標準，具有傳輸品質保證，傳輸流量控制等控制機制，提供群播、單點播送、HTTP 遞送等多種傳輸方式、同時可以透過 Web 來管理系統的運作。整個 DSS 程式都是用 C++以物件導向的概念設計，具有良好的模組化架構。且目前已經支援 MAC OS X、Linux、Solaris、FreeBSD、Windows 2000/NT 等多種執行平台。

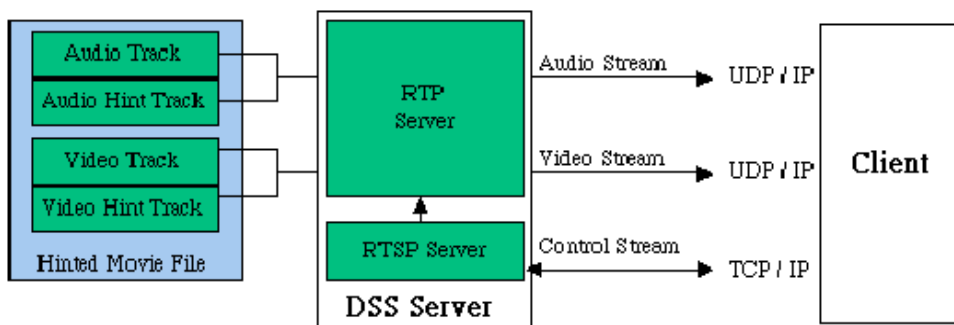
DSS 整個程式可分為兩大部分：共用工具部份與伺服器核心部份。其中共用工具部份主要負責執行緒管理、程式資料結構、網路底層運作以及文字(字串)處理等與執行環境相關性較高的工作。而伺服器核心部份則負責封包的處理、封包流量控制、RTSP 流程控制及建立連接等工作。在本文中我們將重點放在伺服器部份。

DSS 主體分類	各部細項分類	內容
----------	--------	----

工具部分 (Common Utilities)	OS Classes	與執行平台相關的資料結構或變數，例如：OS、OSThread、OSFileSource、OSQueue、
	Sockets	與執行平台相關的 TCP/UDP Socket 函式，如：TCPSocket、UDPSocket、UDPSocketPool
	Parsing Utilities	用來對字串文字作切割或重組等的工具集，如：StrPtrLen、StringFormatter 等類別
	Tasks	主要包含伺服器程式用來實作非同步事件機制的類別，如：Task、IdleTask 等類別
	RTCPUtilitiesLib	主要包含對 RTCP 封包作解碼或編碼及產生封包的類別，如：RTCPPacket 類別等
伺服器部分 (Server Core)	RTSP subsystem	主要負責分析與處理 RTSP 請求，以及實作 DSS 模組化 API 中關於 RTSP 的部分。每一個 RTSP TCP 連線都有一個"RTSPSession"物件。 "RTSPSession"物件是一個處理 RTSP 相關事件的"Task"物件。
	RTP subsystem	負責媒體資料的傳送。"RTSPSession"物件包含與相對應 RTSP session ID 的資料。每一個 RTP 串流都以一個" RTPStream"物件來表示。而一個 RTPSeesion 中可包含數個 RTPStream 物件。DSS 模組中 RTP 的部分就藉由 RTSPSession 與 RTPStream 這兩個類別來實作。
	Server Core	這子系統中各類別主要處理啟動(startup)與卸載(Shutdown)，Server 中具全域性的資料，以及相關的 callback 函式類別。

表四：DSS 程式模組架構分類

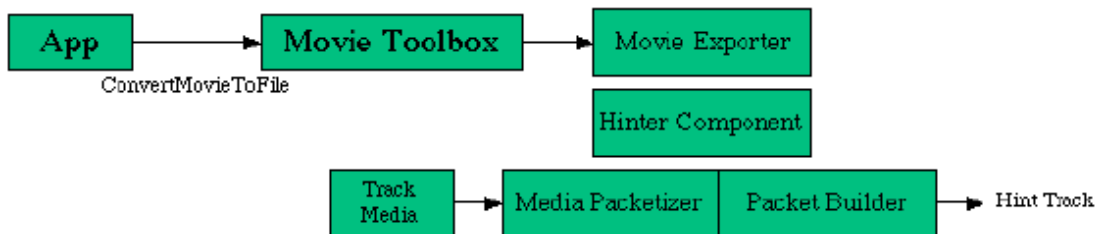
以 DSS 處理串流媒體的過程如圖七所示，客戶端經由 DSS 中的 RTP session 及 RTSP session 建立 RTP/UDP/IP 資料串流連接及 RTSP/TCP/IP 控制鏈路連接，接著串流媒體檔資料經由 RTP 鏈路傳送到客戶端，以 RTSP 鏈路控制 RTP 串流的播放，完成整個串流過程。



圖七：DSS 串流處理架構圖

在 DSS 中，能夠作為串流媒體資料的檔案稱之為 Hinted File。Hint 是指對一個非串流檔案作特定的格式化以使其符合串流檔案格式的過程，方法就是在原本的媒體資料軌道以外再加入一個新的軌道(Track)紀錄關於串流格式的資料，此一 Hint Track 記錄讓 Server 作串流封包切割時的必要資訊，還有一些將來要加入串流封包表頭的資料或是用來指明媒體資料位址的參考位址。通常 Hint 是由媒體封包切割元件(packetizer)來完成，且每一個資料軌道皆會產生一個相對應的 Hint Track(例如：若一檔案有一 video track 與一 audio track，則產生兩個 Hint track 分別指向 video 與 audio track)。經由 Hint Track，DSS 可以完全不需了解所傳輸的媒體資料內容的型別或壓縮格式等，只需要了解 Quick Time 檔案的格式正確找到 Hint Track，並依照 Hint Track 的資料即可切割封包並遞送。

我們可以經由 QuickTime Player pro.來附加 Hint Track 到現存的媒體檔案中，方法是經由呼叫"ConvertMovieToFile"來觸發一個"Movie Exporter"元件。該元件會產生一個對話框讓使用者設定 Hint 時的一些屬性。設定完成後，QuickTime 根據每個媒體資料軌的性質選定相應的 Media Packetizer，並將此 Packetizer 的輸出導入到"packetBuilder"產生包含 Hint Track 的串流檔案。



圖八：利用 QuickTime Player Pro 產生 Hint Track 的流程圖

4.2 Server 程式詳論

在討論 Serer 程式之前，我們必須先了解 DSS 的設計理念。DSS 的運作是以模組為基礎，

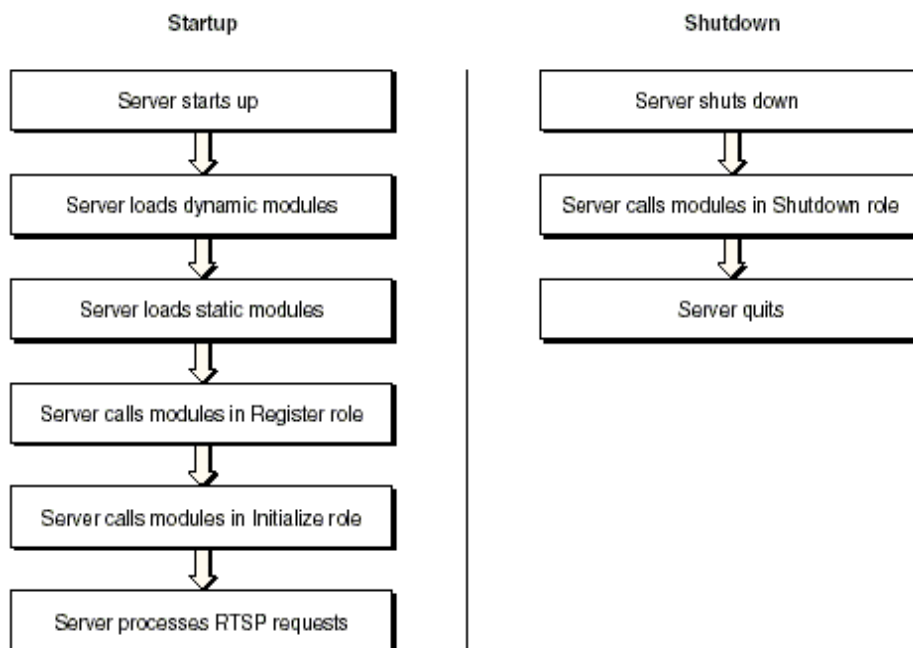
以角色(Role)來分類各項工作(Task) , 每一個角色皆是為完成某一類特定工作而設。當客戶端提出請求時, DSS 根據該請求屬於哪一個角色(Role)來決定並啟動適當的模組。

如前所述, Server Core 主要分三大模組: Server Core, RTSP subsystem 及 RTP subsystem。

Server 從啟始到結束整個流程也大略可分為以上三個部份(segment), 以下就以這三部份作為討論的主題。

Server Core(Startup & Shutdown) :

圖九為程式啟動與結束時的流程圖。Server 啟動時先載入未事先編譯(Compile)在程式中的模組(即 Dynamic Module), 在接下來載入靜態模組時若有相同的模組時則以動態模組為準, 這樣的設計可以讓程式設計者可以撰寫自己的模組以取代程式中具相同功能的靜態模組。當載入模組完成後就進入"Register Role", 此角色主要的工作是將各模組所支援角色註冊已備以後呼叫之用。完成註冊後, 具有"Initialize Role"的模組開始進行初始化的動作, 並進入聆聽(Listen)的狀態準備接收客戶端的 RTSP 請求。

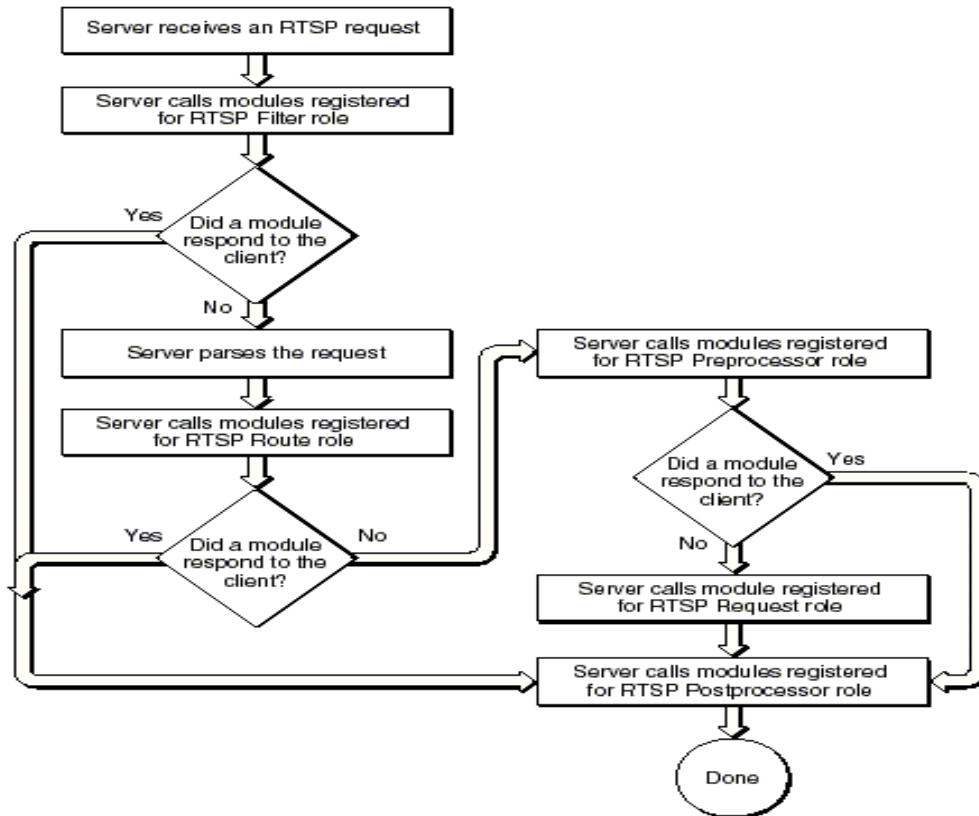


圖九：程式啟動與結束流程圖

結束流程則是呼叫具”Shutdown Role”的各模組執行 Shutdown 的工作,釋放資源並結束程式

當 Server 接收到一個 RTSP 請求的封包後,即進入 RTSP subsystem 的處理程序中。

RTSP subsystem(處理 RTSP 相關工作)：



圖十：RTSP 請求處理流程

當 Server 接收到一個 RTSP 請求時,先建立一個 RTSP 請求的物件,該物件儲存該請求的參數。接著根據圖十的流程處理此一請求。首先經由”RTSP Filter Role”對封包資料作某些改變等前置處理。當 RTSP Filter Role 完成,開始進行對該請求封包的解析以取得各種參數並建立一個 RTSP session 及一個 Client session。RTSP session 主要負責處理此一請求所建立的 RTSP 連線階段的請求/回應。Client session 則負責串流遞送階段的維護工作。對該請求分析完畢後,Server 呼叫

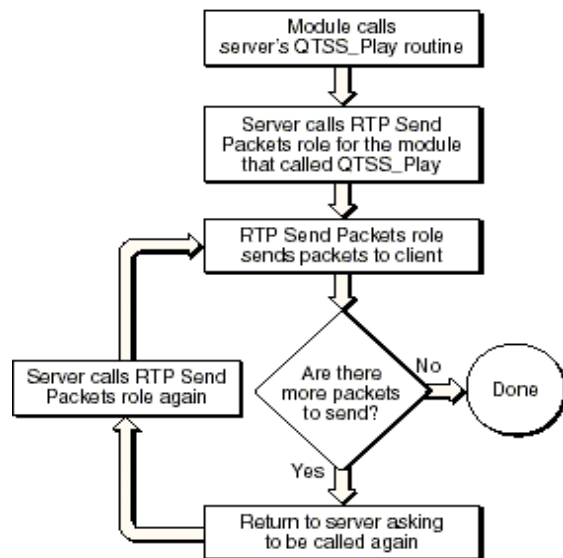
各模組中的 " RTSP Route Role " 對 RTSP 物件進行參數修改或設定。接著呼叫 " Preprocessor Role " 判斷該請求是屬於 RTSP 封包中的哪一種類型，以便把該請求傳送到對應的函式處理此一請求，若無法判別屬於哪一類型，則將其傳送到 " RTSP Request Role "，此一角色處理所有在 " Preprocessor Role " 中未定義的類型。最後，來到 " Postprocessor Role "，此角色負責統計的工作，例如存取紀錄等。

在處理 "Preprocessor Role" 或是 " RTSP Request Role " 中的各模組時，可能會產生串流媒體資料。需要產生媒體資料時，程式會呼叫 Client session 中的 " QTSS_Play " 物件，此物件會觸發在 " RTP Send Packets Role " 中的各個模組，同時進入 RTP subsystem 中。

RTP subsystem(處理 RTP/RTCP 串流封包遞送)：

RTP subsystem 主要包括 RTP 封包傳送及 RTCP 串流控制兩部份。在封包傳送方面，" RTP Send Packets Role " 在 RTP session 中經由呼叫 " QTSS_Write " 或 " QTSS_WriteV " 把媒體資料傳送到客戶端。在傳送過程中，每送一次封包後，Server 等待一個時間值再繼續呼叫 "RTP Send Packets Role" 傳送封包。在串流控制方面，當

Server 接收到 RTCP 封包時會先判定此 RTCP 封包屬於哪一種封包，RFC1889 中定義了五種 RTCP 封包，分別是：SR(Source description)，RR(Receiver description)，SD(Sender Report Description Items，include items)，APP(Application specific functions)及 BYE。根



圖十一：RTP 封包遞送流程

據封包的種類，Server 會呼叫 “ RTCP Process Role ” 中相應的模組進行處理。

在了解 DSS 整個運作過程後，我們把焦點放在下面兩個重點：DSS 對串流資料封包的切割

技巧以及伺服器端如何達成流量控制？這兩者的目的都是在使得客戶端的播放能夠順暢。

媒體封包切割(Media Data Packetization)：

在封包切割前，我們必須得到以下的幾個資訊：

- (1)在單一 RTP 封包中能容納的最大 QuickTime 媒體資料尺寸(Maxima QT Media Data Size, MQD)
- (2)此種媒體資料其所有取樣(Sample)尺寸是否固定？(Constant QT Sample Size, CQS)
- (3)此種媒體資料其所有取樣的持續期間(Duration)是否固定？(Constant QT Duration, CQD)
- (4)當所有取樣尺寸固定時其取樣尺寸為多少？(Constant Sample Size, CSS)
- (5)當取樣尺寸不固定時其某一個取樣的大小(Sample Size, SS)

根據以上的資訊，DSS 採用三種不同的封包切割方式：

- (1) (CQS=true) AND (CQD=true) AND (CSS<=0.5 * MQD)

把多個取樣封裝成一個 RTP 封包，同時每個 RTP 標頭中 M 欄位設為 1，QuickTime 標頭中

的 PCK 欄位也設為 1。(RTP 標頭中 M 欄位設為 1，表示此為某一取樣的最後一個封包；

QuickTime 標頭的 PCK 欄位表示該封包切割所採用的方式。)

- (2) ((CQS=false) OR (CQD=false)) AND (SS<=0.5 * MQD)

數個取樣封裝成一個 RTP 封包，同時此一 RTP 封包標頭的 M 欄位設為 1，QuickTime 標頭

中 PCK 欄位設為 2。

- (3) 不符合(1)和(2)的情況時，採用之。

單一的取樣被封裝成一個或數個 RTP 封包中，同時最後一個屬於該取樣的 RTP 封包之 M 欄

位設為 1，其他設為 0；QuickTime 標頭 PCK 欄位設為 3。

由於客戶端必須以一個取樣為單位進行封包重組，因此根據 RTP 的 M 欄位及 QuickTime 的

PCK 欄位，可以知道此一封包所採用的分割方式，同時採取相應的封包重組方法。另外在處理某一封包遺失時，也可以快速決定是否該丟棄與其相關的封包，增進客戶端的效率。

流量控制(FlowControl)

串流媒體有一個很重要的特色是在播放時必須維持等速度或至少相差不多的速度，否則會導致影片或聲音的不連續，稱之為不流暢(smoothless)。當伺服器端傳送封包的速度快過播放速度時，客戶端的緩衝區很快會被封包填滿，接著而來的封包就會被丟棄，將會導致播放的延遲或是畫面的失真。如果封包傳送慢過該媒體檔案的播放速度，則緩衝區將會因消耗大於接收而耗盡，將導致串流媒體播放的嚴重延遲。因此伺服器端在傳送封包到客戶端時，必須很小心地控制流量。下面介紹 DSS 中關於流量控制的機制。

DSS 利用客戶端傳送回來的 RTCP 封包資訊以及變數 “ Quality level ” 來達成流量控制的機制。這些資訊包括 “ Loss Rate ” ， “ Getting better ” 以及 “ Getting worse ” 訊息。

(1) Loss Rate :

如果客戶端的 RTCP 回報中封包遺失率連續數次大於某一數值(此值為一百分比定值,由程式給定,稱為 SLossThinTolerance), 則將 ratchetless 設為 True, 表示需要調降傳輸速度。如果 RTCP 回報中封包遺失率連續數次小於某一數值(此為一百分比定值, 稱為 SLossThickTolerance), 則將 ratchetmore 設為 True。

(2) “Getting Worse”及“Getting Better” :

如果客戶端的 RTCP 訊息發出“Getting Worse”的訊息，則直接將 ratchetless 設為 True。

如果客戶端的 RTCP 訊息發出“Getting Better”的訊息，則直接將 ratchetmore 設為 True。

由 ratchetless 或 ratchetmore 的設定，可以調整 QualityLevel 的數值。DSS 根據 QualityLevel 的數值大小來決定傳輸速度的快或慢，進而達到一部份的流量控制的目的。另外，如果程式設計者有更好的方法，也可以很容易的藉由覆寫此一模組的方式完成演算法的替換。

總體而言，DSS 是一個極具彈性的串流伺服軟體，程式設計者可以經由對原有模組的覆寫達到功能替換而不需要更動整體架構，同時也可以輕易的以加入新模組的方式達到功能擴充的目的。然而另一個有趣的問題是，在實際環境下其表現會是如何呢？接下來我們將對 DSS 在模擬環境下的表現作一個簡單的測試。

五、利用 WAN Emulator 實測 DSS

我們的測試環境如下：

Server 執行環境為 Linux mandrake8.0 kernel 2. 4. 3，DSS 3.0.1，IP 為 10.0.0.2；Client 端執行環境為 Windows 2000，QuickTime5.0.2，IP 為 192.168.0.2；Gateway 執行環境為 Linux mandrake8.0 Kernel 2. 4. 3，WAN Emulator 為 NIST 2. 0. 10，IP 為 10.0.0.1 及 192.168.0.1；串流媒體測試檔案為 DSS 所附的 sample . mov，size 為 814KB，分為 Video Track 及 Audio Track，壓縮格式分別為 X-Sorenson Video 及 X-QT，資料速度分別為 66Kbits/sec 及 21Kbits/sec。

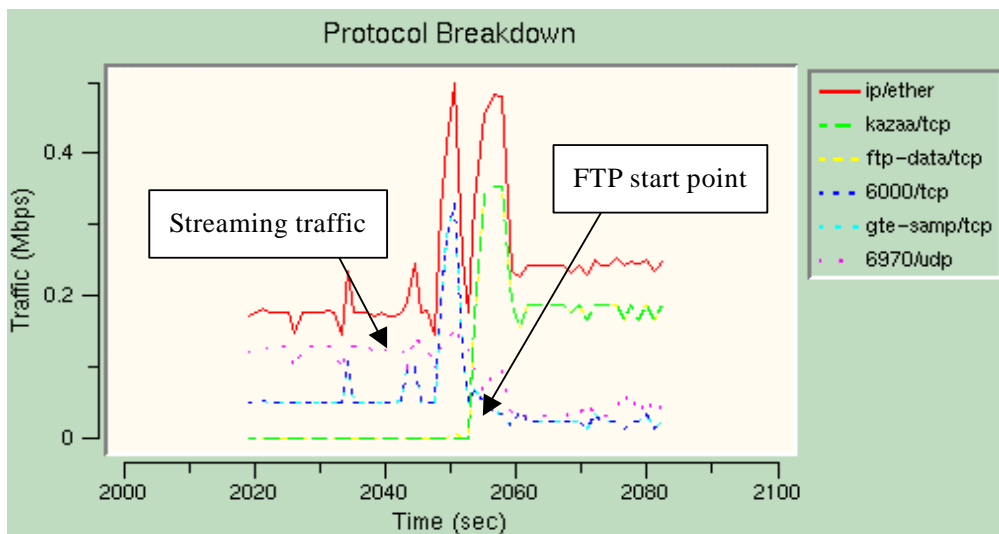
首先要說明的是，一個串流伺服程式的表現是與串流檔案壓縮方式息息相關的，因為壓縮格式的不同，對網路延遲、封包遺失及延遲的改變(Jitter)等情況的反映也不相同。因此這份並不嚴謹的測試僅可作為一個參考。我們的測試分為下面幾項，相關參數的調整是由 NIST 來達成。

調整項目	該項目參數值	結果	備註
網路頻寬 (Bandwidth)	10Kbits/sec 以上	可以順利播放	此項數值與檔案壓縮格式 直接相關
	10Kbits/sec 以下	畫面延遲，且頻寬越小越嚴重	
封包遺失率	0% - 30 %	5% 以下的遺失率尚可接受，畫	實際網路的封包遺失率並

(Loss rate)	(以 10% 為調整單位)	面品質隨遺失率變大而變差。	沒有這麼高。
網路延遲 (delay)	0 - 400 (ms) (以 100 為調整單位)	播放並無太大改變，然延遲過大時會產生畫面延遲。	此項與 Client 端緩衝區大小有關。緩衝區越大越能承受延遲。
網路延遲改變 (Jitter)	0 - 400 (ms) (以 100 為調整單位)	並無太大影響，然會造成封包丟棄的現象發生。	造成封包丟棄是因串流媒體內容是具時間相關性的，如果封包延遲抵達而影片已播放則變成無用，即被丟棄。

表五：DSS 測試項目及結果

另外，我們嘗試讓 TCP 與串流媒體流量共存，發現 DSS 對流量的控制相當不錯，並不會排擠 TCP 流量。圖十二為同時執行 FTP 及 DSS 串流媒體共存時流量圖，當 Ftp 啟動時，UDP 流量就被壓制，直到兩者維持平衡為止，並不會出現串流媒體流量強佔頻寬不放的情況發生。



圖十二：FTP 與串流媒體共存流量圖

六、結論

雖然目前尚無一個統一的標準訂定串流媒體的規格，串流媒體資訊在網際網路上的發展卻是不可避免的趨勢。在文中我們討論了目前串流媒體傳輸所使用的主要技術及實例探討，然這只是對整個串流媒體中最基本的“傳輸”所提供的解決之道而已。更複雜的是，我們該如何調和串流

媒體與既有的 TCP 流量？如何盡量用最少的頻寬去服務最多的使用者？如何充分的保護串流媒體資訊創作者的智慧財產權？在商業化應用時怎樣的付費機制才能為消費者與供應者提供最大的利益？還有其他許多的問題，這些都有賴進一步的研究討論來解決。串流媒體的前景很美好，然而路上的石頭還很多。

七、參考資料

1. Michael Mayer、Wilhelm Mohn、Christian Zabbal , “ PCs vs. TVs ” , The McKinsey Quarterly(2001/Q3) , McKinsey & Co
2. Interent.com , “ Study: Streaming Ad Market to Explode ” , streaming media news.06/6/2001 , <http://www.internetnews.com>
3. H. Schulzrinne, “ IP Networks ” , February 12 , 2000.
4. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: Atransport protocol for real-time applications." , January. 1996.
5. H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.
6. M. Handley, V. Jacobson, “ SDP: Session Description Protocol ” ,RFC2327, April 1998
7. WG, W3C, “ Synchronized Multimedia Integration Language ” , June 15, 1998.
8. RealNetworks Corp. , “ <http://www.realNetworks.com> ”
9. MicroSoft Corp. , “ <http://www.microsoft.com> ”
10. Apple Computer, Inc. “ <http://www.apple.com> ”
11. IceCast , “ <http://www.icecast.org> ”
12. Darwin Streaming Server, “ <http://publicsource.apple.com/projects/streaming/> ”
13. MPEG4IP , “ <http://mpeg4ip.sourceforge.net> ”
14. Obsequiem , “ <http://obs.freeamp.org> ”
15. Apple Computer, “ QuickTime Streaming Server ” , Apple Computer, Inc. , 2001
16. Apple Computer, “ Inside QuickTime:QuickTime File Format ” , Apple Computer, Inc. , March 27 , 2000
17. Apple Computer, “ QuickTime Streaming Server Modules ” , Apple Computer, Inc. , 2001