

# 以 RTP 為基礎之超媒體現場同步教學系統

陳恒佑

國立暨南大學資訊工程學系  
南投縣埔里鎮大學路 1 號  
hychen@csie.ncnu.edu.tw

賈言從

國立暨南大學資訊管理學系  
南投縣埔里鎮大學路 1 號  
g7213509@ncnu.edu.tw

陳景宜

國立暨南大學資訊管理學系  
南投縣埔里鎮大學路 1 號  
sp210086@tcts.secd.net.tw

王振宇

國立暨南大學資訊工程學系  
南投縣埔里鎮大學路 1 號  
mpower@ms30.url.com.tw

楊一正

國立暨南大學資訊工程學系  
南投縣埔里鎮大學路 1 號  
dreamery@ms36.hinet.net

洪政欣

國立暨南大學資訊工程學系  
南投縣埔里鎮大學路 1 號  
jshong@csie.ncnu.edu.tw

## 摘要

本論文，提出了一個使用 RTP 來同步呈現教學視訊內容、網頁內容與導覽事件的『超媒體現場同步教學』(Synchronized Hypermedia Embedded Live Lecture, SHEL) 系統。SHEL 的架構由三個主要的部分組成：

- (1) SHEL 記錄器：用來記錄具有時間性的影音教學內容和整個教學內容所發生的導覽事件。
- (2) SHEL 事件伺服器：用來接收、儲存和傳送 SHEL 事件。
- (3) SHEL 瀏覽器：用來同步播放影音教學內容、網頁教學內容及導覽事件。

對於多種媒體間的同步播放問題，我們以 RTP 為傳輸基礎，結合使用者對於多媒體同步的認知，來探討不同媒體之間在時間上同步的誤差範圍值。為了評估 SHEL 架構和同步模型的效能，我們使用 Java/JavaScript 和 C++ 語言實作一個具有即時性 (real-time) 的網路同步式影音教學系統，透過這個系統證明了同步超媒體現場群播的可行性[1]。

關鍵詞：RTP, Multimedia Synchronization, Distance Learning

## 1 簡介

近十年來，先進的信號壓縮技術已經使得在網路上傳輸多媒體影音資料成為可行，這種改變提供了遠距教學的新方法[5-7]。以現有的 WWW 技術而言，HTML 網頁已經可以呈現靜態文字及高解析的圖片更勝於呈現低速率的視訊。我們相信在 WWW 上遠距教學的最佳模式是配合使用動態網頁與 streaming 的技術來呈影音教學的內容[8-10]。

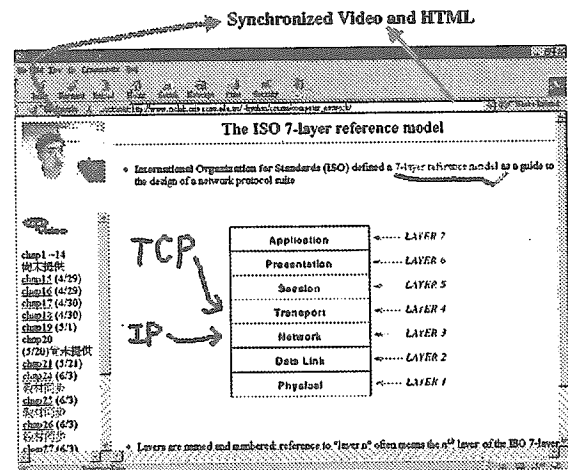
今天，已經有很多的方法可以處理音訊與視訊的同步問題，然而只有少數的研究在於整合多媒體與 WWW 上的導覽事件[11]。如果整個教學過程中的導覽事件（如：將內容反白、滑鼠移動、畫筆的寫字、捲軸的移動等等）配合網頁教學內容同步的播放，則遠距教學的功效將是更好的。

為了發展出可以同步呈現多媒體網頁及網頁導覽事件的遠距教學系統，我們以前研究的專案『以 Web 同步多媒體教學』(Web-based Synchronized Multimedia Lecture, WSML) 架構為基礎。這個系統會自動的記錄與傳播在教學過程中所發生的各種導覽事件。基於這個架構，我們已經實作一個雛型系統來同步呈現教學內容（如 Figure 1 所示）。

前面所提到的 WSML 系統是透過 TCP 來進行資料的傳輸。在評估之後，系統已經達到可接受的效能來呈現需求式 (On-demand) 教學內容給使用者，然而對於現場

(live) 的群播 (Multicast) 效果表現的不理想。主要的原因是 TCP 原本是設計來確保資料介於傳送者到接收者之間能夠以安全、可靠的方式來傳送，此種複雜控制反而會引起一個無法預期的延遲。故 TCP 的傳輸方式反而不適合多媒體資料要以現場實況的方式進行傳送。

Figure 1：WSML 呈現的影音與 HTML 同步教學



為了要改善我們系統現場實況傳輸的效能，我們認為 RTP 傳輸協定中所提供的同步。透過如此的安排使得在我們的系統中“連續型的多媒體資料”與“離散型的 WWW 導覽事件”可以有效的同步呈現在學生端。此外系統中對於不同型態媒體之間人為因素的考量 (human perception) 也是我們強調的重點。

## 2 SHEL 系統雛型架構

超媒體現場同步教學系統 (SHEL : Synchronized Hypermedia Embedded Live Lecture System) 主要由下列三個部分組成 (如 Figure 2) :

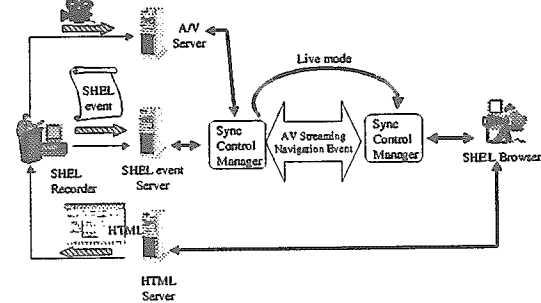


Figure 2：SHEL 系統雛型架構

- (1) SHEL 記錄器：用來記錄具有時間性的影像教學資訊和教學過程中內容導覽事件。
- (2) SHEL 事件伺服器：用來接收、儲存和傳送 SHEL 事件。
- (3) SHEL 瀏覽器：用來同步播放影音、網頁內容及導覽事件。

### 2.1 SHEL 記錄器

影音編碼器 SHEL 的主要功能是在教學期間記錄與 HTML 導覽事件有關的影音訊號。如 Figure 3 所示，記錄器是由(1)計時器(Timer)(2)影音編碼器(AV Encoder)(3)導覽事件記錄器(SHEL Event Server)所組成的。

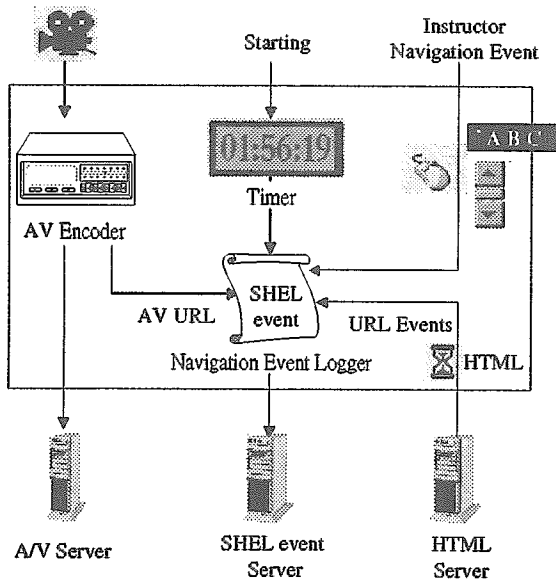


Figure 3：SHEL 記錄器的組成元件

- (1) 計時器：用來初始記錄程序，每次教師起動記錄器時，計時器會初始化且在同時成為 SHEL 系統的時間軸 (time-axis)，並自動地起動影音編碼器與導覽事件記錄器。
- (2) 影音編碼器：用來編碼與傳播影音教學，編碼器可即時將影音訊號編碼，傳送到影音伺服器儲存。此編碼後的內容可供現場群播教學或日後學生點播相關教學內容。
- (3) 導覽事件記錄器：處理影音資料、HTML 教學內容與導覽事件之間時間對應的問題。主要的導覽事件包含滑鼠事件、URL 更新事件、捲動軸與反白事件。這些事件的相關訊息都會透過導覽事件記錄器加以編碼處理後記錄在 RTP 資料封包中。例如，滑鼠事件包含相對時間點、滑鼠座標及移動的位移量。URL 事件則發生於老師輸入 URL、按下相關超鏈結載入網頁、載入上一頁或載入下一頁。導覽事件記錄器處理的範例如 Figure 4 所示。在 T1 時，AV URL1 與 HTML URL2 被記錄。在 T2 時，反白區的座標被記錄。在 T3 時，更新的 HTML URL2 將被記錄。在 T4 時滑鼠托曳的座標被記錄。在 T5 時，捲動軸的位移量被記錄。

### 2.2 SHEL 事件伺服器

SHEL 事件伺服器是負責接收、儲存和傳送 SHEL 事件，SHEL 事件伺服器從 SHEL 記錄器接收各種 SHEL 事件，這些影音資料和導覽事件透過 SHEL 同步管理機制

利用 RTP 協定與群播方式 (multicast) 傳送給學生端的 SHEL 瀏覽器。

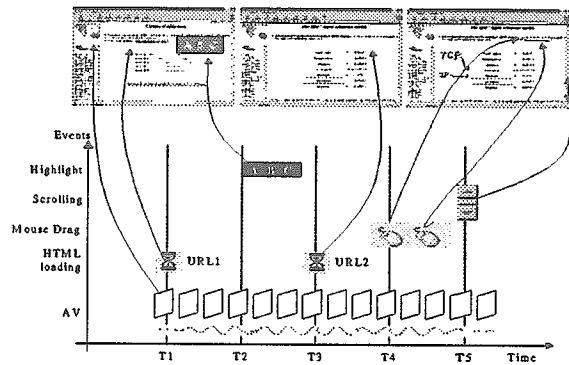


Figure 4：SHEL 記錄器處理事件的過程

### 2.3 SHEL 瀏覽器

SHEL 瀏覽器透過 A/V 資料與 HTML 導覽事件之間相對時間關係同步呈現出來 (如 Figure 5)。

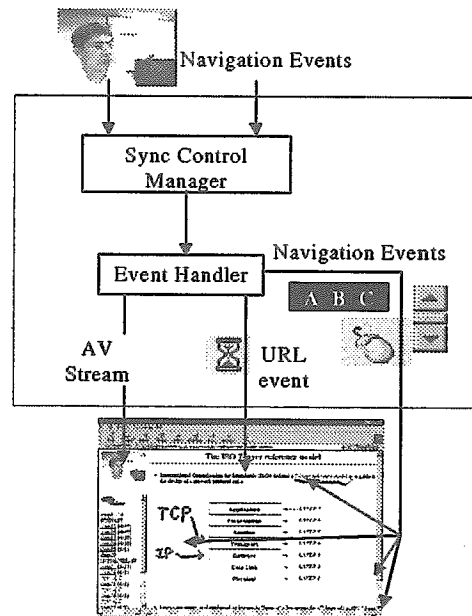


Figure 5：SHEL 瀏覽器的組成元件

SHEL 瀏覽器主要是由同步管理員、事件處理器、AV 播放器及 HTML 瀏覽器組成。SHEL 導覽事件如 Figure 4 所示，SHEL 瀏覽器將呈現以下的導覽事件：如 Figure 6 所示。

- T1: SHEL 瀏覽器從 SHEL 事件伺服器接收 AV URL 與 HTML URL，然後分別將它們傳送給影音播放器與 HTML 瀏覽器。影音播放器則開始播放影音教學，同時 HTML 瀏覽器也呈現出 HTML 之教學內容。
- T2: 事件處理器傳送反白事件給 HTML 瀏覽器去呈現反白區域。
- T3: 事件處理器傳送 URL 事件給 HTML 瀏覽器去載入新的 HTML 網頁。
- T4: 事件處理器傳送滑鼠拖曳事件給 HTML 瀏覽器去呈現滑鼠拖曳軌跡。
- T5: 事件處理器傳送捲動軸事件給 HTML 瀏覽器去調整捲動軸位置。

Figure 6：導覽事件引發的順序

### 3 行程管理與同步管理

在我們提出 SHEL 的架構中（如 Figure 7）有兩個重要的控制機制：一個是行程的管理；另一個是同步的管理。

#### 3.1 行程管理（Session Manager）

一個行程即為 client 端（在本文稱之為學生端）與 server 端（在本文稱之為老師端）之間的連線關係，亦即學生與老師之間的連線關係。對於行程註冊的動作（session enrollment），我們透過 TCP 來建立學生端和老師端之間的連線註冊訊息。而行程管理本身由兩個部分所構成，一個是類似 VCR 瀏覽的控制；另一個是行程錯誤管理。為了建立資料與控制連線，行程管理會分別配置兩個通訊埠號（port number）給 RTP 和 RTCP 作為資料與流量控制用。此外行程管理也會記錄加入現場教學的學生總數來進行回饋控制。其中類似 VCR 瀏覽的控制畫面包含了下列的功能：

- Play** 學生端通知老師端要開始教學的傳送。
- Pause** 學生端中斷所有教學的事件，並凍結播出。
- Resume** 學生端重新開始播放影音資料和載入在暫停後最近的 URL 事件。
- Stop** 學生端通知老師端中斷現場連線。另一方面，下列的功能是由註冊控制畫面所提供。
- Join** 分配系統資源，例如通訊埠號和成員狀態。
- Leave** 釋放由行程所分配的系統資源。

在瀏覽的過程中，透過 RTP 來將相對時間與資料傳送到學生端。在行程錯誤控制機制，我們透過 RTCP 來瞭解封包遺失與傳輸延遲的情形，以進行網路傳輸頻寬的控制。

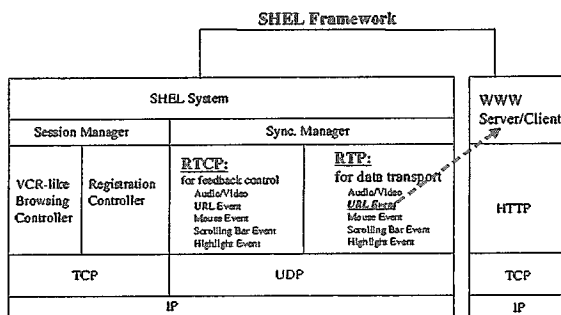


Figure 7：SHEL 架構中的 Protocol Stack

#### 3.2 同步管理（Synchronization Manager）

在同步管理中，我們的採取了“多執行緒”的架構，將媒體之間時間同步的計算工作分配給不同的行程[17]。Figure 8 說明了我們所提出架構的運作。在多執行緒中的行程包含有媒體行程（子行程）和控制行程（父行程）：(1)媒體行程是負責同一媒體種類之間（intra-media）時間同步運算處理與不同媒體種類之間（inter-media）同步的處理、使用者對於操作面板（如按下 pause，resume，stop）的時間處理。(2)控制行程是負責整體系統時間軸的處理與各種媒體行程的產生與結束動作。這架構時間和內容的同步管理是在 inter-media 同步機制中來完成。

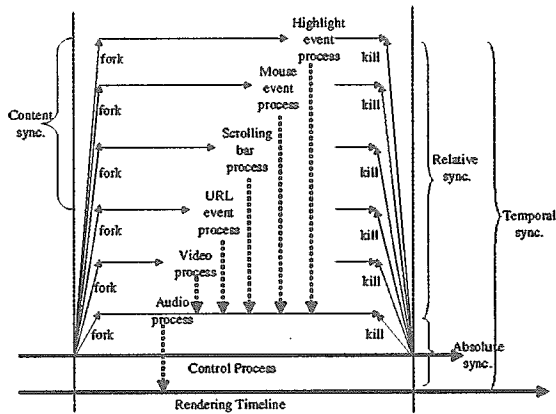


Figure 8：多行程同步的架構

Priority	Media Type	Sampling Frequency	Start in ms relative to audio	Sync type	Action taken based on Priority
1	Audio data	8K samples/second			Audio and Video are presented even URL event is lost.
2	Video data	30 frames/second	+/- 80 ms	Lip sync.	
2	URL event	2 events/second	+/- 2000 ms	Slide sync.	← If packet lost
3	Mouse event	30 events/second	-500 ms ~ +750 ms	pointer sync.	These events are all dropped for no association of correct HTML.
	Scrolling event	5 events/second	+/- 1000 ms		
	Highlight event	5 events/second	+/- 1000 ms		

Figure 9：SHEL system 中的 Media priority, sampling frequency 與 Synchronization type

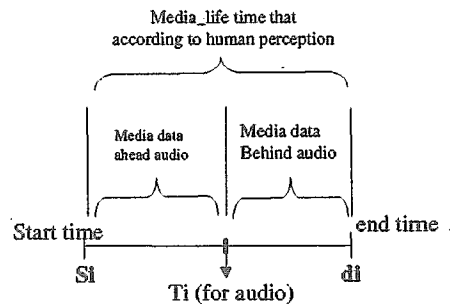


Figure 10：以聲音時間軸為標準時，考慮使用者對同步察覺時的媒體的存活時間。

#### 3.2.1 時間同步（Temporal synchronization）

對於所有影音資料和網頁導覽事件的同步都是以媒體間相對時間關係為基礎。在我們的架構中，對於聲音處理行程，資料的呈現必須配合系統時間和聲音封包到達學生端實際時間的差距值，從中計算出 SHEL 系統整體時間軸。我們稱這為“絕對同步時間”（Absolute Sync.

Time)。另一方面，根據使用者對不同媒體之間同步存在著可容忍誤差範圍值是相對於“絕對同步時間”所計算出，我們稱之為“相對同步時間”(Relative Sync. Time) [18, 19]。

### 3.2.2 內容同步 (Content synchronization)

對於教材內容同步呈現問題中，所有的導覽事件如滑鼠移動、捲軸、反白事件等必須在相對的網頁內容到達學生端時才能呈現出來。因此 URL 事件自然比其他網頁導覽事件來的重要。另一方面，雖然聲音與影像資料具有一定位元傳輸率的特性 (constant bit-rate transmission)，但影像資料卻相對於聲音資料來的龐大。此外，對於斷音的處理可以透過事先的暫存 (buffering) 來克服。因為聲音資料通常比影像資料來得小很多，自然所需的暫存記憶體也比影像來的少很多。因此在我們的架構中，內容的同步是建構在以聲音為時間主軸的原則上。若一個 URL 事件在傳輸中弄丟了，所有相對於這個網頁內容所產生的導覽事件將變的無意義了，而且必須被停止呈現出來。基於這樣的條件，我們定義了不同媒體之間優先權如 Figure 9 所示。在 Figure 9 中，影音資料和 URL 事件比其他的網頁導覽事件具有較高的優先權。

### 3.2.3 使用者對同步差異的考量 (Human perception consideration)

一般而言，使用者對於不同媒體間“不同步”情形 (out-of-sync.) 存有可容忍的差異範圍[15, 16]。對於影像播放時所發生一些干擾，則因為視覺暫留的特性，使得這樣的問題並不是很嚴重的。基於內容同步段落中所述的理由，我們以聲音時間做為 SHEL 系統播放時間的根據，計算出不同導覽事件相對於聲音同步可容忍的誤差範圍。我們稱為“媒體存活”時間 (Media lifetime)。根據現有論文的數據以及實驗的結果，我們提出一個“延遲或丟棄” (delay-or-drop) 的策略，來實現影音資料和網頁導覽事件之間的“媒體存活”時間的概念。如 Figure 10 所示，在使用者對同步察覺層次下，聲音與影像的同步誤差範圍值介於-80ms(聲音落後影像)與+80ms(聲音超前影像)之間。同樣的，在同步的問題中有關前景與背景聲音之間的同步範圍值是 (-500ms ~ +500ms)。在 SHEL 系統中，我們求得媒體可能的存活時間 (如 Figure 9)。Figure 11(a), 11(b)說明了以聲音時間為主，來判斷影像和相關網頁導覽事件是否在同步的範圍內。對於個別的事件，若播放時間落於媒體存活時間之內，則必須馬上播出。反之，如果播放時間是在適當間距之前或之後，則媒體資料必須等待 (delay) 或丟棄 (drop)。

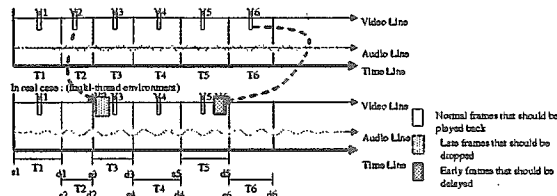


Figure 11(a): 媒體存活時間的概念與延遲/丟棄的法則

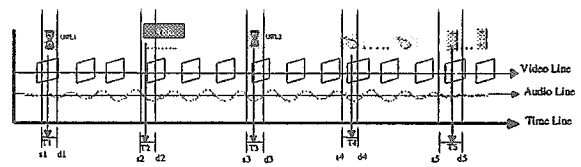
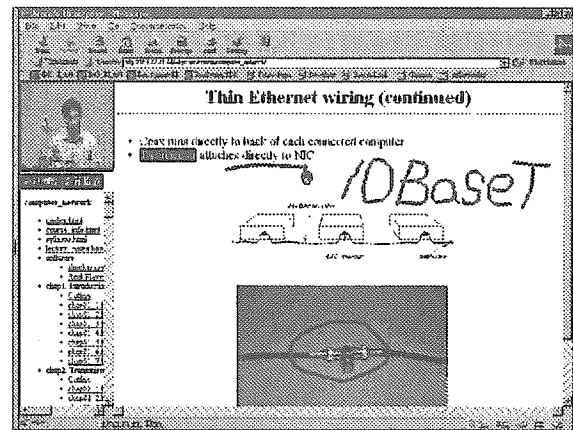


Figure 11(b): SHEL 網頁導覽事件的存活時間的推算

## 4 系統實作 (SYSTEM IMPLEMENTATION)

為了評估我們所提出 SHEL 的架構和多執行緒同步的可行性，我們透過一台 Compaq Pentium 266 雙 CPU 電腦，與數台 IBM Netfinity 3500 電腦進行系統測試。其中 Compaq 電腦被當作影音的編碼器而 IBM 電腦被當作學生端。其中系統測試環境是在 100-BaseT 的乙太區域網路上。RTP 傳輸與流量控制功能是以 C++ Plug-in 的方式結合 Netscape 瀏覽器。我們也在 WWW 瀏覽器中嵌入了許多的 Java Applet[12]和 JavaScript[13]，以提供使用者在 WWW 環境下的互動功能。Figure 12 所示為 SHEL 系統正在播放的畫面。

Figure 12: RTP-based Synchronization Hypermedia



Embedded Lecture System

由於 RTP 主要適用於連續型的影音資料上進行傳輸，故並未定義任何有關於間斷型資料型態如網頁導覽事件。我們利用相對時間的觀念，測試數種可能的導覽事件取樣頻率 (如 Figure 9)。為了記錄所有在老師端的事件，Figure 13(a)說明 SHEL 系統中用來記錄導覽事件被引發的時間標記、媒體型態、網頁發生的序號、所有事件發生的序號和事件訊息屬性等媒體訊息敘述。

### <Timing M\_type H\_serial E\_serial Attribution>

Figure 13(a): SHEL 中媒體訊息的語法敘述

透過 Figure 5 中的網頁導覽事件例子，Figure 13(b)說明了從 T1 到 T5 事件的紀錄格式。為了在學生端能呈現出這些事件，在 SHEL 系統中，Event Serve 會將這些語法敘述轉換成 SHEL 所定義的 RTP 封包中，如 Figure 14 所示。

<T1	video	0	1	rsp://www.mt.cse.nyu.edu/w/dep15fm>
<T1	html	1	2	http://www.nyu.edu/w/slides1.html >
<T2	highlight	1	3	(R, 21, 78, 47) >
<T3	html	2	4	http://www.nyu.edu/w/slides2.html >
<T4	mtag	2	5	(30, 40) >
<T4	mtag	2	5	(31, 41) >
<T4	mtag	2	5	(32, 41) >
.....				
<T4	mtag	2	5	(31, 40) >
<T5	scrolling	2	6	(0, 4) >

Figure 13(b): SHEL 中媒體訊息的範例

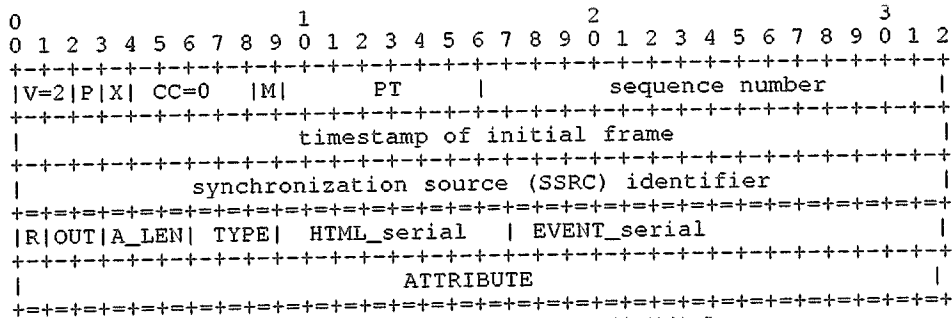


Figure 14: 網頁導覽事件中 RTP 封包的格式

對於我們定義的語法與 RTP 封包如下說明：

定義語法敘述 (Definitions for syntax expression)：

- **TIMESTAMP**：  
說明事件被觸發的相對時間值。
- **MEDIA\_Type**：  
說明所發生的媒體種類。可能是影音、網頁、反白、滑鼠移動或捲軸事件其中的一種。
- **HTML\_serial**：  
說明網頁更新事件所累積的次數。老師端會紀錄更新 URL 的總次數。在多點傳輸的環境下 (unreliable multicast)，當學生端收到一個 URL 事件，會利用此來判斷是否是收到新的網頁，或是重複收到先前的網頁資料。當學生端收到新的網頁內容，則必須在逾時的期間 (time-out interval) 內，盡快的送個回復訊息給老師端以確認。
- **EVENT\_serial**：  
這個序號每當老師觸發一個新的網頁導覽事件將會增加 1。對於學生端，可透過此機制配合 RTCP 中功能瞭解學生端接收資料的情形。
- **ATTRIBUTE**：  
詳細說明每個事件的內容值。

在 SHEL Event Server 中，會把以上的欄位編碼成 RTP 封包。

定義 RTP 中的欄位 (Fields Definitions in RTP)：

- 欄位 "R"：  
說明此封包是重新傳送上一次所發生的 URL 事件封包。
- 欄位 "OUT"：  
當將最新的一次 URL 封包送出去後，會引發一個計時器  $2^{OUT}$ ，來接收所有學生端的回應。此功能可以確保學生端有收到最新的 URL 事件封包。如果學生端收到此封包，可在  $2^{OUT}$  的期間內取得亂數，並在亂數時間內回應老師端它已經收到最新的 URL 封包。這個亂數可以避免和所有學生端的回復訊息發生碰撞 (burst)。
- 欄位 "A\_LEN"：  
使用 32 位元來說明屬性的長度。
- 欄位 "TYPE"：  
描述媒體的種類形式，其中並不包含影音資料。對於影音，SHEL 使用 RFC1889 中的定義來傳送影音封包。
- 欄位 "HTML\_serial"：
- 欄位 "EVENT\_serial"：  
參考之前在語法敘述中所定義的 "HTML\_serial" 和 "EVENT\_serial"。

- 欄位 "ATTRIBUTE"：  
描述由網頁導覽事件所產生的動作和相對的數值。

在我們所完成的系統中，我們發現會有一個很重要的問題，我們稱之為 "內容同步"，在 URL 事件和其他網頁導覽事件間，必須小心的處理。

1. URL 事件在網路傳輸中遺失了；
2. 在老師端已開始播出後，又有新的學生加入；
3. 在現場播出的過程中，有學生端進行暫停和繼續的動作。

關於第一點的情況，我們的解決方式已在前面文中說明了。至於情況二和情況三的問題核心是類似的，可以看成類似的問題來解決。當學生端繼續接收從現場送來的訊息，他會產生一個逾時的紀錄或新的學生端加入正在播出的時，就會重送最新的 URL 事件給學生端。因此網頁教學中其他網頁導覽事件就可被正確的被呈現出來。為了要確定影像或網頁導覽事件是否介於在媒體存活時間內。SHEL 系統中的 "絕對同步時間" 和 "相對同步時間" 可由如 Figure 15 的演算法中求得。

**In Control process:**

1. Send each dummy packet for all audio/video and navigation events to client site to find the maximum present time, and denote it as Initial Presentation Time- IPT, for all data.
2.  $Time\_sht = 0$ . // initial value

PlayOut <sub>m</sub> [I]	Playout time of ith media unit
SYS <sub>m</sub> [I]	System time at arrival of ith media unit
Delay <sub>ma</sub>	Media device delay, m → {audio; video}
Interval <sub>m</sub> [I]	
TS <sub>m</sub> [I]	Time index of ith media unit
SF <sub>mf</sub>	RTP timestamp of ith media unit
D <sub>m</sub> (I, J)	Media Sampling Frequency
J <sub>m</sub> [I]	The time difference between ith, jth media unit
StartTime <sub>m</sub> [I]	The jitter of ith media unit
EndTime <sub>m</sub> [I]	The start point in media lifetime for the ith media unit
	The end point in media lifetime for the ith media unit

**General subroutine: \*\*\***

```

Calculate_Interval()
{
  Dm(I, J) = ( Rm [J] - Sm [J] ) - ( Rm [I] - Sm [I] )
  Jm [I] = Jm [I-1] + ( 1 Dm [I-1, I] - Jm [I-1] ) / 16
  Intervalm[I] = (TSm[I] - TSm[0]) / SFm
  Return Intervalm[I], Jm[I]
}

```

```

in Media process:
For audio:
1.If (1st media unit)
    AssumePlayOutm[0] = IPT + Delayma
    Else
        Intervalm[I] = Calculate_Interval()
1. AssumePlayOutm[I] = PlayOutm[0] +
Intervalm[I]+Jm[I] + time_sht
2. Get the current System Time T for audio packet
present.
3. If (T > AssumePlayOuta[I])
    {
        time_sht=(T- AssumePlayOuta[I])+ time_sht
        RealPlayOuta[I] = T
    }
    Else If (T < AssumePlayOuta[I])
    {
        Wait for ( AssumePlayOuta[I] - T)
        RealPlayOuta[I] = AssumePlayOuta[I]
    }
Play Audio

For Video process and other Navigation Event:
1. If (the associative audio packet is present)
    {
        Calculate the media life time (StartTimem,
        EndTimem) by RealPlayOuta[I] of
        associative audio packet.
        Get the current System time T for
        video/navigation events media present
        If (for navigation events and URL was
        dropped)
        {
            Retransmission URL event for sever and
            drop other navigation events for client.
        }
        else
        {
            If (T < StartTimem[I])
                Wait for ( PlayOutm[I] -
                EndTimem[I])
            If (T > EndTimem[I])
                Discard and get the next proper
                packet
                Playback // between(startTime[I];
                EndTime[I])
        }
    }

```

Figure 15：根據絕對同步時間軸來計算媒體存活時間的演算法

我們使用 RealAudio 系統建立了一個原型系統。植入 Netscape Navigator 的 RealPlayer 是被當作 SHEL 瀏覽器中的影音播放器。在 SHEL 記錄器中 RealServer 是被當作影音伺服器且 RVEncoder 是被當作影音編碼器。影音訊號與 HTML 內容的同步功能則是由 RVEncoder 所提供的 "rmmerge" 來完成的。我們也向 Lucent Lab 合法申請了一套 RTP/RTCP API，應用在我們的系統之內。為了在瀏覽器內設計一個獨立平台的 SHEL 系統，我們選擇 Java/JavaScript 來完成這系統的互動實作。對即時連線技術而言，RealPlayer 能夠提供一些函式去呼叫 Java/JavaScript 來完成 HTML 與 AV 內容的同步呈現。在這種架構下，我們已經發現聲音封包與滑鼠指標兩者的重要不同，分析之後，在 Web 環境，若由瀏覽器立即畫出滑鼠移動的軌跡則不能提供較好的解決方法。我們的解決方法是使用動態 HTML 技術，例如：圖層與 CSS。

在瀏覽視窗中顯示滑鼠移動的軌跡，在這方法中，聲音封包的差值大約是 +/-2000ms，相關導覽事件的數據如 Figure 9 所示。

## 5 結論

在這篇文章中，我們提出—SHEL 系統，說明如何使用 RTP 傳輸協定配合絕對時間與相對時間的觀念將多媒體遠距教學中影音內容、HTML 網頁教材與瀏覽器中的導覽事件透過我們所提出的同步控制機制來達成網路同步教學。

SHEL 的架構由三個主要的部分組成：

- SHEL 記錄器：用來記錄暫時性的影像教學資訊和網頁教學內容的導覽過程。
- SHEL 事件伺服器：用來接收、放置和傳送 SHEL 事件。
- SHEL 瀏覽器：用來同步播放影音教學內容、網頁教學內容及導覽事件。

為了管理不同媒體間的同步播放問題，我們將使用者對於同步感覺的誤差範圍值納入考量，提出媒體存活時間 (media lifetime) 的同步模式。為了評估同步網路影音遠距教學系統 (SHEL) 架構和多重同步模型 (multi-sync) 的效果，我們透過 Java/JavaScript 來提供使用者互動介面，並將 RTP 網路傳輸與控制模式用嵌入的方式 (plug-ins) 結合瀏覽器。這個雛型系統證明了同步超媒體現場群播的可行性。

## 6 參考資料

- [1] Borko Furht, "Multimedia Systems: An Overview," *IEEE Multimedia*, Vol. 1, No. 1, pp. 47-59, Spring, 1994.
- [2] H.Schulzrinne, S. Casner, R.Fredrick and V. Jacobson, "RTP: A transport protocol for real-time applications," *RFC 1889*, Jan. 1996.
- [3] Braun. R., "Internet Protocols for Multimedia Communications." *IEEE Multimedia*, Vol. 44, pp. 74-82, Oct.-Dec. 1997.
- [4] Sridevi Palacharla, Ahmed Karmouch, Samy A. Mahmoud, "Design and Implementation of a Real-time Multimedia Presentation System using RTP," *IEEE*, 1997.
- [5] Ming-Chih Lai, et al., "Toward A New Educational Environment," the 4th International World Wide Web Conference, 1995.
- [6] Kiran R. Desai, Richard S. Culver, "Multimedia Hypertext on the WWW and its use in Education," *FIE' 98 Proceedings*
- [7] Elias N. Houstis, et al., "Internet, Education, and the Web," *Proceedings of WET ICE '96*
- [8] Zhigang Chen, et al., "Real Time Video and Audio in the World Wide Web," the 4th International WWW Conference, 1995.
- [9] James F. Kurose, Hu Imm Lee, Jitendra Padhye, Jesse Steinberg, "MANIC: Multimedia Asynchronous Networked Individualized

Courseware," Proceedings of Educational Media and Hypermedia, 1997.

- [10] Wei-hsiu Ma, Yen-Jen Lee, David H.C. Du, and Mark P. McCahill, "Video-Based Hypermedia for Education-on-Demand," IEEE Multimedia, vol. 51, pp. 72-83, Jan.-Mar., 1998
- [11] Ping-Jer Yeh, Bih-Horng Chen, Ming-Chih Lai, Shyan-Ming Yuan, "Synchronous Navigation Control for Distance Learning on the Web," Proceedings of fifth International World Wide Web Conference, May 1996
- [12] Java, <http://www.javasoft.com>
- [13] Javascript, <http://www.netscape.com>
- [14] RealAudio, <http://www.realaudio.com>
- [15] R. Steinmetz, "Human perception of jitter and media synchronization," IEEE Journal on Selected Areas in Communications, Vol. 14, No. 1, pp. 61-72, Jan. 1996.
- [16] I. Kouvelas, V. Hardman and A. Watson, "Lip Synchronization for use over the Internet: Analysis and Implementation," Proceedings of IEEE Globecom '96, Nov. 1996.
- [17] Heng-Yow Chen, and Ja-Ling Wu, "MultiSync: A synchronization model for multimedia systems," IEEE Journal on Selected Areas in Communications, Vol. 14, No. 1, Jan. 1996.
- [18] Nael Hirzalla, Ben Falchuk, Ahmed Karmouch, "A Temporal Model for Interactive Multimedia Scenarios," IEEE Multimedia, Vol. 23, pp. 24-31, Fall, 1995.
- [19] Yahya Y. Al-Salqan, "Temporal Relations and Synchronization Agents," IEEE Multimedia, Vol. 32, pp. 30-39, Summer, 1996.
- [20] Lucent Lab. RTP SDK <http://www.lucent.com>