分散式共用視訊記憶體之設計與實作之虛擬顯示桌面系統

方 立　曾黎明　　陳奕明

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

TEL: (03) 422-7151 EXT. 7523　　TEL: (03) 422-7151 EXT. 6524

fanton@dslab.csie.ncu.edu.tw　t341727@ncu865.ncu.edu.tw　cym@im.mgt.ncu.edu.tw

摘 要

本論文主要是設計與實作一個以分散式共用記憶體技術爲基礎的遠距顯示幕視訊傳送系統。針對 PC 螢幕顯示採外掛式顯示介面,並利用記憶體映射輸入/出 (memory-mapped I/O)的特性,將網路上的某一群組 PC 的所有視訊顯示記憶體都是爲同一個 DSM (Distributed Shared Memory) ,來達到網路上 PC 群組之顯示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附加處理，來達成多人輪流遠端控制桌面應用程式之互動介面，藉以實現真實的分散式顯示幕共用。

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)、視訊會議 (Video Conference)、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爲三類：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。第二類的系統稱爲共用視窗系統(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]作爲 X server 與 X client 的中介。第三類技術爲特殊設計的群體合作軟體[2]。

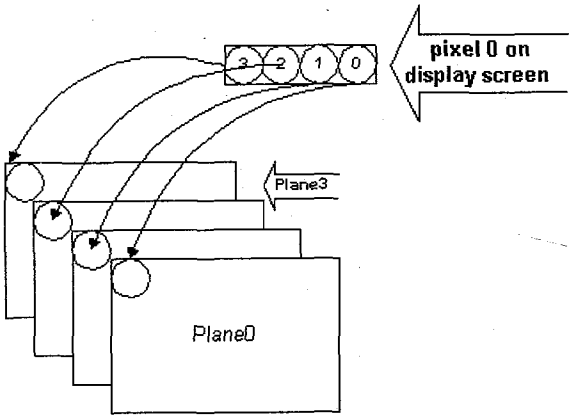
根據上述的背景說明來看，過去所開發的顯示幕內容共用技術，若需要特殊的設備支援，則其無法普遍的適用於廣域網路環境;若建構於 X Window 之上,則應用程式的普及率會使其實用性受限制；若屬於特殊設計的群體合作軟體，其所能提供的能力僅止於該程式。針對這些不足之處，我們提出一個無須硬體設備支援，並且與應用軟體無關的顯示幕內容共用之虛擬顯示桌面系統。同時，我們將開

發平台設定爲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

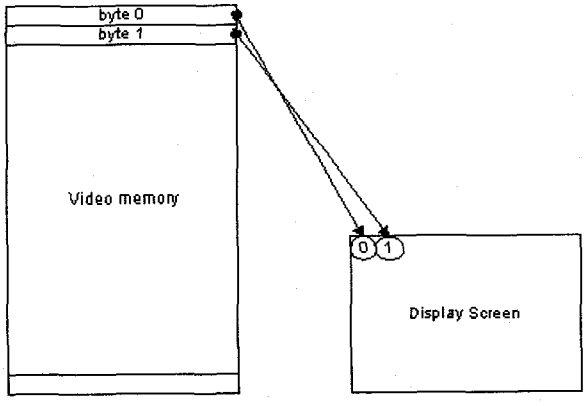
PC 上螢幕顯示的輸出是以顯視卡(VGA display card)上的視訊顯示記憶體(video memory)當作輸入之視訊緩衝區，所以透過一個視訊顯示記憶體的分散式共用即可以達到一個螢幕內容共用的虛擬顯示桌面系統。

螢幕上所展現的圖素(pixel)在視訊記憶體中所存放的方式，依螢幕解析度的不同而有不同的排列格式。其分爲兩種[3]，第一種稱爲(bit-plane)，使用在解析度爲 16 色的視訊解析度上，在此種的排列方式中視訊記憶體可看成四個相重疊的 64Kbytes 平面(plane)，對於螢幕上一個由四個位元表示的圖素而言，每個不同位置的位元放在相對應的視訊記憶體平面中，如下圖(一)所示。



圖(一） bit-plane format

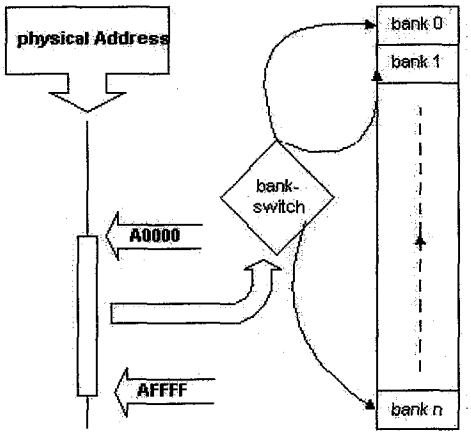
第二種稱爲(packed-byte)，使用在 256 色以上的螢幕解析度上，影像圖素在這種排列方法中視訊記憶體不再被分爲四個重疊的平面，而是一般正規的連續架構，在圖(二)中表示了 256 的繪圖模式下(一個圖素由八個位元，也就是一個位元組來表示)圖素的存放方式，其中每一個圖素的值由一個位元組表示，圖素 0 擺在視訊記憶體中第一個位置，接連而下的就是圖素 1 。



圖（二） packed-byte format

IBM PC 的架構下，在系統的定址空間中定義了 64Kbytes 的位址空間(記憶體位址 A0000~AFFFF)來作爲 VGA 卡上視訊記憶體之存取位址使用，但一般 VGA 卡上之視訊記憶體的大小並不只 64Kbytes ，

通常可以是 1MB 到 4MB 之間，因此光靠 64K 的位址空間並不足夠，所以此處 64K 的定址空間必須能夠分別映射(mapping) 到 1MB~4MB 的實體 I/O 記憶體空間(video memory)上才行；在這裡顯示卡使用了區塊切換(bank-switch) 的技術。我們可把視訊記憶體視爲一個連續的空間，而這連續的空間上分割了接連的 64Kbytes 大小的區塊(bank ,or plane)；在存取這一個連續的記憶體空間時首先要指定所要存取的記憶體區塊編號，再藉由 PC 上 64K(65536)之位址空間來存取這一指定區塊內之資料，藉由這種區塊切換的方法，就可以使小範圍的記憶體定址空間映射到大範圍的實體記憶體中，如下圖(三)。



圖(三) bank/plane switch

本文中針對PC 顯示卡上視訊記憶體爲一I/O記憶體的特性,與Intel處理器virtual memory之架構，建立一個虛擬之共用視訊記憶體空間 (virtual shared video memory space)，藉由此裝置可達成網路上 PC 群組之顯示桌面、視窗共享的效果。

在分散式共用記憶體(Distributed Shared Memory)的研究領域上，要建立一個虛擬共用記憶體空間有兩種系統上軟體實作的方法[4]，分別爲(1)使用者層次的記憶體管理(user level)[5][6]、(2)更改作業系統模組(OS modification)[7]。由於 Windows 95 是一商業性的作業系統，因此我們無法以更改作業系統核心的方式來內建虛擬共用記憶體所需的記憶體管理裝置，而使用 user-level 的記憶體管理裝置的策略上，需要一連串的系統呼叫，會產生相當多 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，故在本篇論文中利用螢幕顯示卡驅動程式採用外掛式介面之標準，使我們可以嵌入一獨立之共用 I/O 視訊記憶體介面裝置(shared video memory I/O device)的管理模組於作業系統核心。

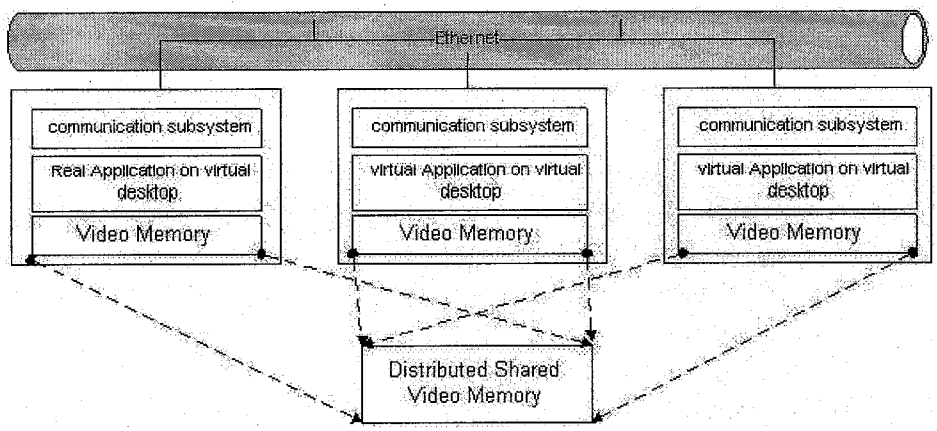
3、系統概觀

論文的目標之一就是提供一個具抽象通透性(transparency)的視訊記憶體共用層，來達成共用顯示螢幕內容，也就是螢幕上一般個人應用程式視窗的共享、共用（本論文中所謂之群組或群體實際是代表網路上共用視訊記憶體之節點的成員集合）。此一通透性之抽象層還必須考量和定義到的有: (1)記憶體一致性更新顆粒;(2)一致性策略。

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)爲顆粒的大小; 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,原因是爲了避免資料更新動作的遲延;。在本論文中所設計的視訊記憶體共享架構爲 Single-writer / Multiple-reader ，是符合中央集權式（ centralized ）管理特性，而非複製式(replicated)管理。

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實體與虛擬的區分可分爲 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視窗的執行結果於共用的螢幕桌面上；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。



圖（四）應用程式共用架構

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，角色即轉爲控制端。

本系統在控制權轉移策略(floor-policy)上提供兩種模式：

- Lecture model

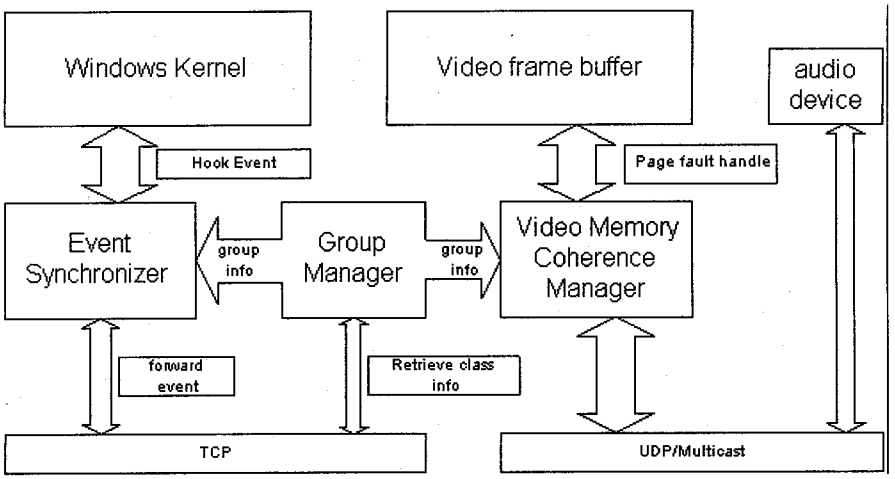
在本模式中，只有執行共用軟體的節點擁有發言權。其餘的節點只能接收由共用軟體所產生的結果。

- 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實作方面，我們採用了 MS-Windows 上虛擬周邊 (virtual device) 的技術，或可稱爲具可嵌入性的延伸 I/O 驅動程式介面獨立模組，並配合我們所設計之共用記憶體一致性模型，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。另外爲了能使本系統應用在遠距教學與虛擬教室之方向，本系統亦可選擇性的外掛語音播放之功能，其如圖（五）所示，爲本系統的一個概觀。



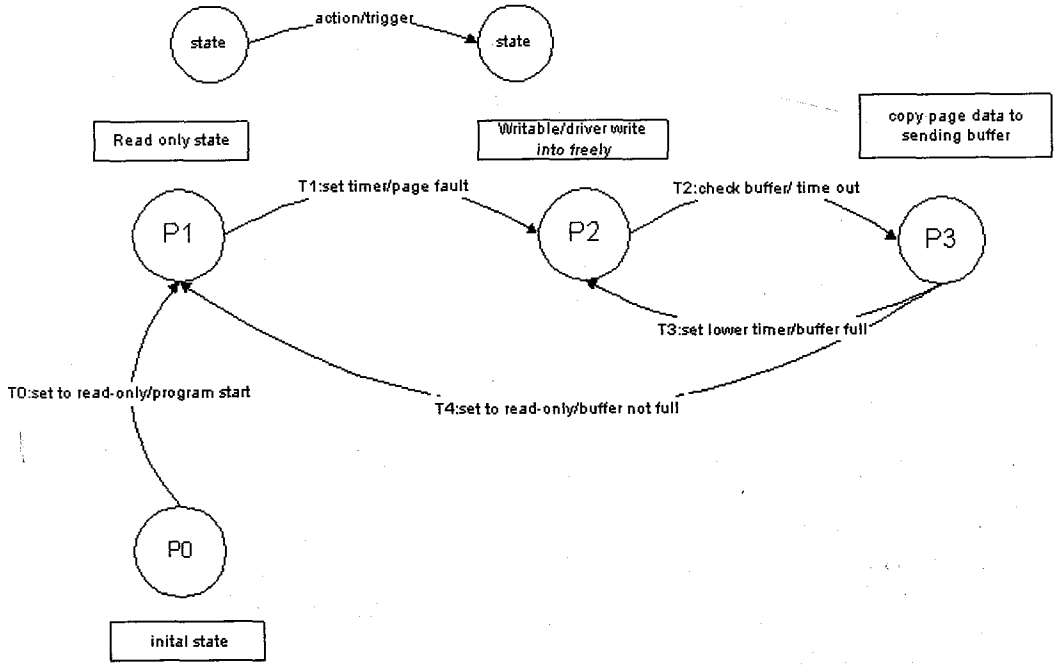
圖(五) 系統架構圖

4.1 、視訊記憶體共用一致性模組

在技術上我們利用視訊記憶體位址空間所產生之分頁錯誤(page fault)，來偵測視訊記憶體被寫入的動作；在初始狀態下我們將 A0000 到 AFFFF 位址空間所處的虛擬記憶體分頁設爲 read-only ，當某一 read-only 記憶體分頁被寫入產生 page-fault 時，將其設爲 writable 使其能夠寫入,當一段可接受的固定時間間隔後，再將先前之分頁設爲 read-only ，並將此一分頁資料更新. 因應人類視覺的感官刺激是可容許些微的 delay,故不需要做即時性的視訊資料更新；這樣一個視訊記憶體;分頁錯誤-一致性更新-read-only 的循環使我們可以維持整個視訊記憶體的分散式共用，和只傳輸被更改過的視訊記憶體區域。視訊分頁控管流程如下頁圖(六)

- state P0 ：初始狀態。
- transition T0 ：程式啓動；將所有的視訊記憶體分頁設爲 read-only 。
- state P1 ：所有的分頁處於 read-only 狀態。
- transition T1 ：若顯示卡驅動程式對分頁號碼 A0 到 AF 中的任一分頁 i 做寫入動做時，產生分頁錯誤的例外(fault exception 0x0e)，把此分頁設爲 writable 並設定一等待的時間間隔。

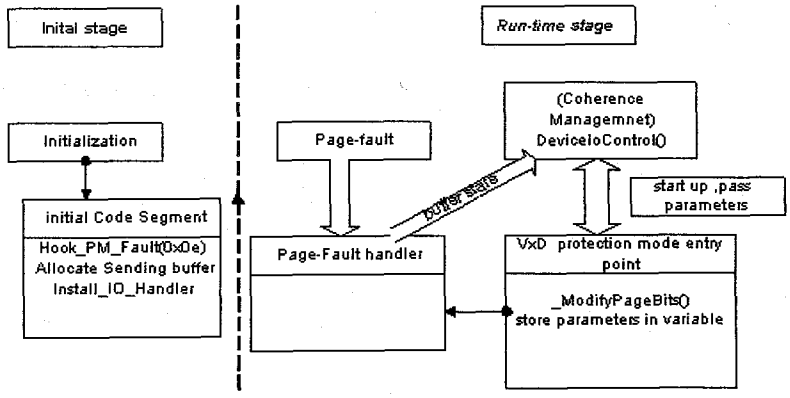
- state P2：此分頁處於 writable 狀態，在此一所設定的時間間隔內此記憶體分頁可被更改寫入，爲了知道被寫入的是視訊記憶體中哪一個 64K 的區塊(bank/plane)，在此 state 中還必須追蹤、記錄顯示卡 bank-switch 的動作。
- transition T2 ：在 T1 所設的時間間隔已達到(time-out)，並檢查 sending buffer 所剩餘的空間。
- state P3：預備將此分頁資料放入 sending buffer，在這個處理狀態中有兩種情況；(1) 若 sending buffer 未滿溢，將分頁 i 的拷貝送至 sending buffer 中，等候被送出更新，以達到共用視訊記憶體的一致性；(2) 若 sending buffer 已滿溢，則降低分頁 i 原本的時間間隔，經由 transition T3 的轉換到 P2 繼續等候處理。
- transition T3 ： sending buffer 若已滿溢，暫時不將此分頁更新，這是爲避免網路流量 burst 所造成的資料大量遺失，及顧及接收端的處理速度而採取的一種流量控制措施。
- transition T4 ： sending buffer 未滿溢，則將此分頁的資料放入 sending buffer 中，狀態爲初始狀態(read only)。



圖(六)視訊分頁控管

在本文中所敘論的可嵌入的延伸驅動程式 I/O 介面，即爲 MS-Windows 系統虛擬週邊（ Virtual Device-VxD ）。由下頁圖(七)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系統 initialization 階段安裝了一個分頁錯誤處理常式，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，因此我們由上觀念中，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，並通知並通知上層的 coherence manager 模組透過 winsock 的使用來實際負責

資料的一致性更新，相反的， coherence manager 模組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），因此稱之爲可嵌入之 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。



圖（七）系統 VxD 架構

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 函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：

- Request_floor( )

群組成員要求發言權（應用程式的控制權）。

- Release_floor( )

應用程式共用架構下的控制端釋放出發言權。

- 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)。

- Switch_floor( )

應用程式的主控端(instructor)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，我們將群集之成員屬性分爲兩類；一爲老師(instructor)也就是課程的開啓者與結束者、一爲學生(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 也將送爲一個 class ID 以作爲此課程的一個標記。

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( )

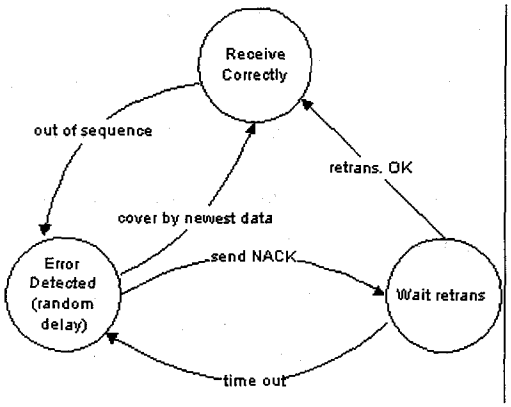
student 可用此服務查詢現今有哪些課程節目正在播放和選擇其一加入。

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 作爲底層的傳輸機制，以降低網路上資料的傳輸量。雖然 IP multicast 提供了一個有效的傳輸機制，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。

因此我們必須在 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 ，且要求接收端在送出 NACK 之前，先等待一段亂數的延遲時間，若此段時間內有其他節點送出了含有相同序號的 NACK ，則該 NACK 即不送出如此不但可以避免同一時間內有大量的 NACK 產生，並能進一步地減少訊息的傳輸。

爲降低發送端的負擔，本協定採用接收端啓動錯誤處理的方式。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 能爲收到重送的封包，則在要求重送；其狀況轉換如下圖(八)。



圖(八)協定意示圖

5 .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 爲其發展平台，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爲避免過大的傳輸量，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。

對於本系統而言尚有一些須改進的部份、首先我們希望能透過對視訊資料的壓縮減少資料的傳送量，這樣相對的可以減低經由網路傳輸所產生的 delay 和封包遺失所需的錯誤處理措施。另外一方面在一致性模型中所採取之”固定時間間隔”更新的方式上，我們提出對於此時間間隔的設定可依視訊記憶體被存取的頻率，使用和目前網路的狀況，動態的作出合適的調整，以符合隨時變化的使用環境作出一最佳化的設定。

6.參考文獻

[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 ” Programer ’ s Guide to EGA,VGA,andSuper VGA Cards ” ,charpter 3

[4]Jelica Protic,Milo Tomasevic,and veljko Milutinovic, " Distributed Shared Memory:Concepts and Systems " ,IEEE Parallel and Distributed System Magezine , pp.63-79,Sunmmer 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. 10th 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. 14th ACM Symp. Operating System Principles, ACM Press, 1989, pp.211-22

[8]李金溪, "以遠端同步執行爲基礎之遠程教學系統" , 中央大學資工所碩士論文,July 1996