

# 分散式共享視訊記憶體之虛擬顯示桌面互動系統 Virtual Desktop Interactive System by Distributed Shared Video RAM

方立            楊政遠            曾黎明\*            陳奕明\*\*  
Fan Li            Zeng-Yuan Yang            Li-Ming Tseng            Yi-Ming Chen

國立中央大學資訊工程研究所            國立中央大學資訊管理研究所\*\*  
Department of Computer Science and Information Engineering  
Department of Information Management\*\*  
Nation Central University, ChungLi, R.O.C

{fanton,yzy}@dslab.csie.ncu.edu.tw    cym@im.mgt.ncu.edu.tw\*\*

## 摘要

本論文主要描述一個以分散式共用記憶體技術為基礎的遠距顯示視訊傳送系統。針對 PC 螢幕顯示模式採外掛的顯示驅動介面,並利用記憶體映射輸入/出(memory-mapped I/O)的特性,將網路上某一組 PC 的所有螢幕顯示視訊記憶體(video memory)都是為同一個 DSM(Distributed Shared Memory),來達到網路上 PC 之顯示視訊記憶體共用之虛擬顯示桌面效果;此技術的應用也是達成群組合作系統(CSCW)、遠距教學(distance learning)、虛擬教室(virtual classroom)的一個重要方法。我們亦利用對系統輸入事件的轉向式處理來達到多人輪流控制遠端桌面應用程式之互動介面,在此我們也針對遠距教學為例,實作出遠距教學視訊共享系統,證明了此技術應用上的可用性與廣泛。

關鍵字: 分散式共享視訊記憶體, 虛擬共享顯示桌面

## Abstract

Desktop visual sharing is the basic function of "Multiuser Application," such as CSCW, distance learning, and virtual classroom. This paper depicts how to apply distributed shared memory technology to develop remote display sharing system. We treat video memory on different PC's video adaptors as distributed shared memory to fulfill shared virtual desktop system. We also take distance learning as an example to demonstrate our approach can apply to "Multiuser Application" widely.

## 1. 緒論

顯示螢幕是個人電腦架構中最重要也是最基本的輸出/入裝備,而透過顯示螢幕內容的共用、共享也是達成群體合作系統(CSCW)、遠距教學(distance learning)、及虛擬教室(virtual classroom)等應用的關鍵技術。

以往,達成顯示內容共用的解決方法大致可歸納為三類:第一類的技術是透過特殊的通訊硬體,將電腦顯示幕的內容廣播至其他的電腦上。第二類的系統稱為共用視窗系統(Shared Windows

System),多數建立在 X Window 之上,利用 X Window client-server 的特性,以一個 X multiplexor[1]作為 X server 與 X client 的中介。第三類技術為特殊設計的群體合作軟體[2]。

根據上述的背景說明來看,過去所開發的顯示幕內容共用技術,若需要特殊的設備支援,則其無法普遍的適用於廣域網路環境;若建構於 X Window 之上,則應用程式的普及率會使其實用性受限制;若屬於特殊設計的群體合作軟體,其所能提供的能力僅止於該程式。針對這些不足之處,我們提出一個無須硬體設備支援,並且與應用軟體無關的顯示幕內容共用之虛擬顯示桌面系統。同時,我們將開發平台設定為普及率高且擁有豐富應用軟體的 Windows 95 作業系統上。

## 2. 研究背景與方法

PC 上螢幕顯示的輸出是以螢幕顯示卡(VGA card)上的視訊記憶體(video memory, or display buffer)當作螢幕輸出影像的記憶體緩衝區;任一時刻內,螢幕所輸出所輸出的影像都會有一份數位的影像拷貝留在視訊記憶體中[3],所以只要能將網路上某一組 PC 的視訊記憶體都視為同一份相同的記憶體區域,就可以達到螢幕內容共用的虛擬顯示桌面環境。

本論文的目標之一是在 Windows 95 上建立一個抽象層的共用記憶體空間,來達到網路上 PC 群組視訊記憶體的一致性。在這裡我們利用了分散式共用記憶體(Distributed Shared Memory, DSM) [4]的技術,在 PC 群組間建立了一個虛擬的共用視訊記憶體空間(virtual shared video memory space)。

在分散式共用記憶體的研究領域上,要建立一個虛擬的共用記憶體空間,有兩種系統上的軟體實作方法:分別為(1)使用者層次的虛擬共用記憶體空間管理(user-level)[5][6]、(2)作業系統的記憶體管理模組

更改(OS modification)[7]。由於 Windows 95 是一商業性作業系統，因此我們無法以更改作業系統核心程式的方式來內建虛擬用記憶體所須的共用記憶體管理模組；而在使用 user-level 記憶體管理裝置或函式庫呼叫的策略上，需要一連串的系统呼叫，會產生相當多的 context-switch 動作，會影響到共用記憶體系統的效率。由於視訊記憶體的性質是屬於 I/O 卡上的 I/O memory，故在本系統中是利用螢幕顯示卡驅動程式外掛式的標準介面，嵌入一個獨立之視訊記憶體管理裝置(video memory management device)於系統核心內，藉以達成網路上 PC 群組視訊記憶體的分散式共用。

由(圖一)所示，我們利用 Windows 95 VxD(virtual device drive)的觀念，在 VGA 驅動程式與顯示卡間所定義的標準介面中[9]，嵌入一個獨立的系統(system-level)視訊記憶體管理裝置來維持視訊記憶體分散式共用的一致性(shared memory consistency)。此 I/O 裝置的使用，一方面為獨立的系統核心模組但卻免去更改原本的系统核心程式；一方面避免多餘的系统 context-switch 動作對整體的效率有相當大的增益。

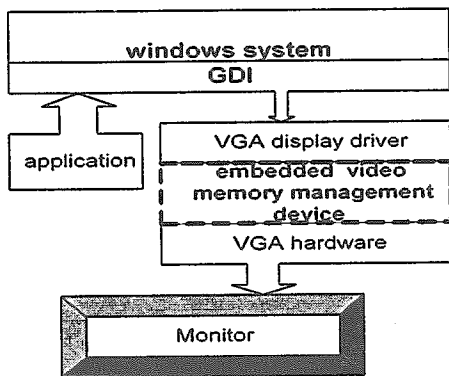


圖 1

### 3. 系統概觀與設計考量

#### 3.1 分散式共用視訊記憶體層

在維持視訊記憶體分散式共用一致性上，我們在系統中提出一個分散式共用視訊記憶體(distributed shared video memory)的通透層，此一通透性之抽象層必須考量和定義到的有：

1. 共用記憶體位址空間(shared memory address space)
2. 記憶體一致性更新顆粒(coherence granularity unit)
3. 一致性策略(coherence policy)
4. 記憶體保護機制
5. 分散式共用記憶體之管理權責(DSM management responsibility)

1.shared memory address space: 在 IBM PC 架構下，存取顯示卡上的視訊記憶體是利用 memory mapped I/O 的方法，在 PC 定址空間上定義了一段 64K 的記憶體空間映射到一個大範圍的視訊記憶體

區域，加上利用 bank-switch 的方法就可以使小範圍的位址空間映射到大範圍的實體 I/O 記憶體。如下圖(二)所示；而共用記憶體的位址空間也就是這 64K(A0000~AFFFF)的記憶體空間。

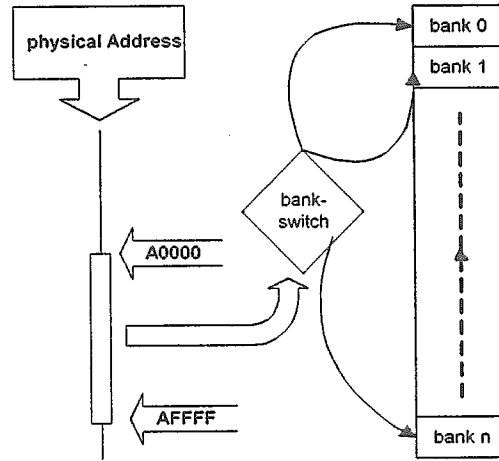


圖 2

2. coherence granularity unit: 記憶體一致性顆粒的大小通常取決於系統設計的環境；硬體導向的系統使用較小的顆粒單位(cache size)，當系統是以軟體導向的解決方案時，取決於虛擬記憶體分頁的記憶體管理方式和記憶體保護原則，通常採取實體記憶體分頁(page size)為顆粒的大小，但在提高一致性效率的前提之下為避免浪費網路頻寬，在本論文中是利用軟體程式的控制來達成 1Kbytes 的記憶體顆粒。在作法上，是將原本 Windows 95 4Kbytes 的分頁切割為 4 個 1Kbytes 區塊，經由與相同位置的視訊記憶體區塊比較運算，過濾出已遭更改過的 1Kbytes 區塊，藉由此種方法就可以得到 1Kbytes 的記憶體更新顆粒，本文在後面將對兩者作一實際測量比較。

3. coherence policy: 視訊記憶體區域隨著螢幕上影像的變化而被更新，其更新動作及區域式對應著螢幕影像的改變；由人類視覺感官刺激與網路頻寬的負荷角度來看，並無法負荷太高的更新頻率，且導致螢幕內容更新的程式執行的回應時間(response time)也並非如此的迅速。基於以上的原因，我們在考量一致性的策略上式採用 time-interval coherence；也就是以一個短的時間間隔(250ms~400ms)頻率來做共用視訊記憶體一致性的更新動作，而非採用即時性的更新。以減少網路的傳輸量；同時也不會使共用視訊記憶體的視覺效果太差。在 write-invalid 與 write-broadcast 的策略運用上我們採用後者，原因是為了避免資料更新動作的遲延。

4. 記憶體保護機制: 共用記憶體的記憶體保護機制是為了維護分散式共用視訊記憶體(PC address space A0000~AFFFF)使用的一致性，並依據前所敘說的一致性策略的原則制定記憶體一致性更新的方法。

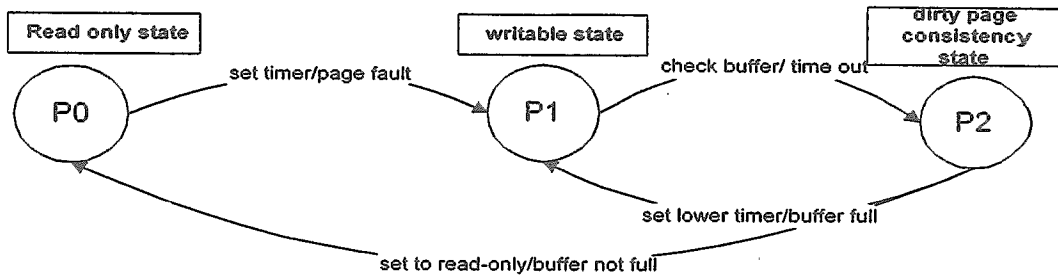


圖 3

在系統中所定義的共用記憶體位址空間上的記憶體分頁之狀態有三(如圖(三)),分別為: 1. read-only state, 2. writable state, 3. dirty-page consistency state. 在初始狀態 P0 所有的視訊記憶體分頁狀態皆為 read-only, 當螢幕影像欲變化而視訊記憶體的某分頁被寫入時, 此時此分頁會產生分頁錯誤 (page fault), 將此分頁設為 P1:writable 並啟動一個預設的等待時間間隔(250ms~400ms); 根據我們所制定的視訊記憶體一致性策略下, 並不作立即性的一致性更新; 而是在等待一個特定的時間間隔後才更新視訊記憶體分頁, 故直到 time-out 此記憶體分頁才進入 P2,執行視訊記憶體的一致性工作也就是更新 PC 群組中其他節點的視訊記憶體分頁資料, 然後再把分頁狀態重設為 read-only。同時我們為了資料的流量控制的理由, 萬一 sending buffer 已滿溢此時就必須押後記憶體分頁一致性的時刻, 以免太多的網路流量造成資料的流失導致傳送的成本增加; 如圖中 P2 到 P1 的轉換。

**5. DSM management responsibility:** 分散式共用記憶體之管理權責, 定義了一個分散式共用記憶體之 PC 群組中負責處理資料一致性相關動作的管理者, 在這裡可分為中央集權式(centralized); 此類方法中是指定一部特定 PC 處理資料一致性的相關動作, 此種方法的優點是簡單, 但容易使系統的瓶頸落在此一特定節點的處理上, 因此有第二種較彈性的方法(分散式管理)提出, 其缺點是設計上較複雜。在本論文中所設計的視訊記憶體共享架構為 Single-writer / Multiple-reader, 是符合中央集權式 (centralized) 管理特性, 複製式(replicated)管理, 故採用前者而非後者之方法。複製式(replicated)管理, 故採用前者而非後者之方法。

**3.2 虛擬顯示桌面環境**

本論文中, 虛擬顯示桌面的定義有別一般 Windows 顯示桌面, 在虛擬顯示桌面的環境之下, 所有桌面上的物件皆為網路上同一群組 PC 所共用、共有的, 不論 local PC 原本顯示桌面的配置環境為如何。所以對虛擬桌面上物件的存取控制規範定義亦是本篇論文所討論的範圍之一。

顯示桌面共用其架構如下頁圖(四); 使用者在螢幕上所看到的共用應用程式根據主動與被動的區分可分為 virtual application 和 real application。後者是應用程式的真正執行端, 會主動顯示應用程式的執行結果於共用的顯示桌面上; 前者是透過顯示記憶體的分散式共用層來被動接收應用程式執行所造成的視訊影像更新。

藉由攔截群組成員操作共用應用程式所產生的系統事件[8](system event, keyboard + mouse)再加以轉送至 real application 端, 經一制訂的控制權轉換策略(floor-policy), 來達到(非本地)應用程式(亦是一個虛擬應用程式)對(本地)應用程式-real application 的控制與操作, 以達到應用程式分散式共用, 共享的效果。再這樣子的一個共用環境中, 我們將應用程式的初始執行端(real application)稱-主控端, 而應用程式影像執行結果的接收端稱-受控端, 而應用程式真正的控制者稱-控制端, 當(受控端)獲得應用程式控制權後, 角色即轉為控制端。

本系統在控制權轉移策略(floor-policy)上提供兩種模式:

- 1. Lecture model: 在本模式中, 只有執行共用軟

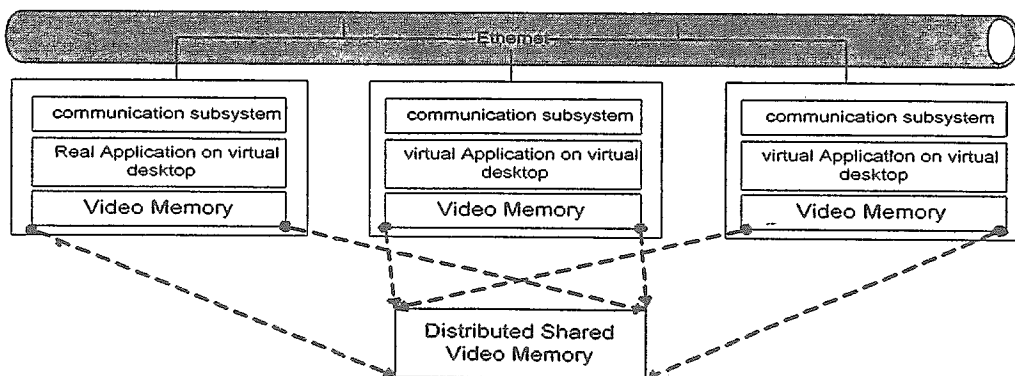


圖 4

體的節點擁有發言權。其餘的節點只能接收由共用軟體所產生的結果。

2. Chaired token passing model: 當使用者欲取得發言權時，必須發出一個要求發言權的 request，request 會依照抵達 sequencer 的先後順序，將發言權給第一個使用者，等目前發言權的持有者讓出發言權時，就由 sequencer 將發言權交給下一個使用者。

#### 4. 系統實作

在本論文中有關分散式共用顯示視訊記憶體裝置實作方面，我們採用了 Windows 95 上虛擬週邊裝置(virtual device)的技術，或可稱為具可嵌入性的延伸 I/O 驅動程式介面獨立模組，來負責共用記憶體的一致性管理；我們針對一個多人使用的環境下對應用程式輸入事件的控管，制訂了一個事件協調與同步模組；而在網路傳輸與群集管理方面亦相對建立了可靠群址傳輸協定，和群集管理模組。另外為了能使本系統應用在遠距教學與虛擬教室之方

理常式，或一個 I/O port 的 hook handler 等；在本系統的運用上，我們在顯示卡驅動程式與顯示卡之間安裝一個可設定(settable)的虛擬週邊，可使用此虛擬週邊對顯示卡的硬體動作產生 trap，藉由此功能可監督管理視訊記憶體的寫入動作，並通知 coherence manager 透過 winsock 的使用來實際負責資料的一致性更新，相反的，coherence manager 也可由一定標準的介面去設定此虛擬週邊所使用之參數（包括 enable/disable 其 VxD 功能），因此稱之為可嵌入之 I/O 介面，它是與作業系統核心模組相獨立的，也就是說透過此方法我們可以在不對作業系統模組修改下增加可用度和方便安裝的效果。

#### 4.2 事件協調與同步模組

本模組中最主要是在處理群組中各節點所發出控制應用程式的訊息，並提供一個協調控制的機制。首先要解決的問題是將 Windows 95 所發出的訊息加以攔截，並轉向至網路輸出，我們利用 Hook 涵式攔截使用者經由鍵盤、滑鼠等操作應用程式硬體所發出的訊息，並透過一個發言權控制（floor control）的策略，將其訊息透過網路傳給欲控制的應用程式。

在本模組中提供了下述功能：

1. Request\_floor: 群組成員要求發言權（應用程式的控制權）。
2. Release\_floor: 應用程式控制端釋放出發言權。
3. Get\_floor: 應用程式的主控端(instructor)要求強制的奪回發言權(應用程式控制權)。
4. Switch\_floor: 應用程式的主控端把應用程式的控制權由目前的發言者(floor holder)，轉移給下一個欲發言者

#### 4.3 群集管理模組

群組管理模組的功能包括整個共用視訊記憶體群集的建立、終止，以及成員的加入離開，當群集的狀態改變時，群集管理模組必須通知視訊記憶體一致性管理模組、事件協調與同步模組以做出適當的反應；在本系統中透過一個 group information server 以應用在遠距教學環境中，對上課成員之群集資訊匯集與提供群組管理服務，並負責群組資源的管理與分配，因應在教學的環境中，我們將群集

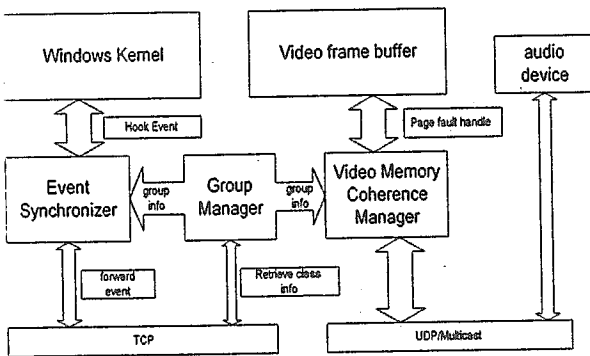


圖 5

向，本系統亦可選擇性的外掛語音播放之功能，其如圖（五）所示，為本系統的一個概觀。

#### 4.1 視訊記憶體共用一致性模組

由圖(六)可知，本系統外掛的虛擬週邊是由一個 callback 函式所驅動，而此 callback 函式被呼叫可能是在 initialization 階段安裝了一個分頁錯誤處

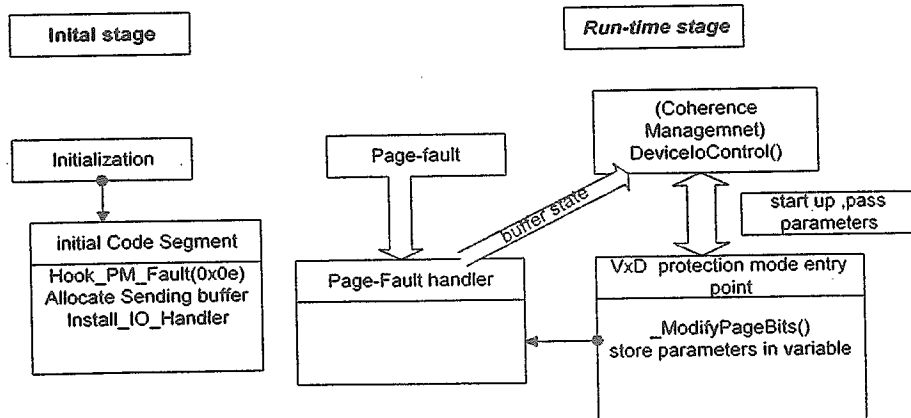


圖 6

之成員屬性分為兩類；一為老師(instructor)也就是課程的開啟者與結束者、一為學生(student)也就是課程的加入者，在我們的群集管理模組下提供了五個 services。

1. Create\_class: instructor 利用此函式呼叫來建立一個新的課程；instructor 必須將建立此課程的相關資料(如：server IP address, multicast IP address used by the class 等)傳送給 class information server，向其註冊，而 class information server 也將送為一個 class ID 以作為此課程的一個標記。
2. Destroy\_class: instructor 下此命令結束正在進行的課程廣播節目，並透過 class information server 通知同一組群的成員。
3. Add\_class: 提供 student 用來加入正在進行或以進行中的課程節目，本系統提供一個動態加入的策略。
4. Leave\_class: 使 student 可以離開進行中或以結束的課程節目。
5. Query\_class: student 可用此服務查詢現今有哪些課程節目正在播放和選擇其一加入。

#### 4.4 群組傳輸協定

因應本系統在應用上群組通訊特性，我們採用 IP multicast 作為底層的傳輸機制，以降低網路上資料的傳輸量。但其所提供的服務品質(QoS)就如傳統的 UDP 傳輸，是屬於 best effort 方式的不可靠傳輸。

因此我們在 IP multicast 上建構一個可靠的傳輸機制。針對本系統的應用，我們提供一個有效的、低負擔的、具有良好的可擴充性的傳輸協定。在協定的設計上，我們採取了 selective repeat 的方式，利用 NACK (negative acknowledgment) 的錯誤處理，由接收端來要求重送錯誤的封包，藉此以降低傳輸協定所必須傳送的訊息數量，同時減少發送端處理確認訊息的負擔，提高系統的可擴充性。再者，本協定以群址傳送方式來傳遞 NACK。

為降低發送端的負擔，本協定採用接收端啟動錯誤處理的方式。每個送出的資料封包都包含了一個序號，當接收端發現資料有誤或是封包傳送的次序錯亂時，即負責發出 NACK 將其懷疑有誤的封包序號傳回給發送端。每個節點再發現錯誤之後必須先等待一段隨機的遲延，才送出 NACK。再傳送 NACK 之前，接收端會檢查是否已有其他的接收端送出該 NACK 如果該 NACK 已有其他節點送出，則沒有必要傳出，相反的，如果 timeout 能為收到重送的封包，則在要求重送；其狀況轉換如下圖(七)

#### 5. 實驗結果與評估

在 3.1 中提到本系統中是利用軟體程式的控制來獲得 1Kbytes 的 coherence unit，在此處的作法是先將欲一致性更新的 4Kbytes 視訊記憶體資料切割成 4 個 1Kbytes 的區塊，在將此區塊與代表相同位址的視訊記憶體內容做 compare，可得出在 4Kbytes

中被新過的 1Kbytes 區塊有哪些，和只更新被寫入過的區塊而不必更新整個分頁。此種方法理論上可大幅減低網路的傳輸量，但是不是可是用於每一種情況中？我們針對此方面作了一個實驗。

#### 實驗環境：

我們播放了兩個預錄的程式使用節目；一個是文字編輯(使用 MS-WORD)，另一個是影片播放(AVI)，利用這兩個播放的節目來比較在 coherence unit 為 4Kbytes 的環境中和 1Kbytes 的環境中，本系統伺服端(real application)所傳送出的網路傳輸量情形，實驗結果如表(一)與表(二)所示。

1. Text Editor (MS-WORD)

Coherence unit \ Time interval	4K	1K
300 ms	13904K	8972K reduce 35%
400 ms	12160K	8456K reduce 30%
500 ms	11472K	8231K reduce 28%

表(一)

2. AVI movie

Coherence unit \ Time interval	4K	1K
400 ms	18352K	18298K reduce 0.2%
500 ms	18160K	17872K reduce 1.5%

表(二)

#### 實驗評估：

在 1. Text Editor 中的實驗結果得知，使用 1K bytes 的 coherence unit 可以獲得 30% 左右的網路傳輸量節省，但在 2. Movie Play 中這兩種 coherence unit 所表現出的是相近似的結果。由於 Text Editor 動作的特性只會更新(寫入)到某一局部的小範圍視訊記憶體區域，所以使用 1K bytes 會得到較好的效果；反之在 2. Movie Play 中，影片播放的特性會更改到的視訊記憶體(顯示螢幕)範圍是連續且大量的，故使用 1K bytes 的 coherence unit 並不會獲得多大的效果。

為了具體評估整體系統實用上的效果，我們在此做了一個實驗加以評斷。以本系統的架構下我們認為會影響系統效能的因素有兩方面：機器執行速度，網路傳輸。

#### 實驗環境：

我們在三間實驗室共架設 7 部 PC (800x600 VGA mode) 的共用視訊記憶體群組，藉以觀察在本系統的運作下網路封包傳輸的 lost rate 狀況。此 7 部 PC 的組成為：

- A1. Pentium+PCI network card (伺服傳送端)
- B1. Pentium+PCI network card (接收端)
- B2. Pentium+PCI network card (接收端)

- B3. Pentium+PCI network card (接收端)
- B4. 486+ISA network card (接收端)
- B5. 486+ISA network card (接收端)
- B6. 486+ISA network card (接收端)

實驗結果:

A1 共送出 18174 個封包。

PC \ Items	接收的封包數目	NACK 數目	Lost rate
B1	18072	52	0.5%
B2	18164	10	<0.5%
B3	18164	10	<0.5%
B4	17014	71	4%
B5	18105	26	<4%
B6	18009	45	<4%

- PS.1 使用 500ms time interval。
- PS.2 所撥放的節目為 powerpoint 投影片播放。

6. 結論

在本論文中，我們提出了一個利用視訊顯示記憶體之分散式共享的方法，來達成顯示螢幕共用之系統，且以遠距教學的例子實作來證明此方法之可行性；並以 MS-Windows95 為其發展平台，使其能獲得較豐富的應用軟體支援。

由於 MS-Windows95 的核心模組並不支援共用記憶體之記憶體管理，所以我們在 MS-Windows95 利用顯示卡與其驅動程式間所定義之標準介面，嵌入一個對 I/O memory(視訊顯示記憶體)的管理裝置來達到對 video memory 的分散式共用。在一致性的模型中較特殊的地方，我們加入了一個時間顆粒(time granularity)的觀念，我們針對視訊記憶體快速頻繁的寫入動作為避免過大的傳輸量，故我們允許共用視訊記憶體是有一個固定時間間隔是不一致的。同時我們也針對是用環境的需求調整此一時間間隔，使得此一 delayed-update 的情形不會影響到使用的流暢度。

7. 參考文獻

[1] John Eric Baldeschwieler, Thomas Gutekunst, Bernhard Plattner, "A survey of X Protocol Multiplexors", ACM SIGCOMM Computer Communication Review. Vol. 23 No. 2 April 1993

[2] J.-Huang, W.-H. Tseng, M.-J. Ding, Y.-S. Su and L.-C. Wu, "VirtualTalker: An On-line Multimedia Systems on Token Passing Network," IEEE Trans. On Consumer Electronics. Vol. 39. No. 3. pp. 609-618, Aug. 1993

[3] Richard F. Ferraro "Programmer's Guide to EGA, VGA, and Super VGA Cards", chapter 13

[4] Jelica Protic, Milo Tomasevic, and veljko Milutinovic, "Distributed Shared Memory: Concepts and Systems", IEEE Parallel and Distributed System Magezine , pp.63-79, Summer 1996

[5] K. Li, "IVY: A Shared Virtual Memory System for Parallel Computing", Proc. Int'l Conf. Parallel Processing, IEEE Computer Society Press, Los Alamitos, Calif, 1988, pp.94-101

[6] S. Zhou, M. Stumm, and T. McInerney, "Extending Distributed Shared Memory to Heterogeneous Environments," Proc. 10<sup>th</sup> Int'l Conf. Distributed Computer Systems, CS Press, 1990, pp.30-37

[7] B. Fleisch and G. Popek, "Mirage: A Coherence Distributed Shared Memory Design," Proc. 14<sup>th</sup> ACM Symp. Operating System Principles, ACM Press, 1989, pp.211-223

[8] 李金溪, "以遠端同步執行為基礎之遠程教學系統", 中央大學資工所碩士論文