

在網際網路上支援即時MPEG視訊的影像選擇性傳輸法之設計、實作與效能評估

李殷廣、孫雅麗
國立台灣大學資訊管理研究所
陳俊良
資策會資計處
臺北市羅斯福路四段一號
TEL:(02)3630231 EXT.2811
EMAIL: r4725014@im.ntu.edu.tw

論文摘要

隨著Internet這個全球最強勢的資訊網路之風行，人們對在Internet上即時的連續媒體傳輸，也愈來愈有強烈的需求，例如隨選視訊（video-on-demand）、電子商務的線上3D型錄購物（catalog）等等。但是想要在以TCP/IP傳輸技術為基礎的Internet上，達到即時連續媒體的傳輸，尚有重重障礙，主要有網路頻寬不足、網路傳輸狀況難以預測、以及TCP/IP的傳輸技術並不支援QoS（quality of service）等三大因素。

本論文即是想要在上述的限制下，發展出一套傳輸架構，適合MPGE視訊在Internet上即時地傳輸與播放。本研究設計並實作了在網際網路上的即時MPEG視訊系統。這個系統中video server與client之間以Internet Engineering Task Force所制定的RTP作為資料傳輸的通訊協定，即時地傳輸MPEG視訊資料並播放。基於維持播放品質的前提下，在server端設計了『以速率為基礎的MPEG影像之選擇性傳輸架構（Rate-based Selective MPEG Video Transfer Scheme）』，能夠在傳輸的過程中，根據客戶端主動回傳的資訊，動態地調整傳輸速率，以便在即時傳輸的要求下，仍舊能適應網路狀況與並維持播放品質。server應用解析MPEG視訊產生的Meta Data來作傳輸的控制。receiver採用資料先存（data preloading）的政策，以降低傳輸品質的下降所引起的播放品質惡化。

在深入評估過本系統的效能後，若欲保證MPEG視訊播放的品質良好，我們得到重要結論有三：第一、重要的資料需要可靠的傳輸，以維持播放的品質，例如MPEG video stream中的Sequence Header、GOP Header、I-frame等；第二、控制封包需要即時且可靠的傳輸，以便能夠迅速有效的達到控制的目的；第三、先存資料量與初始傳輸速率必須謹慎地協調出合適的值，才不致惡化影像播放的品質。

關鍵字：RTP，即時MPEG視訊，選擇性傳輸架構，資料先存

1. 緒論

隨著網際網路（Internet）的地位愈來愈重要，人們在網際網路上即時（real-time）傳輸並播放連續媒體（continuous media）的需求也日益增加。例如隨選視訊（video-on demand）、電子商務之多媒體型錄（multimedia catalog）等等。但是在以TCP/IP傳輸技術為基礎的網際網路上，想要達到即時性的目標，遭遇到的困難重重，主要有網路頻寬不敷使用、網路傳輸狀況難以預測、以及TCP/IP的傳輸技術並不支援QoS（quality of service）等三大因素。

本論文即是想要在上述的限制下，發展出一套傳輸架構，適合MPGE視訊在Internet上即時地傳輸與播放。本研究的目的有三：第一、實作一套以RTP為資料傳輸通訊協定，MPEG視訊為連續媒體的網際網路即時視訊播放系統；第二、以此真實的系統作實驗，從事網路兩節點間即時視訊的傳輸與播放，來量測網路傳輸的狀況與播放的品質，並期望由此實驗的結果來推測影響即時性連續媒體傳輸與播放之因素，以設計出合適之傳輸架構；第三、設計出能實現傳輸速率控制的速率調整機制。

為了達到這些目的，本研究設計了網際網路即時視訊播放系統的主從式（client-server）架構，如圖3-1。扮演伺服器（server）角色的視訊伺服器（video server），與扮演客戶端（client

的視訊播放器 (video player) 中間以 IETF (Internet Engineering Task Force) 制定的 RTP [14] 作為即時資料傳輸的通訊協定。在這個架構下，視訊已經事先以 MPEG [7] 的格式編碼過並儲存起來，待客戶端提出要求，視訊伺服器再至儲存體中讀取出視訊資料並傳輸。RFC 2038 [6] 依據 Application-Level Framing [4] 的概念，根據資料的特性訂出 MPEG 視訊流 (video stream) 的切割規則，視訊伺服器則使用這套規則，並佐以事前處理產生的 Meta Data 來進行資料封包化的工作。這種以網路作即時視訊播放的系統會遭遇到網路上資料傳輸可能發生的封包 (packet) 延遲、遺失，以及客戶端 CPU 處理能力不足等問題，而需要作傳輸的控制。在伺服器端設計了『以速率為基礎的 MPEG 影像之選擇性傳輸架構 (Rate-based Selective MPEG Video Transfer Scheme)』，能夠在傳輸的過程中，根據客戶端主動回傳的資訊，動態地調整傳輸速率，以在即時傳輸的要求下，仍舊能適應網路狀況與使用者需求。

本論文以下章節的架構為，第二章介紹作為資料傳輸之通訊協定的 RTP，以及本論文研究的連續媒體 MPEG 視訊標準，並簡介其他有關於網路上即時連續媒體傳輸的研究。第三章為本研究網際網路即時 MPEG 視訊系統的設計與實作。第四章為以實作出來的系統進行實驗，以評估系統的效能。第五章對本研究的發現作一總結，並提出未來可以作更進一步的研究之方向。

2. 背景知識與相關研究

與本研究相關的背景知識包括 RTP 這一個為傳輸即時性資料而發展出來的通訊協定，以及 MPEG 這個影像壓縮的國際標準。本章對這兩個標準做一簡單扼要的介紹，並佐以一些有關於在網路上傳輸即時性資料的相關研究，以提供對本研究欲解決的問題與設計的方法一個預先的認識與了解。

2.1 RTP (Real-time Transport Protocol)

RTP [14] 由 Internet Engineering Task Force (IETF) 的 Audio/Video Transport Working group 發展出來，目的在於提供具即時性的資料傳送之服務 (delivery services)，例如互動式的音訊跟視訊等。RTP 根據 application level framing (簡稱 ALF) 與 integrated layer processing (簡稱 ILP) 的原則 [4] 來設計，所以 RTP 在針對某特定應用程式提供所需要的資訊方面，試圖做到具有延展性，而且通常不另外實作成一個單獨的通訊協定層級 (protocol layer)，而跟應用程式的處理整合在一起。

RTP 與其他通訊協定層級的關係如圖 2-1 所示。在網際網路上，RTP 通常架構在 UDP 之上運作，利用 UDP 提供的傳輸服務。然而 RTP 跟底層的網路技術或傳輸協定是互相獨立的，所以它也能在 ATM 的架構或是其他種技術的網路上執行。無論如何，RTP 都是跟應用程式整合在一起。

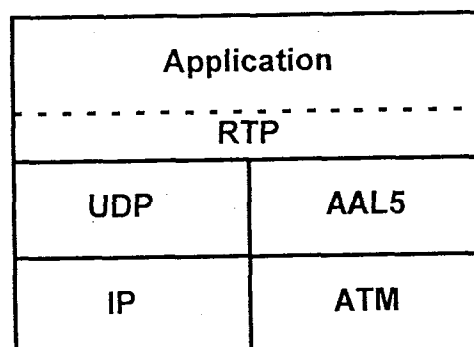


圖 2-1：RTP 與其他通訊協定的層級關係

RTP分成兩個部份：資料傳輸的通訊協定（RTP Data Transfer Protocol），與控制的通訊協定RTCP（RTP Control Protocol）。資料傳輸協定只單純地處理實際的連續媒體資料之傳輸，RTCP則提供了多種控制封包（control packet）來監測（monitor）即時資料的傳輸。RTCP封包定期地在建立起RTP連結的各個實體之間互相交換，以提供最新的資訊。在這些訊息之中，SR（sender report）與RR（receiver report）包含了重要的回饋資訊，例如封包的遺失比率、interarrival jitter 等等，以反應目前網路傳輸的狀況。然而，這些控制資訊的運用完全取決於應用程式的需求，RTP並沒有硬性規定這些資訊該如何被使用。

總而言之，RTP提供了一個具彈性、有助於處理上的效率、利於即時性資料傳輸的通訊協定架構。本研究的傳輸架構即是利用RTP依據的哲學跟提供的功能發展出來的。

2.2 MPEG

MPEG 是 "Moving Pictures Experts Groups" 的簡稱。它是ISO制定的國際標準 [7,8,9]。它的主要目標在於將影像編碼（encoding）壓縮，用較少的位元來儲存。現在有MPEG-1 [8]與MPEG-2 [9]兩種標準存在，這兩種標準訂定的目的不太一樣，但是以編碼技術的角度來看，兩者依據的演算法大同小異，因此MPEG-2可向後與MPEG-1相容（backward compatible）。

一段MPEG視訊是由許多個圖片（frame 或 picture）所組成。在MPEG的標準裡，壓縮的演算法將這些圖片分成三類：I, P, B-frame。I-frame 使用intracoded的編碼方式壓縮而成，根據它內部擁有的資料即可解碼（decoding）出完整的圖片。P-frame 用到了向前預測（forward prediction）的壓縮技巧，在解碼時它必須參照前一個I-frame 或P-frame 的資料才有辦法完整地重建圖片。B-frame 則利用了雙向預測（bidirectional prediction）的技巧，它必須同時參照到前面一個與後面一個的I-frame 或P-frame 的資料來解碼。I-frame與P-frame這二種會被其他圖片解碼參照到的圖片，統稱為參照圖片（reference frame）。圖片編碼及解碼的參照相依（reference dependence）關係示意如圖2-2。

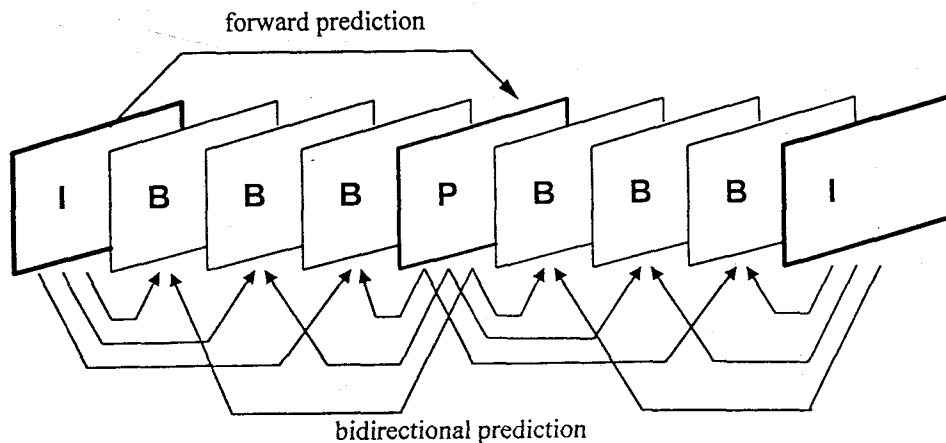


圖2-2：MPEG圖片編解碼參照相依關係

一段MPEG視訊流（video stream）中，可以由不同組合樣式的I, P, B圖片所構成，也就是並沒有限制這些圖片出現的次數或排列的格式，視訊可以全部由I-frame組成，也可以由I, P, B三種圖片用任意的排列組合方式形成。但是在現實的應用上，為了編碼方便起見，常會有固定的圖片組成型式（pattern），例如圖2-2中，圖片組成以一張I-frame、三張B-frame、一張P-frame、三張B-frame這樣的型式重複出現，我們將這樣的圖片組成型式以兩個參數表示：I-distance與P-distance。I-distance為連續兩張I-frame的距離，也等於連續兩張I-frame之間其他的

圖片數再加上一。P-distance為P-frame與其上一張參照圖片的距離，會等於P-frame與其上一張I或P-frame之間其他的圖片（皆為B-frame）總數再加上一。因此以圖2-2來看，I-distance為8，P-distance為4。

2.3. 相關研究

有一些相關的研究與論文出爐討論在網際網路傳輸即時性的連續媒體資料的議題。以下則針對這些研究，所欲解決的問題，以及尚未考慮到的問題與限制，做一簡單扼要的說明。

2.3.1 IVS

IVS[16] 是一個在網際網路上的視訊會議系統，它利用在UDP上的RTP提供音訊與視訊的傳輸。IVS 在視訊方面它採用了 H.261 的影像壓縮標準。為了在像網際網路這種packet switched的網路上傳輸H.261的視訊資料，IVS的作者們提出了一個封包化的架構（packetization scheme），並發展出封包遺失回復（packet loss recovery）與流量控制（flow control）的架構。

IVS在封包遺失的錯誤回復及流量控制方面，是採取調整編碼器（encoder）的參數來達成。當接收端發現有封包遺失，它會馬上送NACK（negative acknowledgment）訊息回發送端，發送端就會針對圖片的遺失部份重新編碼、傳送，來做到更新（refreshment）錯誤的效果。IVS包含了一個回饋控制機制[1]，利用接收端的回饋訊息，來判斷網路的狀況，並調整軟體編碼器的參數，改變產生的壓縮資料量，來控制資料輸出的速率。不過方法，若是降低資料量，則會有圖片品質變差的取捨問題（trade-off）。

2.3.2 vic

在 [10] 中，作者們實作了一個RTP的視訊工具 vic，並設計了具有彈性的系統架構（稱作 conference bus），能夠容易地整合視訊會議的其他各項工具，例如聲音、白板、及協調（coordination）等工具。此外vic提供了可擴充的使用者界面，並可以支援多種不同的壓縮演算法。vic這個軟體在H.261的視訊傳輸方面，達到比IVS更高的品質。

雖然vic號稱實作了RTP用來傳輸，但是卻沒有善加利用到RTP中控制部份的精神。在追蹤過vic原始程式碼（2.8版）後，發現vic雖然有傳送RTCP的控制封包，但它只利用到敘述性的資料，例如SDES（source description）與BYE（goodbye）之RTCP封包，對於記錄網路狀況的SR（sender report）與RR（receiver report），vic並未針對其中包含的回饋資料作任何的回應處理動作。

2.3.3 Vosaic

Vosaic（video mosaic的簡稱）[3]擴充在WWW上傳輸視訊的能力。它根據RTP的哲學制定了VDP，作為WWW上即時視訊傳輸的通訊協定。VDP最重要的精神在於server依據client回饋回來有關網路擁塞（network congestion）與CPU負荷（CPU load）的控制資訊，動態地去調整傳輸的速率。

VDP的回饋訊息每三十秒傳送一次，此訊息內很簡單地包括了圖片丟棄率（frame drop rate）與封包丟棄率（packet drop rate）兩種資料。圖片丟棄率是client收到了圖片，卻因CPU的能力不足無法即時解碼而必須丟棄圖片的比率，是用來衡量client端CPU負荷的程度。封包丟棄率則指因為網路擁塞而被丟棄的封包比率。server端根據收到的回饋訊息，以VDP的適應演算法

(adaptation algorithm) 來動態地改變傳輸的速率。VDP還包含的一個需求重送演算法 (demand resend algorithm)，client須有能力判斷需要重傳的圖片，並主動要求server重傳，來維持播放品質。不過以三十秒這樣長的一段時間才傳送一次回饋訊息，能否即時地反映網路傳輸與視訊播放的狀況，倒是值得商榷的。

3. 網際網路即時MPEG視訊系統之設計與實作

3.1 網際網路即時MPEG視訊系統之設計

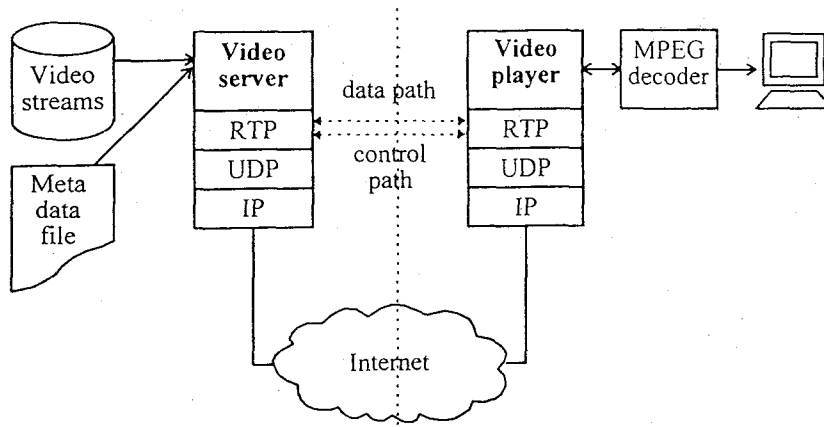


圖3-1：系統架構圖

3.1.1 概述

本研究設計了網際網路即時MPEG視訊播放系統的主從式架構，如圖3-1。視訊伺服器與客戶端的視訊播放器中間以RTP作為即時資料傳輸的通訊協定。在這個架構下，視訊已經事先以MPEG-1的格式編碼過並儲存起來，待客戶端提出要求，視訊伺服器再至儲存體中讀取出MPEG視訊資料，並透過資料途徑 (data path) 傳輸。視訊播放器接收到資料封包後將之儲存在緩衝區 (buffer) 內，並重組為一張張的圖片 (frame)，交由MPEG軟體解碼器 (decoder) 處理並播放。這種在網際網路上從事即時視訊播放的系統，可能會遭遇到網路上資料傳輸發生壅塞而使封包延遲、遺失，客戶端也有可能CPU處理能力不足無法即時解碼與播放，因此需要作傳輸的控制。客戶端依據資料接收的情形及處理的狀況，主動將回饋資料經由控制途徑 (control path) 送回伺服器端，伺服器再根據這些回傳的資料，以本研究設計之『以速率為基礎的MPEG影像之選擇性傳輸架構 (Rate-based Selective MPEG Video Transfer Scheme)』來動態地調整傳輸速率，以便能夠在即時傳輸的要求下，仍舊能適應網路狀況與使用者需求。

為了達到即時且彈性地傳輸MPEG視訊的目的，視訊伺服器的功能，包括了 (1) 切割視訊流做封包化的工作；(2) 根據客戶端的回傳的控制封包動態地調整傳輸速率。

RFC 2038 [6] 中描述了MPEG視訊流封包化 (packetization) 的切割規則:

- (1) Sequence_Header必須放置在RTP payload的最前端。
- (2) GOP_Header必須放置在RTP payload的最前端，或者接在Sequence_Header的後面放置在RTP payload中。
- (3) Picture_Header必須放置在RTP payload的最前端，或者接在GOP_Header的後面放置在RTP payload中。

這些規則將地位重要的header完整地安排在單一封包內，避免被分割至不同的封包，降低封包遺失對即時播放的影響。

3.1.2以速率為基礎的MPEG影像選擇性傳輸架構 (Rate-based Selective MPEG Video Transfer Scheme)

為了解決因網路的擁塞、封包遺失、或是客戶端CPU處理能力的不足，而引起的連續媒體播放品質不佳之問題，伺服器端必須能夠動態調整傳輸速率。在IVS這個系統，從發送端傳輸至網路的資料速率，是靠調整軟體編碼器的參數，讓壓縮現場視訊 (live video) 產生的資料量降低，而達到流量控制的目的。但是在我們設計的系統下，視訊已經事前經過MPEG演算法壓縮過後儲存起來，將MPEG視訊先行解壓縮再以新參數重新壓縮成較小的資料量來傳輸並不可行。因此在視訊播放必須平順、傳輸流量需要控制的限制下，本架構利用捨棄一些較不重要的圖片來達到調整傳輸速率的目的。

以一個例子，看看這種方法在降低速率方面，能夠達到多少效果。考慮圖片組成型式 (pattern) 為每隔三張圖片有一張為P-frame，每隔十五張圖片有一張為I-frame的MPEG視訊流 (IBBPBBPBBPBBPBBPBI)，I、P、B-frame的平均大小分別為150、50、20 Kbits。假若我們決定捨棄所有B frames不傳，則調整過的傳輸速率跟原速率的百分比計算如下：

$$(150 + 50 \times 4) / (150 + 50 \times 4 + 20 \times 10) = 63\%$$

若我們決定捨棄所有B frames跟P frames不傳，則

$$150 / (150 + 50 \times 4 + 20 \times 10) = 27\%$$

根據這種捨棄圖片達到的速率調整效果，設計了線性選擇捨棄演算法 (Linear, Selective Skipping Algorithm) 如下：

```
AdjustRate(X)
if the video stream has fixed pattern then
R1 = rate of skipping all B-frames
R2 = rate of skipping all P and B-frames
if X >= R1 then
skip (1-X) B-frames every (1-R1) B-frames
else if R2 <= X < R1 then
skip all B-frames, and skip (R1-X) P-frames every (R1-R2) P-frames
else
skip all P,B-frames, and skip (R2-X) I-frames every R2 I-frames
```

X 是調整後的傳輸速率，為傳入的參數。在MPEG視訊流具有固定的圖片組成型式 (fixed pattern) 情況下，本架構根據欲達到的效果X，按照比例來捨棄不同種

類、不同張數的圖片。我們優先考慮捨棄對播放影響最低的B frames，因為它們不會被其他圖片在解碼時參照到，而對解碼最為重要的I frames 則盡量不要捨棄。

3.1.3 Meta Data

為了視訊伺服器切割封包及選擇性傳輸的需要，必須知道視訊流的結構中各資料區塊的邊界，這工作需要靠解析 MPEG 視訊流的結構來達成。因此，我們發展出一個 MPEG 解析器（MPEG parser），預先解析 MPEG 視訊流而產生相關的 Meta Data，並將這些資訊記錄儲存成 Meta File，供未來傳輸時參考的依據。Meta Data 應該包含下列種類的資訊：

(1) 一般資訊

包含了圖片的解析度（resolution）、frame rate、以及bit rate。

(2) 圖片組成型式（pattern）

圖片組成型式是MPEG視訊流中圖片的組合方式，會跟選擇性的傳輸架構有關。有幾個參數用來描述MPEG 視訊的圖片組成型式：

I_distance：描述連續的兩張 I-frames 中間隔了幾張其他的frames。

P_distance：描述一張 P-frames 與其上一張 I-frame 或 P-frame，或是與其下一張 I-frame 或 P-frame 中間隔了幾張其他的frames。

I_number：整段視訊流中 I-frame 的個數。

P_number：整段視訊流中 P-frame 的個數。

B_number：整段視訊流中 B-frame 的個數。

(3) 圖片大小（frame size）

依照選擇性傳輸架構來調整傳輸速率，我們必須要能夠估計由捨棄某些圖片不傳而能達到的效果，因此需要知道各種圖片的平均長度。

Avg_I_framesize：I-frame 的平均長度。

Avg_P_framesize：P-frame 的平均長度。

Avg_B_framesize：B-frame 的平均長度。

Avg_Framesize：不分類型，一張 frame 的平均長度。

(4) 邊界（boundary）

MPEG 視訊各項資料區塊的邊界範圍，在封包化及選擇性傳輸時，是一項極為重要的資訊。我們利用多維的 tuple 來描述一個視訊流各個重要資料區塊的邊界資訊。目前共定義了六種 tuples—S、G、I、P、B、E tuples，分別來描述Sequence_Header、GOP_Header、I_Picture_Header、P_Picture_Header、B_Picture_Header、Sequence_End等邊界資訊。一個 tuple 的格式包含一個固定為四維的資訊 [type, offset, header_length, extension_length]，I、P、B等 tuples後面還會接續一個以上的slice_length。type 指示了這個 tuple 所要描述的資料區塊為何，即是上述的S、G、I、P、B、E中的一種。offset 標示了此資料區塊在整個視訊流中的絕對位置。header_length 則是資料區塊中header包含的位元組數，extension_length是指接續在header後的extension data之長度，slice_length記錄了每一個slice的長度。根據這項邊界的資訊，視訊伺服器能夠隨意地找到某特定資料區塊的位置跟大小，並依此作封包切割與選擇性傳輸的工作。

3.1.3資料先存政策 (Data Preloading)

爲了維持MPEG視訊的播放品質，降低網路傳輸品質不佳帶來的影響，我們設計以緩衝區 (buffer) 儲存一定的資料量，以便在網路傳輸品質突然變差下 (例如封包大量遺失)，播放器還能夠有資料可以繼續播放不致中斷 (playback interrupt) 影響到播放品質，另一方面又可以在緩衝區資料耗盡之前，做出適當的補救措施，例如資料重傳等等。於是在MPGE視訊資料的傳輸開始後，希望在播放器真正到緩衝區拿取資料開始播放之前，能夠先存滿一些資料，以緩衝資料傳輸狀況不佳帶來的播放品質下降。我們將此政策稱之爲資料先存 (data preloading)。

3.2系統實作

本研究在個人電腦的Windows 95作業平台下實作了三個程式—MPEG 解析器 (MPEG Parser)、MPEG視訊播放器 (Video Player)、以及視訊伺服器 (Video Server)，功能如3.1節所述。在MPEG視訊流的解析部分，修改自MPEG Software Simulation Group (MSSG) 提供之MPEG-2軟體解碼器 [19]，因此本程式能夠解析MPEG-2與MPEG-1這兩種格式的視訊流。而MPEG視訊的解碼部分，則是拿取美國UCB大學發展的MPEG-1軟體解碼器[18]。此解碼器應用了程式撰寫的一些最佳化技巧 [18]，讓解碼的效能 (performance) 能夠符合即時性的要求。此外，我們擴增了負責資料傳輸的RTP之功能，描述如下。

3.2.1 RTP功能的擴增

RTP提供了基本的通訊架構，但是爲了配合即時MPEG視訊傳輸的需要，我們擴增了RTP的功能。在原本RTP的通訊協定中，定義了RTP (RTP Data Transfer Protocol)，與RTCP (RTP Control Protocol) 兩個部分，分別傳輸連續媒體資料及控制資訊，但是在RTP的規格中 (RFC1889 [14])，爲了不讓RTCP控制封包佔用太多傳輸的頻寬，而限制平均5秒才能傳送一個RTCP封包。這種限制，阻礙了我們設計的傳輸架構中欲達到即時控制的目標。因此我們設計將即時性的控制封包透過具即時性的RTP資料途徑傳送。此外爲了封包重組成圖片的效率，我們利用RTP header extension的能力，在RTP的資料封包內附加圖片序號 (frame sequence number)、圖片內的封包序號 (packet sequence number within a frame)、以及圖片包含的封包數 (number of packets of the frame) 來快速地重組封包成爲圖片，並便於判定出圖片遺失與封包遺失的狀況。我們定義之4 bytes的RTP extended header如圖3-2。

RTP header	RTP fixed header (12-72 bytes)
	RTP extended header (4 bytes)
RTP payload	MPEG video-specific header (4 bytes)
	MPEG video data (variable)

(1) RTP資料封包

RTP header	RTP fixed header (12-72 bytes)
	RTP extended header (4 bytes)
RTP payload	control parameters (variable)

(2) RTP即時控制封包
圖3-2：RTP extended header格式

這兩種extended header以一個T bit來區別此封包是RTP資料封包還是RTP即時控制封包。在RTP資料封包中的extended header裡除了T-bit外，尚包含了圖片序號（frame sequence number）、圖片包含的封包數（number of packets of the frame）、以及圖片內的封包序號（packet sequence number within a frame）三個欄位。這些資訊幫助封包重組為MPEG視訊的圖片。

RTP即時控制封包中的extended header裡，包括5 bits的控制動作（control action）、10 bits的參數長度（parameter length）、以及16 bits隨不同控制動作之封包而定義的欄位。控制動作定義了一個RTP即時控制封包的類型，目前設計了播放（PLAY）、停止（STOP）、速率（RATE）以及完成（FINISH）等四種類型的控制動作。前三者定義了客戶端送到伺服器端的即時控制封包類型，功用分別為要求伺服器開始播放、停止播放、以及調整速率。FINISH為伺服器端傳給客戶端的即時控制封包類型，用來告知客戶端資料傳輸完成的情形。5 bits的控制動作欄位留下了未來增加控制功能的空間，例如可以增加定義快速前進（fast forward）、快速後退（fast backward）、暫停（pause）等一般在視訊播放時使用者會使用到的動作。

不同類型的RTP即時控制封包會攜帶不同的控制參數（control parameter），例如開始播放（PLAY）之RTP即時控制封包會包含欲播放的video clip之檔案名稱，而這些參數的位元組數（bytes）就是以參數長度這個欄位來記錄。如果控制參數資料量少於十六位元，則不需放至RTP payload裡，而可以利用extended header中最後十六位元的欄位，例如速率調整（RATE）之RTP即時控制封包，只需攜帶欲調整的圖片傳輸速率值（單位為frames per second），因此這個參數直接可以放至extended header中。而傳輸完成（FINISH）之RTP即時控制封包，包含傳輸結束的理由碼（reason code），由伺服器端告訴客戶端傳輸是否正常結束，這個不會超過十六位元的理由碼也可以直接放至extended header中可利用的十六位元之欄位。停止播放（STOP）並未攜帶任何控制參數，只是客戶端單純地要求伺服器直接停止目前正在傳輸的MPEG視訊。

本實作中一個RTP封包的大小最大設定為1024 bytes，扣除最小12 bytes的RTP fixed header、各4 bytes的RTP extended header與MPEG video-specific header，一個RTP資料封包最多可攜帶1004 bytes的視訊資料。實際上產生的封包大小視RTP payload攜帶的資料量而定，在一張圖片的最後一個封包，其大小通常小於1024 bytes。RTP即時控制封包由於攜帶的資料不多，通常封包

3.2.2 最近參照圖片 (Most recent reference frame) 的解碼政策

在MPEG的標準裡，P、B-frame必須參照緊鄰於它之前或之後的參照圖片 (reference frame) 才能正確解碼。但是在網路傳輸的過程中，這些參照圖片有可能遺失，而使其他需依賴他的圖片無法正常解碼，我們將這些被影響到的圖片稱之為孤兒圖片 (orphan frame)。若我們堅守MPEG標準不對孤兒圖片解碼，則會導致這些圖片即使完整接收到卻不會使用到。這一方面造成頻寬的浪費，另一方面影響到播放的順暢性，例如圖片速率 (frame rate) 為30 fps (frames per second) 的MPEG視訊，若其I-distance為15，則遺失掉一個I-frame會造成十四張孤兒圖片，而使得視訊畫面暫停半秒鐘，影響到播放的順暢性 (smoothness)。因此我們希望孤兒圖片還是能夠解碼，於是我們採取的策略為『投靠近親』，也就是孤兒圖片參照最接近它的上一張參照圖片 (most recent reference frame) 來解碼。此策略的好處是能夠善加利用接收到的完整圖片，並讓播放的視訊畫面能夠較為順暢。但是也有代價，就是參照不正確的解碼會讓圖片產生雜訊 (noise)。在視訊內容場景變化不大，I-distance與P-distance不大的情況下，雜訊產生的範圍較少，反之則雜訊會較多。不過以個人觀點來看，雖然有雜訊但是會流動的播放畫面，比斷斷續續甚至靜止不動的播放畫面較令人接受。

3.3 播放品質的定義

本研究的最終目標在於維持MPGE視訊播放的品質，但是品質的定義很抽象，認知也很主觀，因此首先必須將所謂的品質予以量化，設定一些指標，以便能夠客觀的比較。

以播放的觀點來看，視訊是否順暢 (smooth)、畫面有無雜訊 (noisy)、動作是否連續 (continuous)，為影響播放品質認知的因素。如果視訊播放的速率忽快忽慢 (jerky)，有時畫面會突然停滯到人類能夠察覺的程度，又突然很快的播出現下來的畫面，這種斷斷續續的現象我們稱之為不順暢。若畫面看起來像是衛星電視被雜訊干擾那樣，一個畫面有些部分看的清楚，有些部分是雜亂無章的圖案，則稱作有雜訊。如果一連串的畫面會讓人感覺到畫面之間的動作接不起來，則是不連續。於是定義三個衡量播放品質的向度 (dimension)，分別為順暢性 (smoothness)、與雜訊 (noise)、連續性 (continuity)。接著我們需設立一些指標 (index) 來量化所謂的播放品質。在順暢性、雜訊、與連續性這三個向度下，定義順暢指標 σ 、雜訊指標 γ ，以及連續指標 μ 。

順暢指標 σ 的算法，為每秒播放出來的圖片張數之變異數。若每一秒播放的圖片張數差異性愈大，顯示視訊播放為忽快忽慢 (即所謂的jerky) 的情形愈嚴重，愈不夠順暢。所以我們用 σ 值來量化視訊播放的順暢程度， σ 值愈小表示播放的愈順暢。

根據『最近參照圖片』的解碼政策，I、P-frame的遺失會使得需參照它們才能解碼的圖片產生雜訊。當I、P-distance愈大，表示需要參照I、P-frame的圖片愈多，因此I、P-frame的遺失，影響到的圖片張數會較多，此外找到的最近參照圖片也會較為過時，讓圖片解碼產生的雜訊較大。於是我們將雜訊指標 γ 的公式定為：

$I\text{-distance} \times I\text{-frame loss ratio} + P\text{-distance} \times P\text{-frame loss ratio}$ 。

來量化有無雜訊的播放品質。不過需要注意的是，雜訊的產生跟視訊的內容很有關係，場景變化愈小的視訊內容，其圖片產生的雜訊範圍會較小，但是雜訊指標 γ 值並無法將視訊內容考慮進去。

連續指標 μ 的公式為： $\text{average actual frame rate} / \text{nominal frame rate}$

假設 $I\text{-distance}$ 為 8， $P\text{-distance}$ 為 4，圖片速率為 30 fps 的 MPEG 視訊流在網路頻寬無法負荷高資料傳輸量，而必須利用我們的選擇性傳輸架構捨棄一些圖片時，若捨棄了所有的 B-frame，則圖片播放率剩下 7.5 fps，其 μ 值為 0.25，表示原本四張能夠構成連續畫面的圖片，只剩下一張播放出來，連續性當然變差。若捨棄了所有的 P、B-frame 不傳，則每秒只剩 3.75 張圖片可以播放，不連續的情況會更加明顯， μ 值為 0.125。 μ 值愈小顯示畫面的內容愈無法連貫， μ 值最大為一。不過同樣的是，連不連續跟視訊的內容也有關係，這點並無法包含在 μ 值之內。

4. 實驗

本章以實作出來的系統作了一系列的實驗，目的有幾：第一、想要了解在網際網路上 MPEG 視訊即時傳輸與播放的能力與現況；第二、比較在不同的網路傳輸狀況下，傳輸與播放的品質；第三、發掘會影響到 MPEG 視訊即時傳輸與播放的因素，以便加以控制。

4.1 實驗環境

本實驗的伺服器端與客戶端，為了涵蓋遠近不同節點的網路傳輸狀況，希望能選取位於 LAN、MAN 及 WAN 上的節點來作實驗。在 LAN 的實驗，我們選擇以台大計算機中心的機器與台大資管系多媒體實驗室的機器相連結。MAN 及 WAN 則以同樣位於 TANET 上的交大與成大的機器，與台大相對應。

本實驗選取了幾個 MPEG video clips 作為實驗用的資料，它們的特徵如下表。

表4-1：實驗用MPEG video clips之特徵

File	Frame Rate (fps)	Resolution	Number of Frames	Pattern (I_dist, P_dist)	Playback Time (sec)
simpson6.mpg	30	192×144	960	(6, 3)	32
simpson8.mpg	30	192×144	545	(60, 3)	18.17
us.mpg	30	352×240	731	(6, 3)	24.37

這些 MPEG video clips 由網際網路上的 public domain 擷取而來，各具有不同的特徵。simpson6.mpg、simpson8.mpg 這些 clips 的圖片播放速率 (frame rate) 以及解析度相同，內容為美國辛普森卡通的動畫影片，但它們各具有不同的 I、P、B-frame 組成型式 (pattern)，用來比較選擇性傳輸架構中圖片的捨棄對播放品質的影響。us.mpg 與其他 clips 相比，解析度較大，內容為美國某一動作片電影的片段，被拿來與其他 clips 比較即時解碼與播放的處理時間。

我們主要設計了三個實驗，爲了描述簡單明瞭起見，底下將使用一些簡稱。我們將台大校園網路內的實驗稱之NTU，台大與交大，以及台大與成大之間的實驗，則分別稱之爲NCTU與NCKU。在伺服器端的初始發送速率（initial sending rate）簡稱爲ISR。ISR又分爲以nominal rate（e.g. 30 frames per second）傳輸，以及用best-effort rate傳輸，分別以N-ISR以及B-ISR稱之。若500 Kbytes的先存資料量，配合N-ISR，則以N-500稱之，100 Kbytes的先存資料量配上B-ISR則稱爲B-100，其餘依此類推。

除了實驗一外，其餘實驗的客戶端機器的CPU 皆具Pentium-150以上的處理能力。

實驗一：不同等級機器之NTU實驗

爲了測知不同處理能力的機器，在接收網路傳輸的資料，以及即時解碼與播放的能力，我們在大校園網路，以分別配備Pentium-150、Pentium-133、及Pentium-90 CPU及32 Mbytes RAM的個人電腦當作客戶端，Pentium-166的個人電腦當作伺服器端，在N-ISR及B-ISR交替使用以及500 Kbytes的先存資料量之設定下，進行了MPEG視訊即時傳輸與播放的實驗。

實驗二：分時之NTU、NCTU、與NCKU實驗

爲了探知遠近不同的網路環境，以及網路負荷量不同的時段，MPEG視訊即時傳輸與播放的情形，我們在一般時段（早上八時至十時）以及尖峰時段（中午十二時至下午二時）[21] 分別做了NTU、NCTU、與NCKU的實驗。系統設定的客戶端之先存資料量爲500 Kbytes，伺服器端使用B-ISR傳輸先存資料。本實驗以表4-1中的五個video clips，分別作了所有圖片皆傳輸、捨棄所有B-frame不傳、以及捨棄所有B、P-frame不傳的傳輸與播放實驗。

實驗三：不同先存資料量與不同初始發送速率之NCKU實驗

我們想知道不同的先存資料量配合不同的ISR跟MPEG視訊即時傳輸與播放的關係，因此設計了50、100、200、300、400、及500 Kbytes的先存資料量，配合N-ISR與B-ISR進行此實驗。

4.2 結果分析討論

根據上一節設計的實驗進行得到的結果，我們可以發現一些值得討論的議題，本節即針對我們觀察到的現象，以打破不同組實驗界限的方式，做一綜合全盤的分析與討論。在以下討論到的封包遺失率若未特別指明，則爲當天的一個實驗時段內，伺服器端送出而客戶端卻沒收到的封包總數，佔所有送出封包總數的百分比率。

4.2.1 校園網路環境

在NTU實驗中，MPEG視訊傳輸發生封包遺失的現象非常的稀少，封包遺失率不到萬分之一，幾乎所有的封包都能完整重組成MPEG視訊的圖片，並即時的解碼與播放。觀察視訊播放的情形，就如同直接由硬碟裡開啓檔案播放一樣，非常地順暢。由此實驗可知即時視訊的傳輸與播放，在類似台大校園網路的環境下，是十分可行的。

4.2.2 機器的處理能力對即時傳輸與即時解碼播放的影響

原本我們不清楚我們使用的軟體解碼器在網際網路即時視訊傳輸與播放的效能 (performance) 如何，是否能在需兼顧處理網路傳輸的條件下，達到即時的解碼與播放。由實驗一，我們觀察到配備為Pentium-150個人電腦的客戶端，MPEG視訊的解碼與播放能夠達到即時性，如表4-2至表4-4。

表4-2：Pentium-150個人電腦下分類圖片平均處理花費時間 (msec)

	I	P	B	Average
simpson6.mpg	18.46	20.08	12.06	14.47
simpson8.mpg	19.80	23.36	12.66	16.19
us.mpg	41.36	36.58	20.80	26.98

表4-3：Pentium-133個人電腦下分類圖片平均處理花費時間 (msec)

	I	P	B	Average
simpson6.mpg	28.26	29.30	17.20	21.07
simpson8.mpg	30.12	31.51	19.55	23.44
us.mpg	51.72	47.48	27.68	34.86

表4-4：Pentium-90個人電腦下分類圖片平均處理花費時間 (msec)

	I	P	B	average
simpson6.mpg	56.32	57.99	39.17	45.18
simpson8.mpg	55.83	59.24	39.61	46.03
us.mpg	94.12	83.81	54.86	66.30

除了資料解碼的即時性外，在網路傳輸方面，我們發現機器的處理能力 (processing power)，也是影響MPEG視訊即時傳輸品質的因素之一。

表4-5：不同機器處理能力下封包遺失狀況

	packet loss ratio (%)	
	N-ISR	B-ISR
Pentium-150	0	0
Pentium-133	0	17.34
Pentium-90	0.32	26.47

由表中可以觀察到，在N-ISR的設定下，封包傳輸的情形依舊良好，但是在B-ISR的設定下，處理能力越弱的機器，產生封包遺失的情形就越多。根據比較表中數值以及4.2.1節談到台大校園網路封包遺失率幾乎是零等結果，可判定封包遺失並非是在網路傳輸的過程中產生，而

是機器的處理能力不足而有遺失的現象發生。在用best-effort的發送速率傳輸先存資料情況下，發送端的ISR平均會在900 Kbytes/sec以上，在這樣高速的傳輸速率下，處理能力不足的客戶端當然無法負荷，而有封包傳輸到卻無法即時處理被丟棄的機會。這個問題為傳統流量控制（flow control）的議題，在資料透過網路傳輸的兩端點間，節點的能力必須能夠互相匹配，才不致會有能送卻不能收的情形發生。

4.2.3封包遺失的狀況

資料在網路的傳輸過程中，遭受到的封包遺失是降低播放品質最主要的因素，在順暢性（smoothness）、連續性（continuity）、以及有無雜訊（noise）等三個播放品質指標上，封包遺失的影響程度都佔了相當的比重。在我們實驗中觀察到的封包遺失狀況可分為連續大量的封包遺失、小額的封包遺失、以及突發的大量封包遺失等三種情形，本節則詳細地描述這些現象，並分析其成因與討論對應方法。

資料先存階段（preload phase）封包大量遺失的情形

在我們的實驗二中，發現以best-effort的ISR傳輸先存資料，會引發大量的封包遺失，如圖4-1。

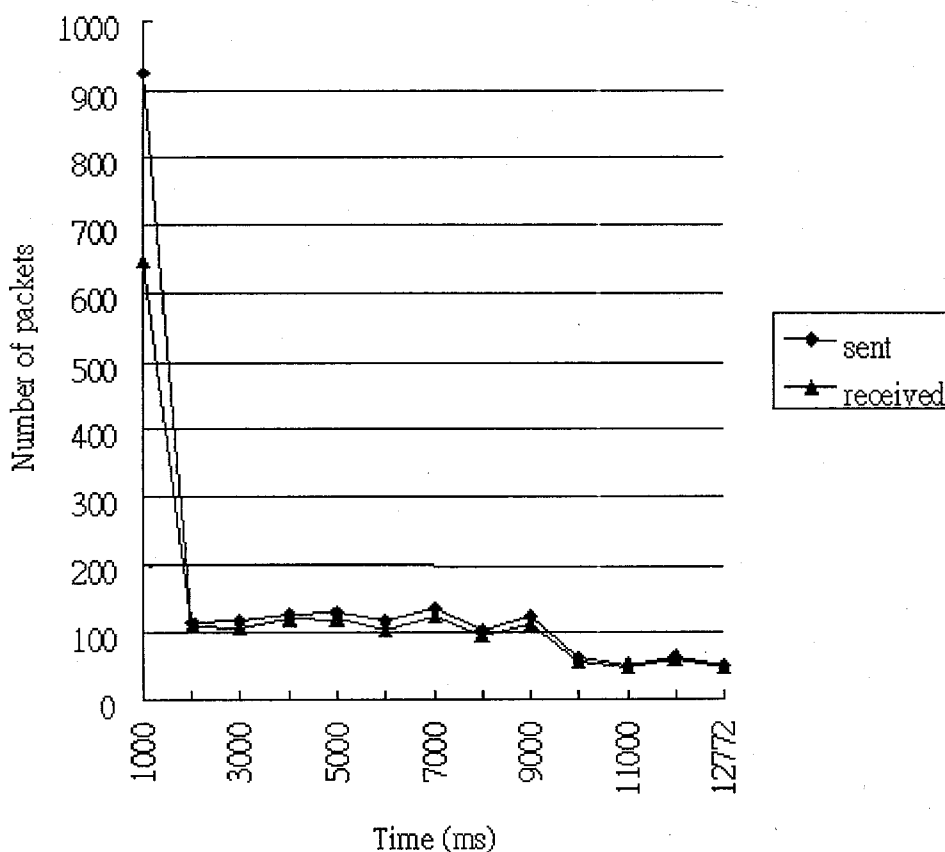


圖4-1：先取階段封包大量遺失的傳輸封包數與時間關係圖（simpson8.mpg）

將在資料先存階段（data preloading phase），以及後來調降到nominal rate階段的估算封包遺失率製表如下，確實可發現大部分的封包遺失發生在高速傳輸的資料先存時間內。

表4-6：在Best-effort Initial Sending Rate下資料先存階段與後續階段的估算封包遺失率

一般時段				
	先存時間 (sec)	先存階段	後續階段	整體
NCTU	1.048	18.44%	11.21%	15.23%
NCKU	1.686	30.33%	5.62%	28.33%

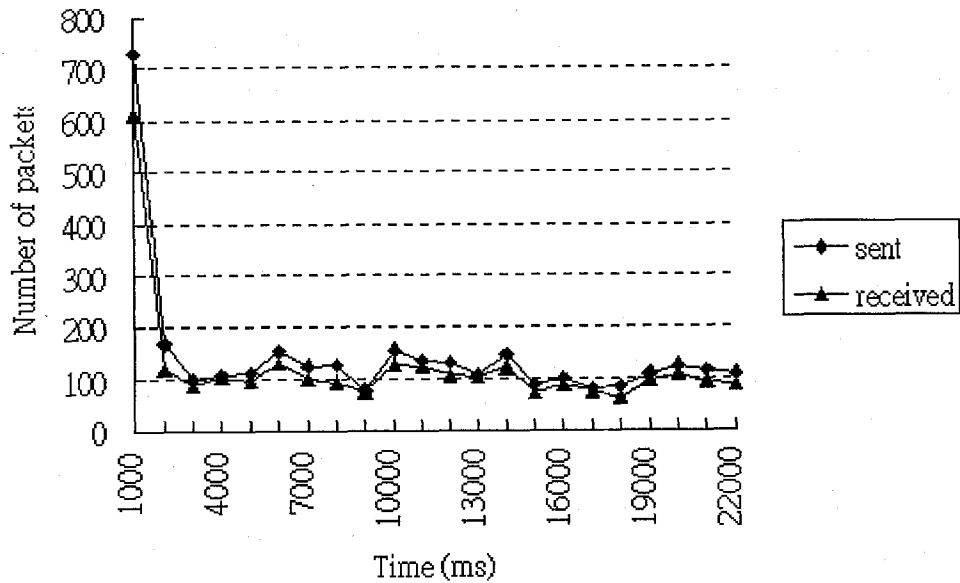
尖峰時段				
	先存時間 (sec)	先存階段	後續階段	整體
NCTU	2.209	52.94%	12.79%	45.14%
NCKU	2.311	32.88%	9.33%	29.52%

由以上分析可知造成封包大量遺失其原因是資料先存階段內封包發送的速率太快了，而使網路傳輸的交通量（traffic）有bursty的現象，導致一時間網路無法消化，進而丟棄無法處理的封包。一旦客戶端經過了資料先存階段，而伺服器也將速率調降到nominal rate後，封包遺失率就沒有那麼高了。

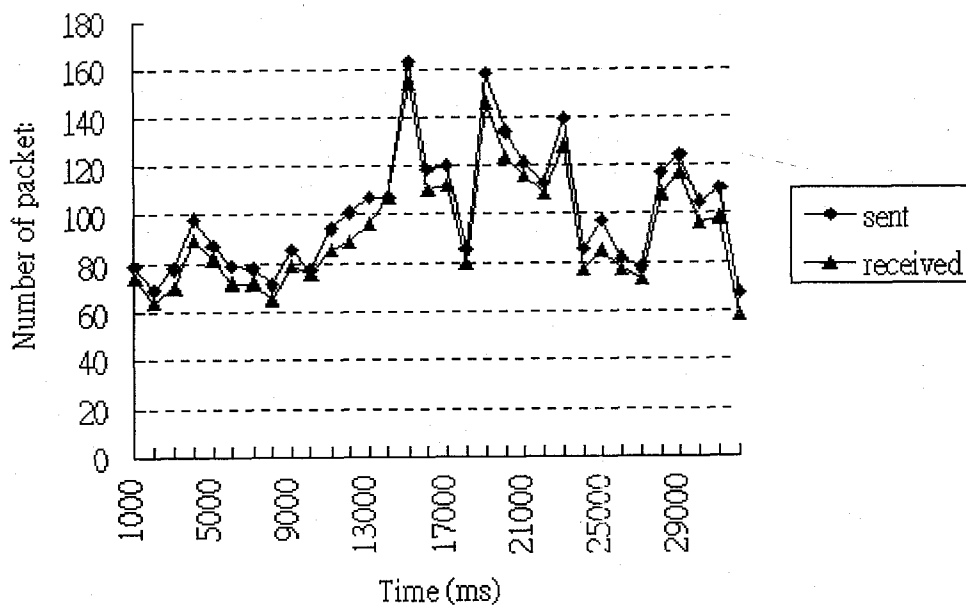
更進一步的分析，這種剛開始以best-effort initial sending rate傳送資料的政策，很容易讓封包接收的品質持續惡化。因為初始以極快的發送速率傳輸，造成過多的封包遺失。而封包遺失的越多，接收端緩衝區填滿到500 Kbytes先存資料量的時間經歷越長，而使速率調降請求的控制封包越晚發送出去。發送端越晚收到速率調整之控制封包，高發送速率的時間越長，而導致了最後整體的封包遺失率偏高。

小額的封包遺失

在實驗三裡，不管是在B-ISR或N-ISR的條件下，我們發現到在傳輸的過程中，時時都有封包遺失的發生，但是量都不大，如圖4-2。這種現象可歸因為網路的變動性（network dynamics）。因為網路是一個共享的傳輸媒介，除了我們的資料外，還會有各式各樣的traffic在網路上流通，因此也會產生各種的不確定性（uncertainty），導致本例中時時都會發生的小額封包遺失。這種無法預測的變動性，也是在我們的研究中，想要設計適當的措施去因應的理由。



(1) 在Best-effort Initial Sending Rate下



(2) 在Nominal Initial Sending Rate下

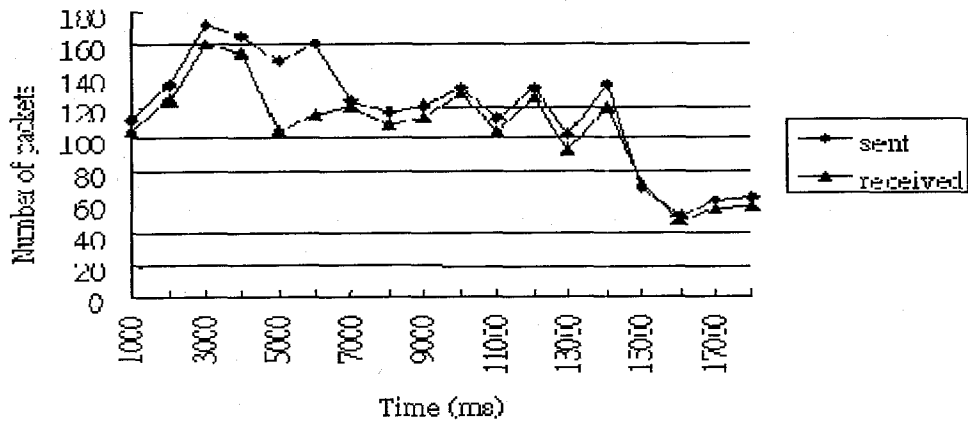
圖4-2：小額封包遺失之傳輸封包數與時間關係圖（simpson6.mpg）

突發的大量封包遺失

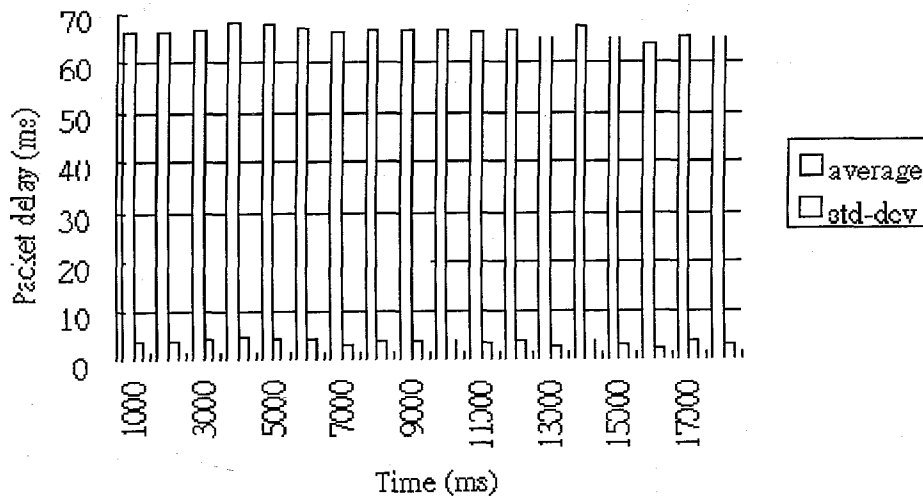
除了隨時都有可能發生的小額封包遺失外，在實驗中我們還觀察到了偶爾會發生的大量封包遺失，如圖4-3。我們可看到在第五秒內封包接收量突然降了下來，觀察封包到達間隔時間的情形（圖4-3之(3)），發現在這一秒內的封包到達間隔時間，與傳輸過程中的其他時間相比，平均值與標準差都較大，因此可研判此時網路可能正值壅塞，進而導致封包傳輸的遺失。

若欲面對這種突然發生的瞬間（transient）壅塞，我們實作的系統應做的是盡量不要造成 bursty traffic，以降低網路的變動性。另一方面，壅塞會造成的資料遺失，而使得沒有資料可以播放，形成播放的中斷（playback interrupt）。我們設計的客戶端必須先存資料在緩衝區內，就是為了克服這種窘境，而在緩衝區內的資料耗盡之前，做出補救的措施。

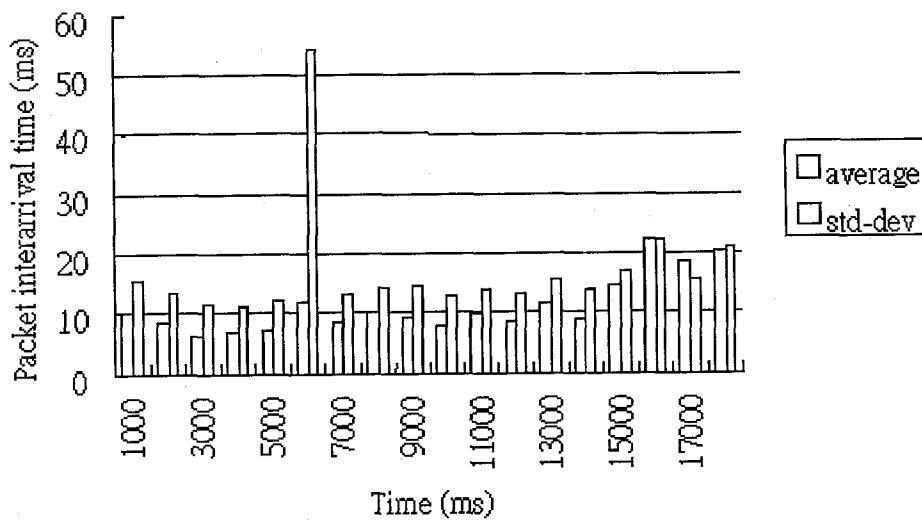
由以上的觀察分析，可以看到封包遺失的概況，由這裡衍生了一些值得探討的議題。尤其是資料先存的策略，有必要再深入研究先存資料量該設定多少才適宜，以及該不該以best-effort initial sending rate傳輸先存資料等問題，這些在往後的小節中會陸續探討到。



(1) 傳輸封包數與時間關係圖



(2) 封包延遲與時間關係圖



(3) 封包到達間隔時間與時間關係圖

圖4-3：在Nominal Initial Sending Rate下突發之大量封包遺失 (simpson8.mpg)

4.2.4即時且可靠的控制封包傳輸之必要性

除了觀察到以best-effort initial sending rate傳輸先存資料會造成大量封包遺失，我們還觀察到不可靠的（unreliable）以及不夠即時的控制訊息傳輸會惡化封包遺失的情形。由於RTP架構在UDP之上，我們的RTP即時控制封包在傳輸的過程中有可能發生遺失。在我們的系統架構中，伺服器被動地接收客戶端傳來的控制訊息，才會從事控制的動作。若客戶端在資料先存完成後，送出的速率調整之控制封包在傳輸中途丟掉了，則伺服器不會調整速率，於是在實驗二的設定下，伺服器會一直以best-effort initial sending rate傳輸資料，灌輸大量資料到網路中而造成高封包遺失率。

伺服器若太晚接收到速率調整之控制封包，甚至沒有接收到，則封包遺失的情形會較嚴重，如圖4-4、圖4-5。

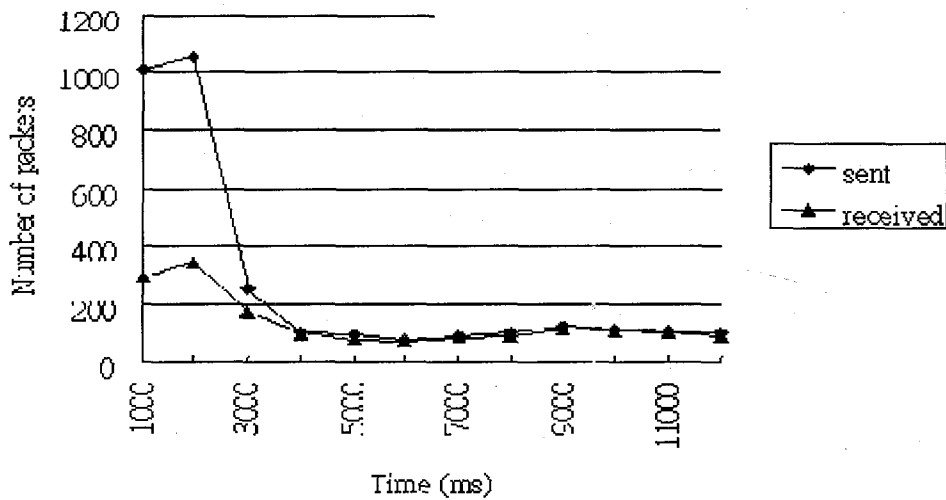


圖4-4：速率控制封包太晚到達之傳輸封包數與時間關係圖（simpson6.mpg）

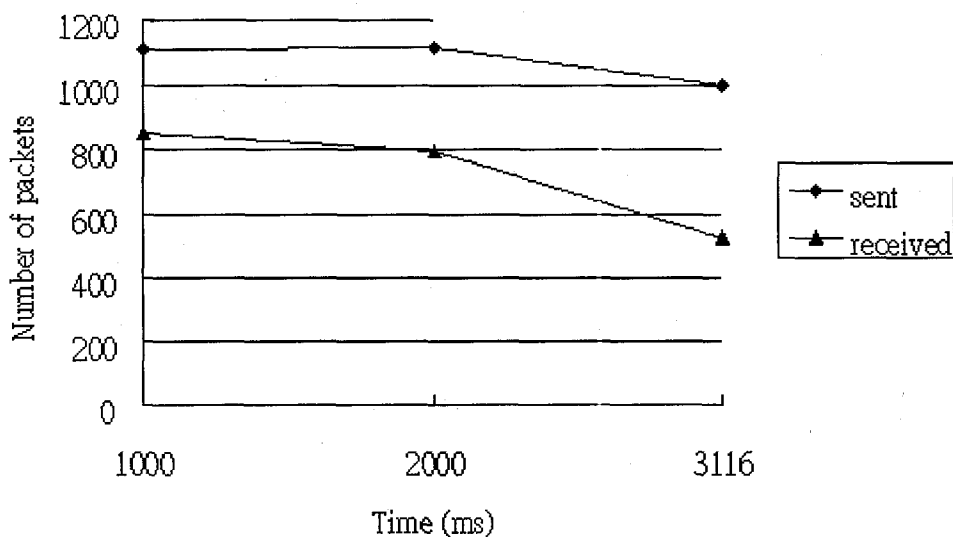


圖4-5：速率調整之控制封包遺失之傳輸封包數與時間關係圖（simpson6.mpg）

4.2.5 傳輸狀況對播放品質的影響

在討論了這麼多的傳輸狀況，到底實際上播放的情形如何，這些狀況又對播放品質有何影響，在本節中會詳加討論。

Sequence Header的遺失

在實驗二中，所有的video clip皆是以B-ISR來傳輸，我們觀察到視訊完全沒有播放的比率很高。以實驗二為例，在NCTU與NCKU的實驗有將近三、四成的比率視訊沒有播放出來。以收集的統計資料來看，發現這種資料完成了傳輸卻未播放的原因為MPEG視訊流中的Sequence Header並未成功傳輸到客戶端。Sequence Header為一段MPEG視訊的起頭，它是解碼器辨認MPEG視訊資料以及解碼的根本依據。在我們實作的系統中，Sequence Header的資料會放置在整段視訊中第一張I-frame切成的第一個封包內。若此I-frame遺失掉，則等於此Sequence Header也遺失掉了，而使得解碼器無法對收到的資料解碼，更遑論播放了。

既然Sequence Header這麼重要，其他資料又需依據它才能解碼，於是直覺的可以知道這種重要的資料絕對需要可靠的傳輸。

封包遺失的影響 (packet loss effect)

封包的遺失到底在frame level造成多大的影響，我們以us.mpg這個video clip為例，來看真實傳輸的情形。

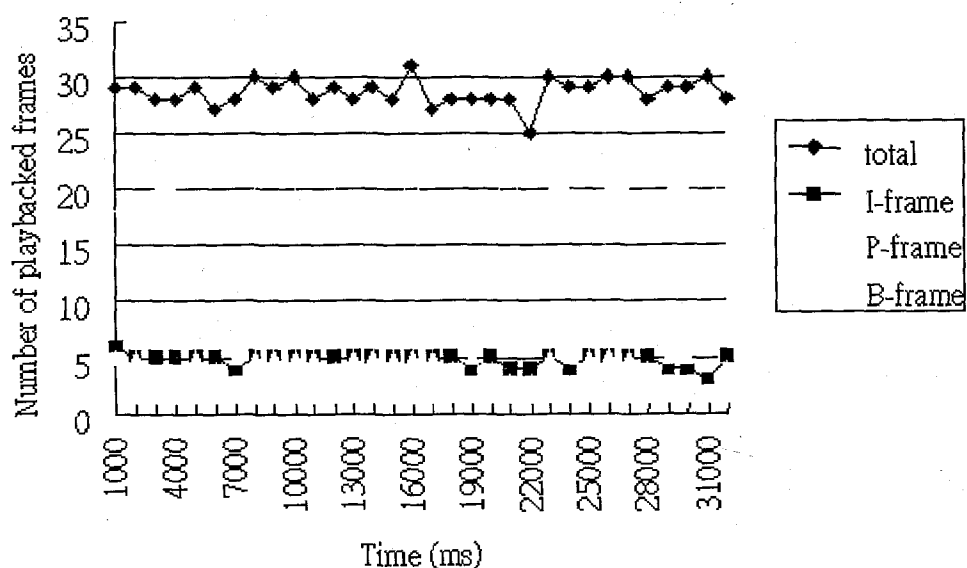
表4-7：us.mpg之封包遺失影響

	packet loss ratio	frame loss ratio	packet loss effect (frame loss/packet loss)	useless packet ratio
I	8.02%	38.21%	4.76	13.14%
P	5.76%	24.59%	4.27	6.40%
B	8.83%	16.46%	1.86	1.26%
Total	7.76%	21.48%	2.77	6.77%

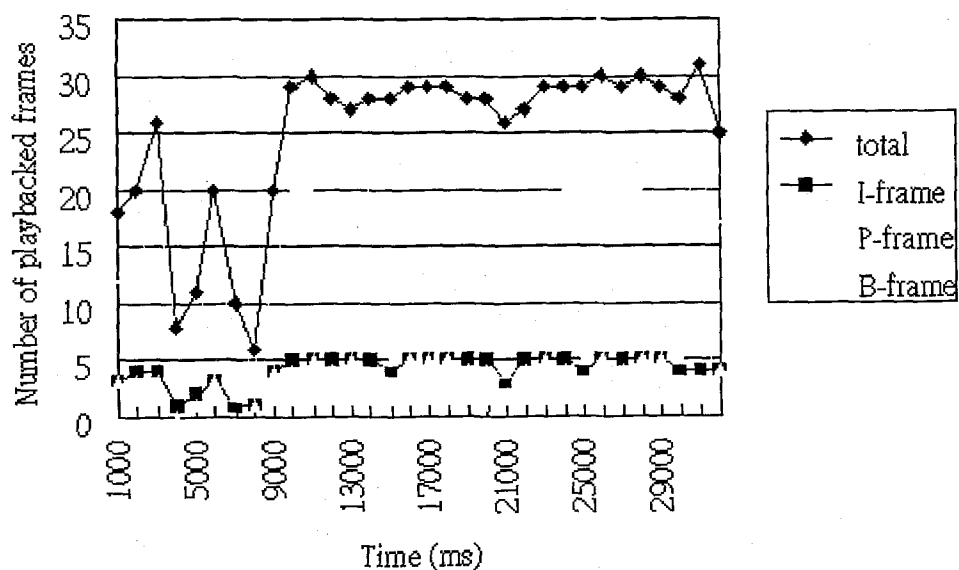
表4-7將封包遺失以及相對應的圖片遺失 (frame loss) 與無用封包的數據列出來。在此圖片遺失包括了沒有傳送到的圖片，以及收到的損壞圖片 (broken frame)。平均而言，我們可以發現1%的封包遺失造成2.77%的圖片遺失。以播放品質的觀點來看，圖片遺失是造成不順暢、不連續、有雜訊的主因。不同類型的封包遺失，其影響程度也有不同。例如1%的I-packet的遺失，導致此次實驗4.76%的I-frame遺失，更影響了4.76% × 5的P、B-frames (此clip的I-distance為6) 解碼的不正確。此外還造成13.14%收到的I-packet沒有用處，等於浪費了頻寬。由此可知，盡量保證I、P-packets的傳輸不被遺失，是維持播放品質的準則。

不順暢的播放畫面

本小節以實驗三中simpson6.mpg在N-500與B-500設定下之播放情形為例，來比較不同順暢程度的播放品質。此video clip在N-ISR與B-ISR的傳輸下，封包遺失率分別為1.39%及8.02%，而圖片遺失率則分別為4.58%及17.29%。將這兩次實驗的播放情形畫成橫軸為時間，縱軸為分類之圖片播放數的關係圖，如圖4-6。



(1) Nominal Initial Sending Rate with 500 Kbytes preloaded data



(2) Best-effort Initial Sending Rate with 500 Kbytes preloaded data

圖4-6：不同初始發送速率下之圖片播放數與時間關係圖（simpson6.mpg）

由圖4-6之(1)中很明顯地可以看到播放的情形波動不大，每秒播放出的圖片數落在25至31張之間。以我們定義的順暢指標來衡量，其 σ 值為1.16，圖(2)的 σ 值為7.10，我們也可以從圖中觀察到其播放的波動程度較大，所以我們可以說圖4-6之(1)中的例子在播放品質的順暢性上，比圖4-6之(2)的例子優異。

更進一步探討造成圖4-6之(2)例子比圖4-6之(1)例子不夠順暢的因素，為資料先存階段載入的資料封包遺失率較高，而導致前面500 Kbytes的資料重組為損壞圖片 (broken frame) 的比率也較高，所以在圖中看到播放不順暢、忽快忽慢的情形，發生在影片之前段。

此外導致影像播放速率忽快忽慢的因素尚有決定圖片播放時間的time code資料沒有傳送到。在MPEG視訊流的結構中，有一GOP header，此header中包含了一個time code的欄位，記錄視訊從頭開始到目前為止，應該經歷的播放時間。在我們系統，每一張圖片的播放時間是根據此time code來相對決定的。若包含GOP header的封包遺失掉了，則圖片的播放時間值會無法正確計算出來，而造成圖4-6之(2)中高高低低的播放速率。

由以上分析，我們可以知道資料封包的遺失，是造成視訊播放不順暢之原因，而且I-distance越大的視訊，其在順暢性上的播放品質越容易惡化。這也給予我們一個啓示，若MPEG視訊流的I-distance越大，其I-frame重要程度越大越不能夠遺失。

有雜訊的播放畫面

本節繼續使用圖4-6的例子來討論。(1)的雜訊指標 γ 值為0.47，而(2)的 γ 值為1.77，後者播放的影像畫面中有雜訊的機會較多。根據我們解碼的『最近參照圖片』(most recent reference frame)政策，遺失掉一張I-frame，會導致接續在其後面張P、B-frame的解碼參照到上一張完整收到的I-frame，而產生雜訊，於是圖片遺失率較大的(2)雜訊較多。

動作不連續的播放畫面

會讓播放畫面中的動作產生不連續之因素，為我們捨棄了一些圖片不傳送，而使每秒能夠播放的圖片數減少到會使視訊的內容無法連貫，人眼又能察覺到的程度。根據我們實驗觀看視訊播放的感覺，主觀上來說，10 fps的播放速率還能令人接受。單從數值上的分析，表4-1中的video clips之P-distance為3，若以我們的選擇性傳輸架構捨棄所有的B-frame不傳，則剩下的frame rate為10 fps，品質還能令人接受。若捨棄所有的B、P-frame不傳，則剩下的frame rate依序為5 fps，0.5 fps，及5 fps，這樣播放的影像內容已經難以連貫，尤其I-distance愈大者frame rate愈小，連續性也愈差。

用實驗的結果來看，將simpson6.mpg在N-ISR與B-ISR條件下，分別以傳輸所有圖片、捨棄所有的B-frame不傳、捨棄所有的P-frame不傳等三種設定作實際的傳輸與播放，得到的連續指標 μ 製表如下。

表4-8：不同frame rate下的連續指標 μ 值

	All frames	Skip B-frame	Skip P-frame
N-500	0.95	0.31	0.15
B-500	0.83	0.23	0.07

因為圖片遺失的關係， μ 值會小於1。表中發現高封包遺失情況（B-500）下，其 μ 值會更小。 μ 值為0.15表示平均構成連續畫面的100張圖片中只播放了15張，很明顯會有動作不連續的現象發生。

4.2.6 不同先存資料量與不同初始發送速率

在我們設計的即時視訊播放系統中，需要將緩衝區先存放一定量的資料，以在傳輸品質下降時仍然能夠維持播放品質，並在緩衝區資料消耗殆盡之前有充裕的時間作出補救傳輸品質的措施。先存資料量設定的愈大，能夠容忍傳輸品質下降的程度也愈大，但是資料先存的時間會增長。在我們剛開始作實驗以best-effort initial sending rate傳輸的目的，就是為了讓資料先存能夠快速的完成，減少使用者需等待視訊開始播放的時間。preloading time、preload data size、以及initial sending rate的關係式如下：

preloading time =

當先存資料量愈大，則需以較大的ISR才能讓preloading time降低。然而在之前我們已經討論過傳輸速率過快的資料先存階段，會造成大量的封包遺失，進而影響視訊播放的品質。因此有必要仔細探討先存資料的大小對即時視訊的傳輸與播放有何影響。實驗三即是為了這個目的而設計的。我們以simpson6.mpg來說明，圖4-7為不同先存資料量，以不同ISR傳輸的封包遺失率圖。在B-ISR下，隨著資料先存量設定的愈大，封包遺失率也愈大。以N-ISR傳輸的情形，在不同的資料先存量下，封包遺失率差距不大。

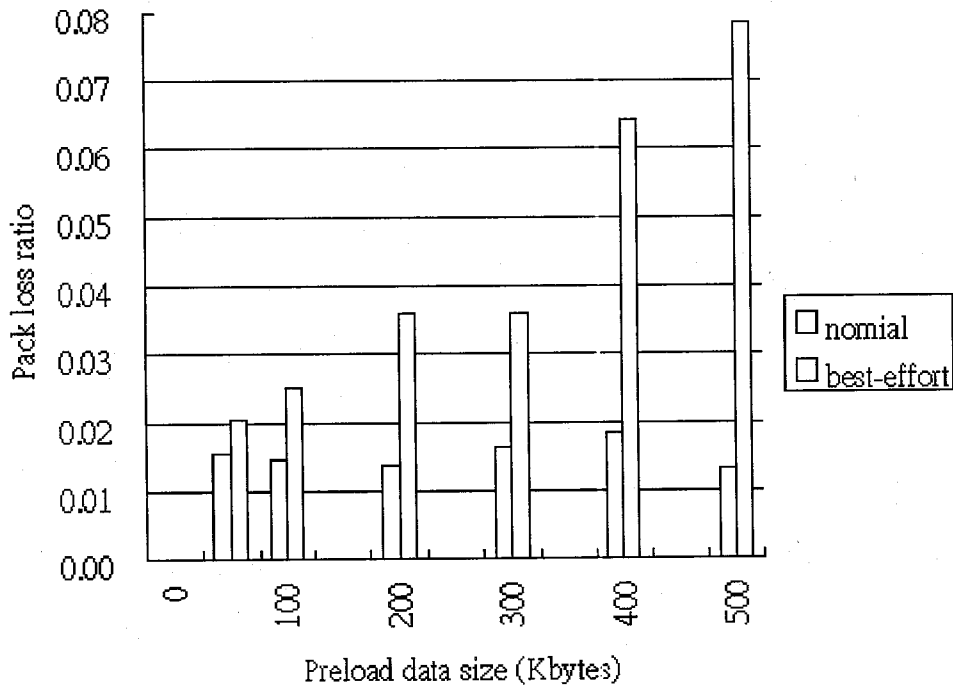


圖4-7：在不同初始發送速率下封包遺失率與先存資料量關係圖（simpson6.mpg）

最後來看看這些實驗的資料先存時間（圖4-8）。資料先存時間決定了使用者提出視訊播放請求到真正看到畫面開始播放的等待時間，以個人觀點來看，三秒鐘的等待時間尚可接受。由圖中可看到若以N-ISR傳輸資料的話，在先存資料量為200 Kbytes以上的情況下，資料先存時間已經超過四秒鐘。以B-ISR傳輸，雖然資料先存時間會較短，但是播放品質在先存資料量愈大的情況下會愈差。因此我們有必要在資料先存時間、先存資料量、以及初始發送速率三者之間，找尋一個平衡點。

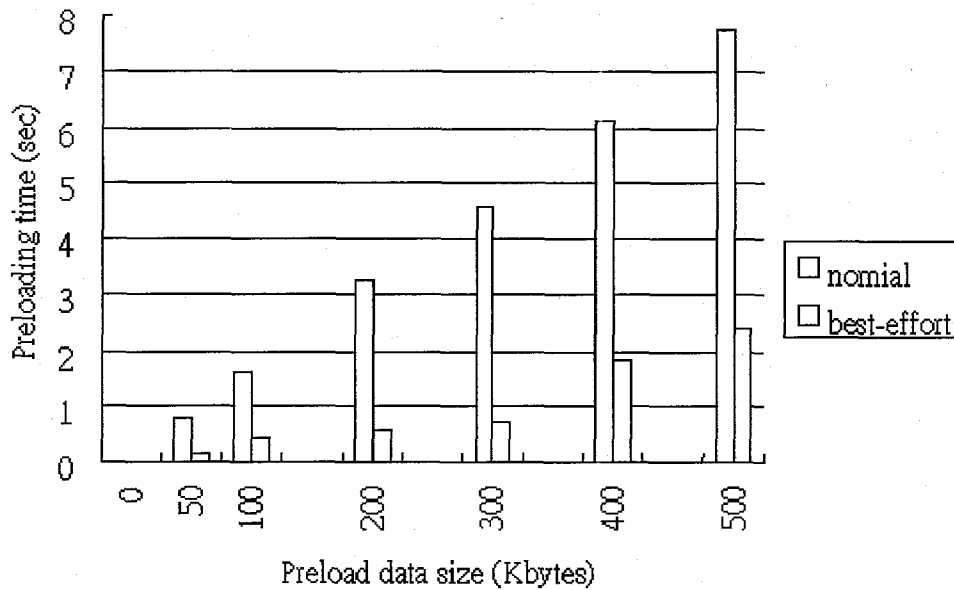


圖4-8：在不同初始發送速率下資料先存時間與先存資料量關係圖

4.3 總結

經過以上的分析與討論，我們已經深入了解即時MPEG視訊傳輸與播放會遭遇到的問題，在此將觀察到的現象作一個整理，並提供一些設計即時MPEG視訊傳輸與播放系統的建議：

- 機器的處理能力，會影響MPEG視訊傳輸與播放的即時性。在電腦硬體的效能愈來愈佳，價格愈來愈便宜的趨勢下，我們相信對即時MPEG視訊傳輸與播放有所幫助。
- 軟體解碼器在電腦處理能力的提昇下，已經可以達到即時解碼的能力。以軟體解碼器取代硬體解碼器，可以提供更富彈性的解碼功能，且能降低電腦硬體配備的成本。
- 在類似台大校園網路的環境裡，單點對單點（peer to peer）的即時MPEG視訊傳輸與播放之品質令人非常可以接受。
- 網路變動性（network dynamics）會造成小額封包遺失與瞬時大量的封包遺失等多種無法預測的網路傳輸品質下降之情況，可利用資料先存政策來應付這種變動性。
- 控制封包（control packet）需要即時且可靠的傳輸，才能快速有效地控制傳輸與播放品質，達到控制的目的。
- 過快的傳輸速率是導致網路傳輸的封包遺失率升高、播放的品質不佳之因素。
- 資料先存量不宜太小，以免造成播放中斷；也不宜太大，而使資料先存時間太長。
- MPEG視訊流中的資料有相依的關係（dependence relation），因此需要區分出重要程度不等的資料，以更可靠的方式傳輸，以維持視訊播放的品質。以frame level來看，I-frame比P-frame重要，P-frame又比B-frame重要。以packet level來看，I-frame的第一個封包異常重要，因為它會包含Sequence Header、GOP Header、以及Picture Header。
- 先存資料量與初始發送速率需要謹慎地調整，以便能不讓使用者等待太久，使傳輸狀況不會惡化，且又能維持播放品質。
- 網路頻寬還是保證即時MPEG視訊傳輸與播放之品質能夠維持的主要因素。

5. 結論與未來展望

在網際網路上傳輸且播放即時的連續媒體資料，並維持傳輸與播放的品質，是本論文想要研究的議題。在深入研究後，以我們設計及實作的即時MPGE視訊傳輸與播放系統，從事一系列的實驗，我們得到重要結論有三，若欲保證MPEG視訊播放的品質良好，則：第一、重要的資料需要可靠的傳輸，以維持播放的品質，例如MPEG video stream中的Sequence Header、GOP Header、I-frame等；第二、控制封包需要即時且可靠的傳輸，以便能夠迅速有效的達到控制的目的；第三、先存資料量與初始傳輸速率必須謹慎地協調出合適的值，才不致惡化影像播放的品質。

本研究尚有許多議題可以再深入探討。我們設計出的選擇性傳輸架構目的在支援動態的傳輸速率控制（dynamic rate control），以因應網路傳輸的變化。但是目前尚未有速率控制的策略出爐應用此架構，因此未來可以研究動態的傳輸速率控制對MPEG視訊傳輸與播放的品質有何影響。本研究觀察到重要資料的遺失對視訊播放的品質有很大的影響，因此一方面可以設計重傳（retransmission）的機制，保證資料能夠可靠地傳送。另一方面，也可以發展錯誤回復（error recovery）的方法，補救封包遺失的影響。此外，我們尚未善加利用到RTP傳輸機制下提供的功能，今後可以朝著利用RTCP收集的統計資料，找出欲動態調整的傳輸速率值這個方向研究。

6. 參考文獻

- [1] J. C. Bolot, T. Turetti, and I. Wakeman, "Scalable feedback control for multicast video distribution in the Internet", ACM SIGCOMM, pp. 58-67 Aug 1994
- [2] I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS control of multimedia applications based on RTP", Computer Communications, Jan 1996.
- [3] Z. Chen, S. Tan, R. Campbell, and Y. Li., "Real time video and audio in the World Wide Web", Proc Fourth International World Wide Web Conference, 1995.
- [4] D. D. Clark, and D. L. Tennenhouse, "Architectural considerations for a new generation of protocols", Proceedings of SIGCOMM '90, ACM, Sep 1990
- [5] D. E. Comer, *Internetworking with TCP/IP, Volume I*, 2nd ed, Prentice Hall, 1991.
- [6] D. Hoffman, G. Fernando, and V. Goyal, "RTP Payload Format for MPEG1/MPEG2 Video", Internet Engineering Task Force, Audio-Video Transport Working Group, RFC2038, Oct 1996
- [7] D. Le Gall, "MPEG: A Video Compression Standard for Multimedia Applications", Communications of the ACM, Vol 34, Num4, pp46-58, Apr 1991
- [8] ISO/IEC International Standard 11172; "Coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbits/s", Nov 1993.

[9] ISO/IEC International Standard 13818; *'Generic coding of moving pictures and associated audio information'*, Nov 1994.

[10] S. McCanne, and V. Jacobson, *'vic: A flexible framework for packet video'*, ACM Multimedia, Nov 1995.

Source code available: <ftp://ftp.ee.lbl.gov/conference/vic/>

[11] J. B. Postel, *'Internet Protocol (IP)'*, Network Information Center, SRI International, RFC 791.

[12] J. B. Postel, *'Transmission Control Protocol (TCP)'*, Network Information Center, SRI International, RFC 793, Sep 1981.

[13] J. B. Postel, *'User Datagram Protocol (UDP)'*, Network Information Center, SRI International, RFC 768, Aug 1980.

[14] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, *'RTP: A Transport Protocol for Real-Time Applications'*, Internet Engineering Task Force, Audio-Video Transport Working Group, RFC 1889, Jan 1996.

[15] H. Schulzrinne, GMD Fokus, *'RTP Profile for Audio and Video Conference with Minimal Control'*, Internet Engineering Task Force, Audio-Video Transport Working Group, RFC 1890, Jan 1996.

[16] T. Turetti, *'The INRIA videoconferencing system (IVS)'*, *ConneXions* 8, 10, Oct 1994.

[17] D. E. McDysan, and D. L. Spohn, *ATM Theory and Application*, McGraw-Hill, 1995.

[18] K. Patel, B. C. Smith, and L. A. Rowe, *'Performance of a Software MPEG Video Decoder'*, ACM Multimedia 1993.

[19] <http://www.mpeg.org/MSSG>

[20] H. Tokuda, *Operating System Support for Continuous Media Applications*, chapter 8, *Multimedia Systems*, edited by J. F. Koegel Buford, Addison-Wesley, 1994.

[21] 惠普科技公司，*國立台灣大學資訊暨網路中心之網路健診報告*，1997年1月