

在 WLAN 中支援無縫群播視訊換手之多重時差群播通道傳送

機制

童曉儒

國立屏東科技大學資訊管理系

strong@mail.npust.edu.tw

洪振璋

國立屏東科技大學資訊管理系

M9456029@mail.npust.edu.tw

余遠澤

高雄醫學大學醫療資訊管理學系

yuyt@kmu.edu.tw

謝孝禧

國立屏東科技大學資訊管理系

M9556029@mail.npust.edu.tw

摘要

在無線網路下，無縫串流換手的群播服務因各種換手減損的因素而受到極大的挑戰，像是高位元錯誤率、通訊中斷以及不完全的串流銜接。為解決這樣的問題，我們提出一個「多重時差群播通道機制」(Multiple-Time-shifted-Multicast-Channel)，基本理念是存取路由器(AR)複製串流資料在多個無線群播通道上，並藉由時間差的排程方法來傳送串流資料，而行動主機(MH)同時接收這些串流。這個機制能在換手的過程中，藉由增加錯誤回補率以及提供資料緩衝區暫存機制來幫助 MH 達成幾乎無縫的播放品質。透過此機制我們探討了幾個設計因素間的關係像是 AR 與 MH 緩衝區大小、channel 數量、網路延遲變異度以及錯誤率等。從結果顯示，在群播服務中 AR (MH) 大約配置 2~3 秒的緩衝區容量，同時使用 3

個時差通道通常能在一般 WLAN，transmission error zone 範圍 (<1.5sec)的環境外，達到無縫串流換手。

關鍵字:群播串流、串流換手、緩衝區管理、無線行動網路

Abstract

Title of Thesis: A Multiple – Time – Shifted – Multicast - Channel Transmission Scheme for Supporting Seamless Multicast Video Handoff in WLAN. The seamless stream handoff for the multicast services in wireless networks is challenged by various handoff impairments, such as high bit-error-rate, connection disruption and defective stream cascading. We tackle this problem by proposing a Multiple-Time-shifted-Multicast-Channel

scheme, where an access router (AR) replicates a stream over multiple wireless multicast channels with time-shifted transmission schedules, and a mobile host (MH) subscribes to all of them. The scheme helps a MH to keep hiccup-free playback during the handoff process by enhancing error recovery rate and providing data buffering. The tradeoffs among several design factors, such as the AR and MH buffer sizes, the number of channels subject to the network delay variance and error rate are analyzed through simulation. The results reveal that three time-shifted channels are usually capable of fixing the handoff impairments and achieving the seamless stream handoff for the multicast services most of the time given an AR (MH) buffer size of two or three seconds.

Keywords: multicast streaming, stream handoff, buffer management, wireless mobile network

一、緒論

近年來 Internet 已成為民生必需品，在網路技術蓬勃發展與資訊爆炸的現代，資訊應用多元化使得全球吹起了一股網路風暴，此股網路風潮同樣的也帶動了行動網路與 Internet 整合的需求，最近更發展至寬頻

及遠距的無線網路連線技術，所提供的頻寬已能從過去的數 Kbps 發展至現今的數 Mbps 的傳輸頻寬，由於無線網路的發展已漸漸能補足過去頻寬不足的問題，因此無線傳輸的資料型態已能從過去簡單的文字、圖片資料，進而能傳輸多元化的多媒體資料，如聲音、影像等，也使得目前在 Internet 上豐富多樣的資訊服務能迅速的傳送至行動用戶端(mobile host/MH)。並且為因應行動式線上影音傳輸服務的熱潮，多媒體視訊串流(video streaming)技術已成為各方學者所關注的議題，在許多重要的應用中，如隨選視訊、電視廣播、新聞報導、旅遊諮詢以及商業廣告等，成為無線網路傳輸服務的主流。

無線區域網路(WLAN)，係以電腦或行動設備透過無線網路卡(wireless card)結合存取橋接器(access point)來進行網路資源的利用，用戶端的網路線路傳輸是以無線傳輸的方式，而它具備了有線網路所欠缺的行動性，但也受到用戶端與 AP 之間的距離限制，距離 AP 愈遠，所接收的訊號愈差，因此只能在區域範圍內才能連上網路，而當一端的 AP 訊號愈來愈弱，而另一端的 AP 訊號愈來愈強時，即會進行 layer 2 換手(L2 handoff)的動作，其作法是透過 MH 向新的 AP 進行註冊並建立通道，等待新 AP 準備就緒後，即與舊 AP 結束通道，並將未傳給 MH 的資料轉向給新的 AP 傳給 MH，而此換手的動作是由 MH 主動啟動的，故此作法也稱為 mobile controlled handoff (MCHO)[9]。由於技術不斷的突破，從早期的 IEEE 802.11b/g 的建立，至今已發展出 IEEE 802.11n，不但傳輸效能大大的提升，實際傳輸距離也提高，而 IEEE 802.11n 增加 MIMO (multiple-input multiple-output) [10] 的標準，即使用多個發射及接收的天線來達到更高的資料傳輸率，並且能抑制訊

號干擾的問題，因此對於未來無線網路的發展將可望超越傳統的有線網路，成為應用服務最廣的傳輸媒介。

以多媒體傳輸技術而言，「視訊串流技術」即表示用戶端不須將影片完全下載完，而是透過邊收邊播的模式來播放影片，然而由於 Internet 網路是屬於共享資源的媒介，且 Internet 的頻寬不穩定常造成串流傳送的延遲，而透過此技術可先在用戶端的電腦上建立一個緩衝區(buffer)，於播放前預先下載一段影片片段作為緩衝，當網路實際連線速度小於播放所耗用的速度時(如 MPEG-1 為 1.5Mbps、MPEG-4 為 56Kbps)，播放器就會取用這一小段緩衝區內的資料，避免播放的中斷，以提升播放的品質。

當伺服器欲提供廣播性質的節目內容給廣大的使用者時，為了節省頻寬的耗用，一般常用的作法是採用 IP Multicast 的技術，簡單的說，就是媒體伺服器只需要發送一個串流，即能將此串流廣播至 multicast group，而同屬相同 multicast group 的成員即能共享這一個串流，因此就能減少網路上傳輸的封包量及提高傳輸效率。一個 multicast tree 主要由 video server 及多個中繼器(Access Router/AR)所構成，且傳輸方式是以 multicast (class-D) IP address 從 video server 將 stream 傳送至枝節點的 ARs，而每個 AR 也會將該 stream 廣播至位於相同區域(subnet)的 APs 並且至少有一個 MH 從 subnet 上接收 stream。在 MH 的移動過程中，如圖 1-1 所示，將遭遇 AP 間換手的程序，如果兩個 AP 是屬於相同 subnet，即屬於 L2 handoff，其運作方式為先前所提之 MCHO 來進行換手，且延遲時間較短，但如果 AP 是屬於不同的 subnet，除了 L2 handoff 外，這也會觸發 MH 的 IP 位址必須跟著變動，即 MH 在新的 AR 取得新的位址(Care-of Address/CoA)，而新 AR 會通知舊

AR 目前 MH 落在它所管轄的區域內並將 CoA 更新至舊 AR，並且新 AR 會產生一個 multicast protocol 來重建 multicast tree 來接收 stream，因此當有 stream 在傳送時，則會先由舊 AR 轉送 stream 至新 AR，而新 AR 則會將接收到的 stream 廣播至它所屬的 subnet，而位於新 AR 之 subnet 的 MH 即能接收到此 stream，完成 layer 3 換手(L3 handoff)，其運作是透過 Mobile IP(MIP)[5] 的方式來進行換手，且延遲時間長。然而在無線群播網路的環境下，串流資料常因為伺服器所傳送的路徑有所不同，而所遭遇的延遲情況也不盡相同，從圖 1-1 可以看出 AR 在接收 stream 時產生了非同步情況(舊 AR 接收到 stream 8，新 AR 接收到 stream 9)，並且當 MH 在進行 L3 handoff 時，可能在接近換手點的區域內，將遭遇信號減損所造成的 packet loss，這種現象稱為「transmission error」。而當進入換手點時，此時通訊是處於中斷的情況，因此也會有 packet loss，這種現象稱為「handoff loss」。在換手後因 AR 傳送的非同步情況而造成播放點銜接不上，這種問題稱為「AR synch error」。以上這些問題都將造成視訊播放上的嚴重停格或跳格，影響其播放品質，這些問題統稱為群播網路環境下的「視訊換手(video handoff)」問題。

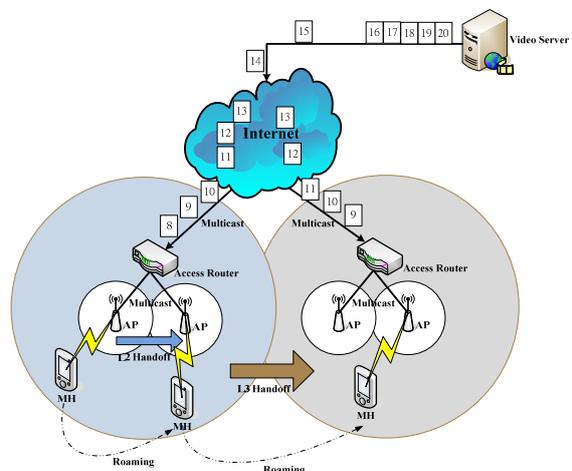


圖 1-1 無線群播網路下之換手示意圖

許多研究結果指出，在傳統的無線網路環境下，透過 MIP 結合 IP multicast 的架構已能實現無縫群播服務，如 IETF RFC 所定義的 remote subscription(RS)及 bi-directional tunneling(BT)[13]，而本研究特別針對在無線群播網路之串流傳送的過程中，handoff 所產生的問題來進行探討與分析，但由於 L2 handoff 的延遲時間幾乎為零，因此本研究主要針對 L3 handoff 的問題提出有效的解決方案，以期提升視訊播放的品質，並達到無縫換手(seamless handoff)。

二、 研究架構

2.1 基本假設與架構

在 mobile multicast 網路環境下，當行動使用者(mobile host/MH)在此環境下點播影片觀賞時，發出的訊息會經由無線路由器(access router/AR)向 video server 要求影片片段，之後經由 multicast 將視訊串流(video stream)以 CBR 的方式傳送至 multicast IP，而 MH 則透過此 multicast IP 來接收並觀賞影片。在傳送端(video server)我們假設將一部影片封裝成多個 UDP 封包的 stream 來傳送，並且我們定義以封包傳送的間隔作為一個時間單位，而封包的 packet id 表示為 $\Sigma = \{1, 2, 3, \dots, N\}$ ，packet size 表示為 p ，封包傳輸率表示為 r packet/s。而正在觀賞影片的 MH 在移動的過程中，可能在接近換手點時，將造成訊號的減損而有大量的 bit error 情況，我們將此現象稱為「transmission error」。而當 MH 進入換手點時，此時通訊是處於中斷的情況，因此會造成 packet loss，這種現象稱為「handoff loss」。而在換手後，由於 video server 在傳送 stream 時，因傳送的路徑及延遲度的不同，因此造成 AR 在接收 stream 時的不同步現象，在此我

們稱之為「AR synch error」。這些問題都將使得影片播放時的不順暢而嚴重影響視訊播放品質，而為了改善這個問題，本研究提出了 Multiple - Time - shifted - Multicast - Channel(MTMC)的機制以多道 multicast channels 並且結合時間差的方式來傳送 stream，其架構如圖 2-1 所示，在 MTMC 的架構下，本研究將以上一章所探討到的 remote subscription 機制為基礎，其主要是從 video server 至多個 AR 間建立一個 multicast tree，此 tree 是建構在 wired Internet 之上，並且我們分別在 AR 及 MH 上配置一個 FIFO Queue 來暫存封包，其 buffer 所能容納之封包個數分別表示為 B_{AR} 及 B_{MH} 。AR 將最近收到的一段 stream 暫存於 buffer 中，傳送時則將 stream 複製在不同的 multicast channel 上，並將傳送時程彼此錯開，而 MH 則同時接收來自不同 multicast channel 上的 stream，並將其儲存至 buffer 中，當視訊換手的動作發生時，將造成網路傳輸中斷，buffer 中的資料可暫時供給 MH 的 stream 播放。

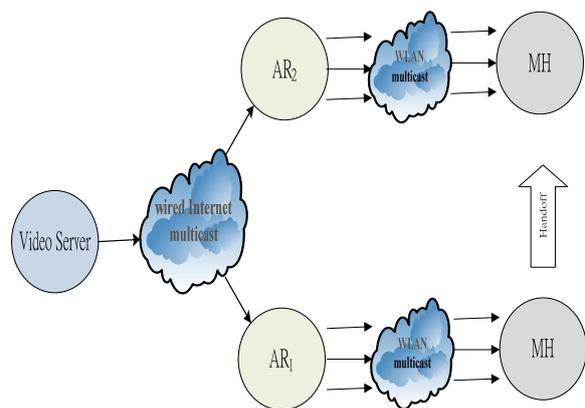


圖 2-1

Multiple-Time-shifted-Multicast-Channel 之架構圖

茲將本論文中所使用之變數及其定義彙整於表 2-1 中以供參考。

表 2-1 變數及其定義

| 變數名稱 | 定義 |
|----------------|--|
| B_{MH} | MH buffer 所能儲存的封包量 |
| B_{AR} | AR buffer 所能儲存的封包量 |
| $f_{AR}(t)$ | 在時間點 t 時，位於 AR buffer 的起始存放位置之 packet id |
| $f_{MH}(t)$ | 在時間點 t 時，位於 MH buffer 的起始存放位置之 packet id |
| c_i | AR 傳送之第 i 個 multicast channel |
| $f_c^{(i)}(t)$ | 在時間點 t 時，被 c_i 傳送出去的 packet id |
| p | packet size |
| r | 連續封包的傳輸率(packets/s) |
| k | multicast channel 數量 |

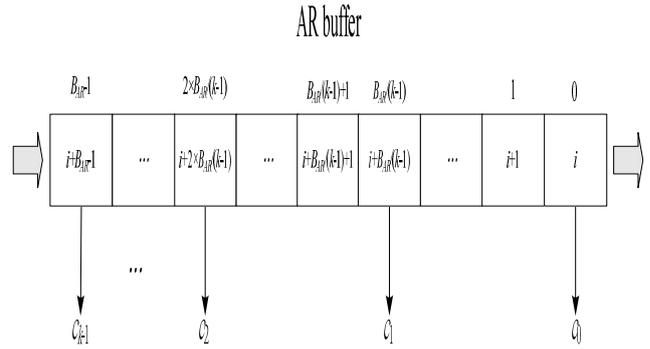


圖 2-2 AR 傳送封包示意圖

我們假設在時間 t 時，被 c_j 傳送出去的 packet id 函數為 $f_c^{(j)}(t)$ ，此函數可表示為

$$f_c^{(j)}(t) = f_{AR}(t) + j \times B_{AR} / (k-1) \quad (1)$$

舉例來說，假設 $B_{AR} = 2000, k = 3, f_{AR}(t) = 100$ ，則 $f_c^{(0)}(t) = 100, f_c^{(1)}(t) = 1100$ 以及 $f_c^{(2)}(t) = 2100$ ，其代表在時間 t 時， c_0 將封包 100 傳送出去， c_1 將封包 1100 傳送出去， c_2 將封包 2100 傳送出去。

2.2 MTMC 之 AR 運作流程

本節主要探討 AR 端的運作流程，首先初始狀態，在 AR 端的 FIFO queue 內無任何封包，假設有一個 MH 向 video server 訂閱某個 stream，當 video server 開始傳送 stream 時，此時 AR 開始接收封包，並將其暫存至 buffer 中，等待 AR buffer 存滿資料後，即開始以下的傳送步驟。

假設在時間 t 時，位於 AR buffer 起始存放位置(位置 0)的 packet id 表示為 $f_{AR}(t)$ ，則 AR 所能持有封包之最大 packet id 為 $f_{AR}(t) + B_{AR} - 1$ ，在這兩端點間的所有封包將被分配至 k 道 multicast channels 上來傳送，其表示為 $c_0, c_1, c_2, \dots, c_{k-1}$ 。假設 $f_{AR}(t) = i$ ，配置狀況如圖 2-2 所示， c_0 傳送封包 i ， c_1 傳送封包 $i + B_{AR} / (k-1)$ ， c_2 傳送封包 $i + 2 \times B_{AR} / (k-1)$ ，以此類推到 c_{k-1} 傳送封包 $i + B_{AR} - 1$ 。

2.3 MTMC 之 MH 運作流程

接著我們說明 MH 端的運作，首先初始狀態，在 MH 端的 FIFO queue 內無任何封包，而後 MH 會傾聽所有的 multicast channels，並接收 packet id 落在 MH buffer 所能容納範圍內的所有封包，而後 MH buffer 在接收封包時，會將其暫存至 buffer 中，等待 MH buffer 存滿資料後，播放器即開始消耗 MH buffer 內的資料。我們假設 MH buffer 存放第 i 個 multicast channel 所傳送的資料其位置函數表示為 $L(i)$ ，存放方式如圖 2-3 所示，位置 $B_{MH} - 1$ 儲存 c_{k-1} 的封包，位置 $B_{MH} - 1 - B_{AR} / (k-1)$ 儲存 c_{k-2} 的封包，位置 $B_{MH} - 1 - 2 \times B_{AR} / (k-1)$ 儲存 c_{k-3} 的封包，以此類推，因此我們可以推導出以下 channel id 與資料放置位置之關係式

$$L(i) = B_{MH} - 1 - (k-i) \times B_{AR} / (k-1) \quad (2)$$

如果 $L(i) < 0$ 則表示 c_j 所傳送的封包，其存放位置落在 MH buffer 所容納範圍之外，則此封包將被丟棄。

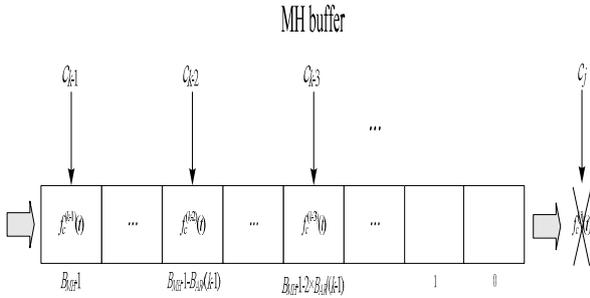


圖 2-3 初始狀態下 MH 接收封包示意圖

MH 換手後將接收新 AR 所傳來的封包，我們假設在時間 t 時，位於 MH buffer 起始存放位置(位置 0)的 packet id 表示為 $f_{MH}(t)$ ，則 MH 所能持有封包之最大 packet id 為 $f_{MH}(t) + B_{MH} - 1$ ，當 c_i 所傳來的封包，其 packet id 是落在 $f_{MH}(t)$ 以及 $f_{MH}(t) + B_{MH} - 1$ 之間時，也就是 $f_{MH}(t) \leq f_c^{(i)}(t) \leq f_{MH}(t) + B_{MH} - 1$ ，此封包將存放在位置 $f_c^{(i)}(t) - f_{MH}(t)$ 。假設 $f_{MH}(t) = k$ ，換手後 MH buffer 存放資料的方式如圖 2-4 所示，MH 接收落在 MH buffer 容納範圍以內的 c_i ，並將其封包存放到位置 $f_c^{(i)}(t) - k$ ，反之落在 MH buffer 容納範圍以外的 $c_j(f_c^{(j)}(t) < k)$ 以及 $c_p(f_c^{(p)}(t) > k + B_{MH} - 1)$ ，其封包將被丟棄。

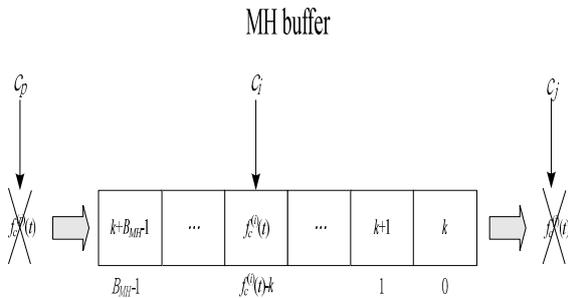


圖 2-4 換手後 MH 接收封包示意圖

當 MH buffer 同時接收多道 multicast channels 的封包時，將能提供錯誤修補的功能，並且在 AR 換手後，MH buffer 的資料可消弭 AR synch error 所帶來的影響。以下我們將舉一個實例來說明 MTMC 下之 error 修補。

首先我們假設 AR 及 MH 皆配置了 5 個存放區塊，如圖 2-5 所示，在時間 T 時，AR₁ 存有 5 個 packet 為(8,9,10,11,12)，此外 AR₁ 配置 3 道 multicast channels 並以前述時差的方式來傳送。而 MH 則同時傾聽所有 multicast channels 上的封包，並將 packet id 落於 MH buffer 所能容納範圍內的封包儲存至 buffer 中，從圖中顯示 MH 同時接收 packet id 分別為 8、10 及 12 的資料，並且目前播放器播放至編號為 8 的 packet。

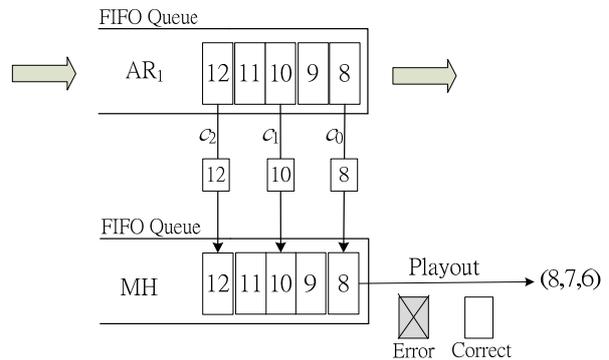


圖 2-5 MTMC 在 T 時之正常情況示意圖

在 $T+1$ 時，MH buffer 存放的 packet 更新為(9,10,11,12,13)。在 $T+2$ 時，如圖 3-6 所示，因無線訊號受到干擾而造成 packet loss(10,12,14)，因此 MH 則無法收到這些資料。然而編號為 10 及 12 的 packet，由於在其他的 channel 上已正確傳送過，因此真正造成 packet loss 只有編號為 14 的 packet，因此 MH buffer 存放的 packet 為(10,11,12,13,?)。

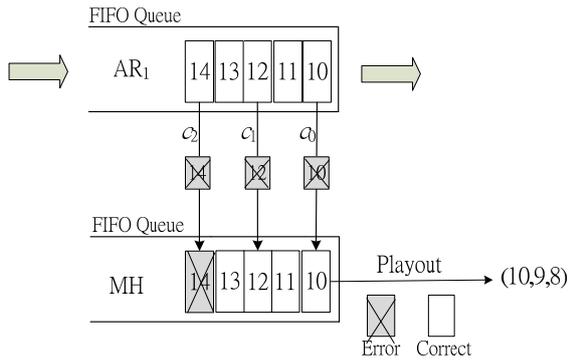


圖 3-6 MTMC 在 $T+2$ 時之錯誤情況示意圖

在 $T+3$ 時，MH buffer 存放的 packet 更新為(11,12,13,?,15)。在 $T+4$ 時，如圖 2-7 所示，假設此時無線通訊處於正常的狀態，編號為 14 的 packet 由 c_1 所傳送的封包來填補，達到 error 的修復能力。

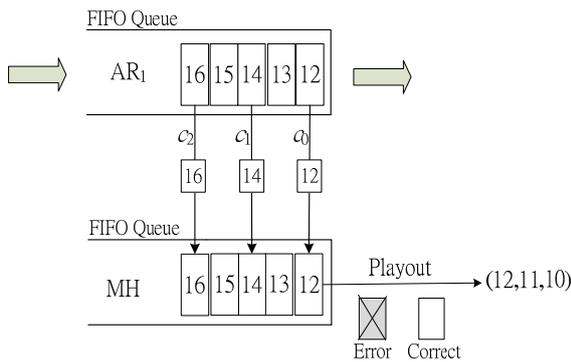


圖 2-7 MTMC 在 $T+4$ 時之錯誤修補機制示意圖

而後 MH 經歷了視訊換手，假設在時間 T 時，換手前 MH buffer 的狀態為 (12,13,14,15,16)，此時進行換手的動作，我們假設有 2 個時間單位的 handoff 中斷，因此在 $T+1$ 時，MH buffer 存放的 packet 為 (13,14,15,16,?) 以及在 $T+2$ 時，MH buffer 存放的 packet 為 (14,15,16,?,?)，在 $T+3$ 時，MH 換手後將接收 AR₂ 所傳送的資料，如圖 2-8 所示，假設此時 MH 同時接收落在 MH buffer 所能容納範圍內的 c_0 (編號為 17 的 packet) 以及 c_1 (編號為 19 的 packet)，而落在

MH 所容納範圍外的 c_2 (編號為 21 的 packet)，則會直接丟棄，因此目前欠缺的 packet 只剩編號為 18 的 packet，MH buffer 存放的 packet 為 (15,16,17,?,19)。

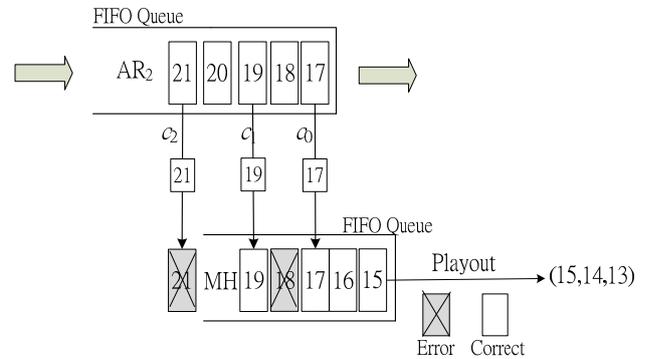


圖 2-8 MTMC 在 $T+3$ 時之換手後封包銜接示意圖

在 $T+4$ 時如圖 2-9 所示，編號為 18 的 packet 已能從 c_0 所傳送的封包來填補，因此 MH buffer 存放的 packet 為 (16,17,18,19,20)，而在整個視訊換手的過程中，stream 的播放沒有中斷，我們稱此狀態為 seamless handoff。

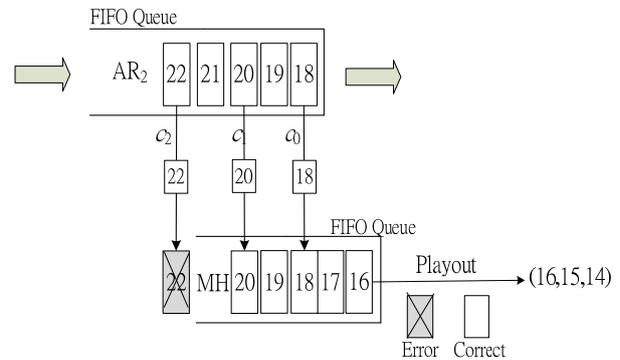


圖 2-9 MTMC 在 $T+4$ 時之換手後資料填補示意圖

2.4 AR Synch Error 處理

本節我們將探討 MH 換手後，沒有任何 channel 落在 MH buffer 所能容納範圍內的特殊情況，此情況將造成視訊播放的 lag 或 skip，lag 情況代表換手後所要銜接的

channel 落於 MH buffer 所能容納的範圍之前，因此 MH 在消耗完 buffer 所預存的資料後，會產生播放暫停的情況並等待是否有 channel 所傳來的 packet 趕上 MH 的播放進度後，才會開始接收資料，同時，播放點才會繼續前進。skip 情況代表換手後所要銜接的 channel 落於 MH buffer 所能容納的範圍之後，但在 MH buffer 的控制下，該 channel 所傳送的封包會在 buffer 消耗完之前，銜接在最後一筆資料之後，因此當播放點已播放至先前所預存的最後一筆 packet 時，之後的播放則會產生視訊跳格的情況。而這兩種情況都將嚴重影響視訊品質，因此我們將選擇一個影響最小的 channel 來銜接，假設 $f_{MH}(t) = m$ ，並且 MH buffer 最後一筆資料的存放位置表示為 n ，如圖 2-10 所示， c_j 與 c_p 分別落在 MH buffer 所能容納的範圍之前以及之後，如果 $m - f_c^{(j)}(t) > f_c^{(p)}(t) - (m+n)$ 則 c_j 將被選擇來銜接，反之如果 $f_c^{(p)}(t) - (m+n) > m - f_c^{(j)}(t)$ 則 c_p 將被選擇來銜接，以下我們進一步舉例說明這兩種情況的效果。

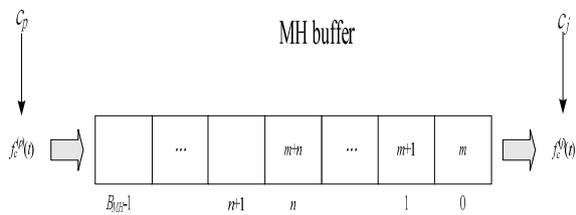


圖 2-10 特殊情況下之 MH 選擇 channel 示意圖

Lag 情況：假設目前有 2 道 multicast channels 所傳送的 packet 落於 MH buffer 所能容納的範圍內，如圖 2-11 所示，而落在 MH buffer 之外的 channel 所傳來的 packet 將直接丟棄，且播放器目前播放至編號為 6 的 packet。

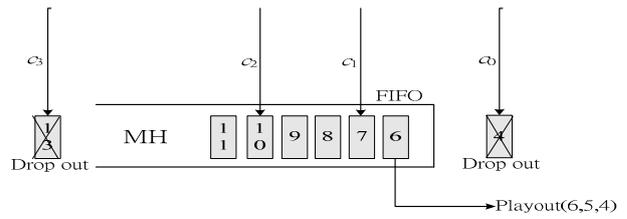


圖 2-11 MH 換手前 buffer 接收資料示意圖 (Lag case)

此時 MH 發生換手的動作，如圖 2-12 所示，換手時消耗了 3 個時間單位，目前播放至編號為 9 的 packet，此時所選擇的 channel c_j 其傳來的 packet 落於 MH buffer 所能容納的範圍之前，因此該 packet 會先被丟棄。

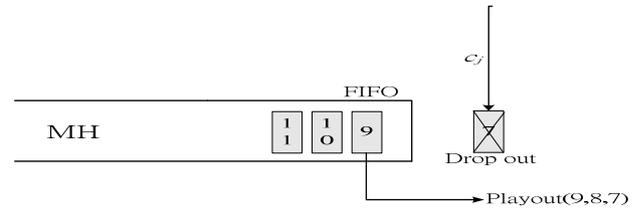


圖 2-12 MH 換手後狀態示意圖 (Lag case)

經過 3 個時間單位後，如圖 2-13 所示，此時 MH buffer 處於 empty 的情況，並且會等待 c_j 所傳來的 packet 落於 buffer 之內，而此時播放器處於暫停的狀態，也就是 $f_{MH}(t)$ 停在 12。

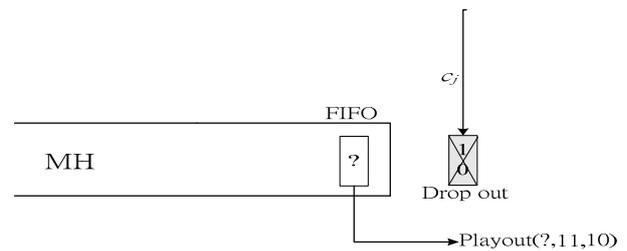


圖 2-13 MH 播放暫停示意圖 (Lag case)

而當 c_j 傳送編號為 12 的 packet 時，如圖 2-14 所示，此時 c_j 已落入 MH buffer 的存放區塊內，其傳來的 packet 才會儲存至 MH buffer 中並將其送至播放器，此時播放

點才會繼續前進。從這個例子顯示，MH 換手的過程造成播放器 2 個時間單位的 lag 情況。

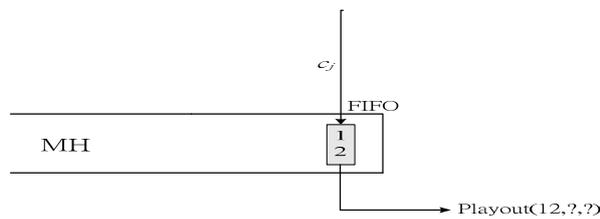


圖 2-14 MH 資料銜接示意圖(Lag case)

Skip 情況：假設目前有 2 道 multicast channels 所傳送的 packet 落於 MH buffer 所容納的範圍內，如圖 2-11 所示，且播放器目前播放至編號為 6 的 stream，此時 MH 發生換手的動作，如圖 2-15 所示，換手時消耗了 3 個時間單位，目前播放至編號為 9 的 stream，此時所選擇的 channel c_p 落於 MH buffer 所能容納的範圍之後，因此 c_p 所傳來的封包會被存放於 MH buffer 之最後一筆資料之後。

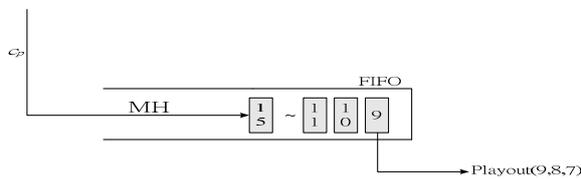


圖 2-15 MH 換手後狀態示意圖(Skip case)

經過 3 個時間單位後，如圖 2-16 所示，packet 的播放由編號 11 跳至編號 15，造成 3 個 packet 的 skip。

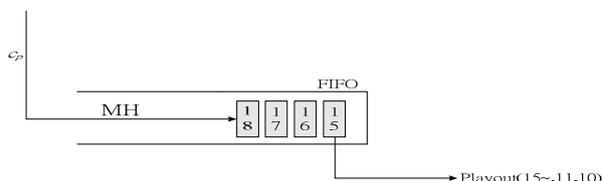


圖 2-16 MH 播放跳格示意圖(Skip case)

由以上的討論 MTMC 機制具有以下優點(1)透過 multiple channels 並結合時間差

的傳送機制，能有效將 burst error 分散在不同 channel 中，以提升錯誤修復能力。(2)換手後，透過 MH 預存於 buffer 的資料能彌補 handoff loss。(3)multiple channels 能使 MH 有更多的機會找到合適的 stream 來銜接，藉此減輕 AR synch error 所帶來的負面影響，以達 seamless handoff 的目的。下一章我們將用模擬的方式，探討 AR 及 MH 之 buffer 量與 channel 數之關係。

三、 模擬與分析

根據上一章節的探討，本章將以先前所提之 Multiple - Time - shifted - Multicast - Channel (MTMC)機制作一模擬與分析，並針對 single multicast channel 以及 multiple time-shifted channels 作一詳細比較，以了解 MTMC 機制的效能。

3.1 環境概述與模擬假設

在模擬的環境上，我們假設在一個無線區域網路，AR 接收來自 video server 的 stream，並以 multicast 的方式傳送 stream 給 MH。在本實驗中，配置了 10 個 AR($AR_0 \sim AR_9$)，並且彼此間有 AR synch error 的差異，除了 AR_0 外，其餘 AR 相對於 AR_0 的 AR synch error 值隨機分佈在 $\pm 1500ms$ 之間，MH 一開始將傾聽 AR_0 以多個 channel 所傳來的 stream，並且接收落在 MH buffer 所能容納範圍內的 stream，在本實驗中 video server 是以 56kbps 的速率來傳送資料，代表每秒傳送 7 個封包，每個封包大小為 1KB，MH 等待 buffer 內的資料存滿之後才開始消耗 stream，MH 會遭遇 9 次的換手動作，依序換手至 AR_1, AR_2, \dots, AR_9 ，而在接近換手的區域時，會有 transmission error 的情況發生，愈接近換手的時間點時，則產生 error 的機會愈大，而當換手後，

遠離換手的時間點時，其 error 發生率正好與換手前的情況相反，愈遠離換手的時間點，error 發生機率愈低，等待 9 次換手的動作完成後，整個模擬才算結束，模擬環境中所使用的參數與對應值如表 4-1。

本實驗將觀察在群播服務中，channel 數與 AR 及 MH buffer 量應配置多少，才能在受訊號干擾的環境中，修補大部分的 error 情況，並且在視訊換手的過程中，達到無縫換手的目的。

表 3-1 模擬環境參數

| 參數 | 單位 | 數值 |
|-----------------------|------|---------------------|
| AR 數量 | 座 | 10 |
| AR synch error 範圍 | ms | ±1500 |
| AR buffer(B_{AR}) | ms | 1000、2000、3000、4000 |
| MH buffer(B_{MH}) | ms | 1000、2000、3000、4000 |
| 換手中斷的時間 | ms | 50 |
| 換手影響範圍 (Error-Field) | ms | 500、1000、1500、2000 |
| Channel 數量 | 個 | 1~5 |
| AR 傳輸速率 | kbps | 56 |

3.2 實驗結果

在無線網路環境下，封包傳輸的過程中，時常造成封包的遺失或是延遲到達的情況，而造成視訊播放上的跳格(我們稱之為 skip)，或是停頓的情形(我們稱之為 lag)，因而嚴重影響其視訊品質，在此我們定義 hiccup H 代表 skip 的封包數加上 lag 的封包數。在本研究中，我們將透過 2 個實驗來探

討在 channel 由 1 變動到 5 的模擬環境中，以 AR 及 MH buffer 量的變動來分析所造成的 hiccup 影響。實驗 1 將比較在 error-free channel 的環境下，比較各 channel 數中所產生的 hiccup 量，並分析其現象。實驗 2 中將比較在 error-prone channel 的環境下，比較各 channel 數中所產生的 hiccup 量，並分析其現象。最後透過調整 error-prone channel 的影響範圍，並比較在各 channel 數中的平均錯誤情況以及錯誤改善率。

3.2.1 實驗一：Error-free channel

在 error-free channel 的模擬環境下，MH 的換手過程中，channel 在傳輸時不會發生 error 的情況，因此換手所帶來的影響只有 AR synch error 以及 handoff loss，並且透過 channel 數由 1 變動到 5 來實驗觀察 AR buffer 以及 MH buffer 的需求與 hiccup 之間的關係。

Channel 數為 1 的 hiccup 分佈情形，如圖 3-1 所示，當 MH buffer 增大時，hiccup 的情況即稍有改善。而 AR buffer 的變化在 channel 數為 1 時，對於 hiccup 改善情況較不明顯。一般而言，channel 數為 1 時，channel 落於 MH buffer 所容納範圍內的機會相對較小，因此在換手的過程受到 AR synch error 以及 handoff loss 影響明顯，將有嚴重的 hiccup。

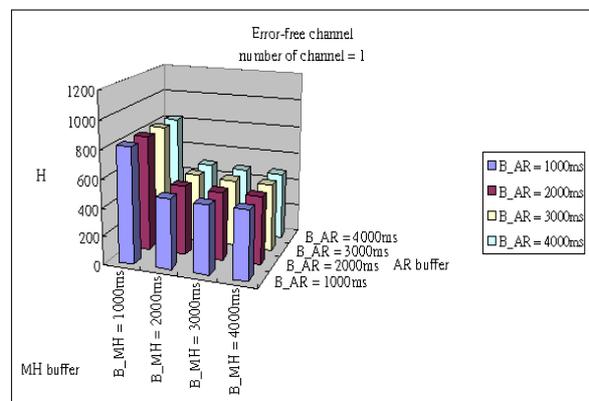


圖 3-1 Error-free channel 下 channel 數為 1 之 hiccup H 分佈圖

Channel 數為 2 之 hiccup 分佈情形，如圖 3-2 所示，MH 在換手後，將至少有 1 道 channel 同時落在 MH buffer 所能容納的範圍內，因此 hiccup 有明顯的改善。但 AR buffer 配置愈大時，(如 $B_{MH} = 1000\text{ms}$, $B_{AR} = 4000\text{ms}$)，由於 channel 間傳送時程被拉大，在換手後 channel 落於 MH buffer 所能容納範圍內的機會反而減少，因此就呈現出 hiccup 較高的情形。而當 $B_{MH} > B_{AR}$ 時，(如 $B_{MH} = 3000\text{ms}$, $B_{AR} = 2000\text{ms}$)代表將有更多的機會讓 2 個 channels 所傳送的 packet 同時能落於 MH buffer 的容納範圍內，因此在 MH 換手後，buffer 內能積存更多的資料，因而能大為降低 AR synch error 所帶來的影響。

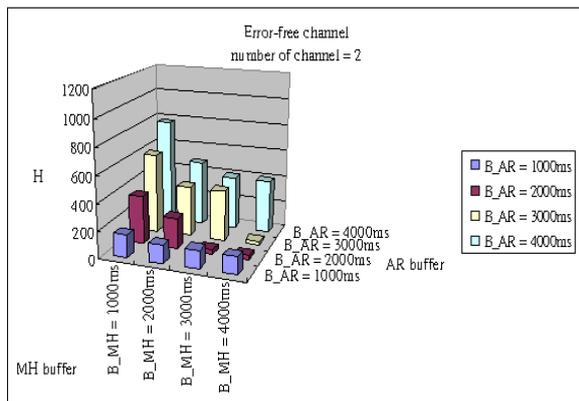


圖 3-2 Error-free channel 下 channel 數為 2 之 hiccup H 分佈圖

Channel 數為 3 之 hiccup 分佈情形，如圖 3-3 所示，由於 channel 數增多，channels 間的傳送時程縮小，channel 能有更佳的機會落於 MH buffer 所能容納範圍內，因此 buffer 有更多的機會積存的資料量來消弭 AR synch error 所造成的影響，hiccup 會隨之降低。然而在 AR buffer 容量增大的情況下，如同前面 channel 數為 2 的情形，仍會出現嚴重的 hiccup。一般而言， $B_{MH} \geq 3000\text{ms}$, $B_{AR} \geq 2000\text{ms}$ 的情況下，能改善大部分 hiccup 所帶來的影響，達到幾近 hiccup-free 的效果。

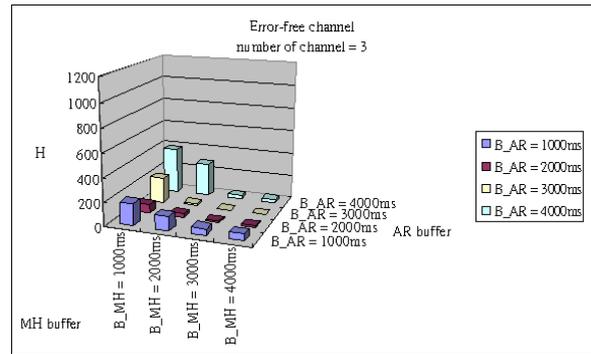


圖 3-3 Error-free channel 下 channel 數為 3 之 hiccup H 分佈圖

圖 3-4、3-5 分別顯示 channel 數為 4、5 的 hiccup 分佈情形，channel 數為 4 時，已能大量的改善 AR synch error 所造成的影響，但對於 channel 數為 3 的 hiccup 情形，其改善程度已不太明顯，因此在足夠的 MH buffer 配置下，已能幾乎達到 seamless handoff 的目標。一般而言，在 channel 數為 4 以上時， $B_{MH} \geq 2000\text{ms}$, $B_{AR} \geq 1000\text{ms}$ 將能達到幾近 hiccup-free 的效果。

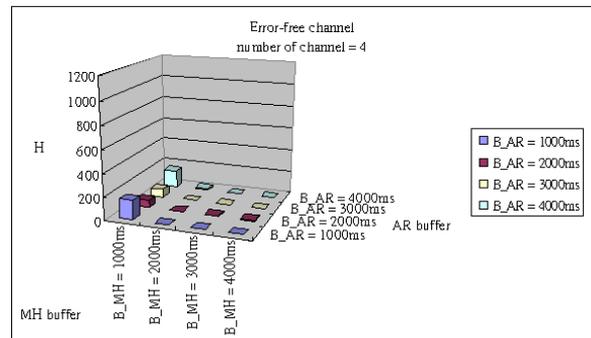


圖 4-4 Error-free channel 下 channel 數為 4 之 hiccup H 分佈圖

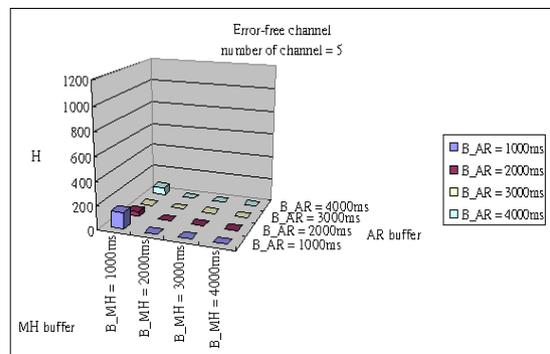


圖 3-5 Error-free channel 下 channel 數為 5 之 hiccup H 分佈圖

從實驗一結果得知，當只有一道 channel 時，MH 換手前無法積存 data，因此 AR synch error 以及 handoff loss 容易造成嚴重 hiccup。而使用 MTMC 後，MH 能從多個 channels 積存資料量，因此可弭平因 AR synch error 以及 handoff loss 所產生的 hiccup。本實驗中，配置 3 道 channels 以及約 2 秒的 AR 與 MH buffer 量，即能在 error-free channel 的環境下，達到幾乎 hiccup-free 的換手效果。

4.2.2 實驗二：Error-prone channel

在 error-prone channel 的模擬環境下，MH 的換手過程中，除了遭遇到 AR synch error 及 handoff loss 外，還包含接近換手時所遭受到的訊號干擾，因此就會產生 transmission error 的情況。而在本實驗中，我們假設 transmission error 發生的機率呈以下指數分佈，其 channel error model 如圖 4-6 所示，transmission error 發生在一個時區內，表示為 transmission error zone，而 MH 進入 transmission error zone 到離開 transmission error zone 所耗費的時間(ms)為 F ，而 MH 所在的位置到離開 transmission error zone 所耗費的時間(ms)為 X ，因此 MH 移動到 transmission error zone 時，其 error rate E 的計算公式為 $E = (X/F)^2$ 。當 MH 離開 transmission error zone 而進入到 handoff zone 時，代表已到了換手的程序，因此將造成 handoff loss。當 MH 離開 handoff zone 而進入到新 AR 的 transmission error zone 時，則使用相反的 error model($1/E$)。以下我們將透過這個 channel error model 來探討在 error-prone channel 的環境下，hiccup 的分佈情形。

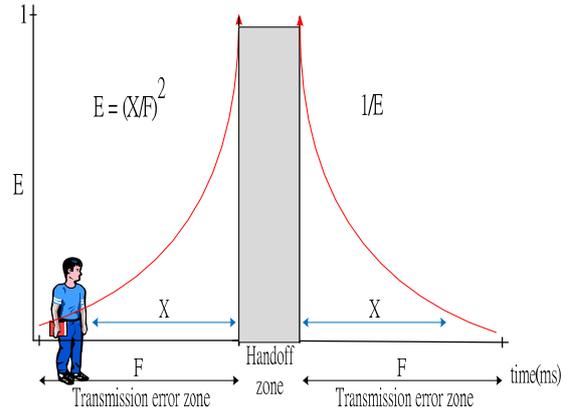


圖 3-6 Channel error model 示意圖

Channel 數為 1 之 hiccup 分佈情形，如圖 3-7 所示，當考慮因素包含 transmission error 的影響時，因考量到 channel 傳送時造成 error 的情況，因此 hiccup 數明顯比 error-free channel(如圖 3-1)還高出 200~250 個封包數。當 MH buffer 配置愈大時，其能改善的 hiccup 數愈多。而 AR buffer 的變化在 channel 數為 1 時，其改善情況較不明顯。一般而言，channel 數為 1 時將受到 AR synch error、handoff loss 以及 transmission error 的影響，造成相當嚴重的 hiccup。

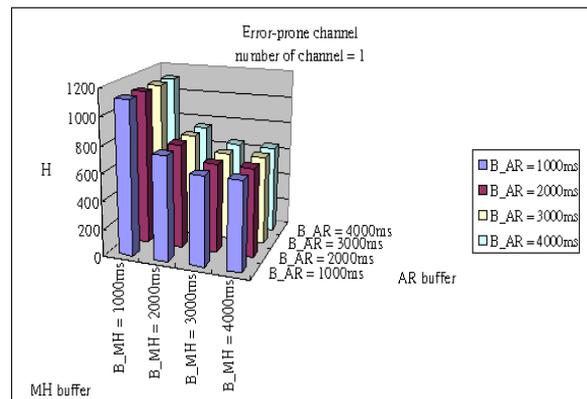


圖 3-7 Error-prone channel 下 channel 數為 1 之 hiccup H 分佈圖

Channel 數為 2 之 hiccup 分佈情形，如圖 3-8 所示，MH 在換手後，將至少有 1 道 channel 落在 MH buffer 所能容納的範圍

內，因此 hiccup 有明顯的改善。當 AR buffer 配置愈大時，(如 $B_{MH} = 1000\text{ms}$, $B_{AR} = 4000\text{ms}$)，代表 channel 間傳送時程被拉大，在換手後 channel 落於 MH buffer 所能容納範圍內的機會反而減少，因此就呈現出 hiccup 較高的情形。然而當 $B_{MH} > B_{AR}$ 時，(如 $B_{MH} = 3000\text{ms}$, $B_{AR} = 2000\text{ms}$)，由於有很大的機會使所有 channels 所傳送的 packet 皆能位於 MH 之 buffer 範圍內，因此 MH buffer 不但能積存更多的 packet，而且還具有 error recovery 的功用，因此 hiccup 改善的程度更大。

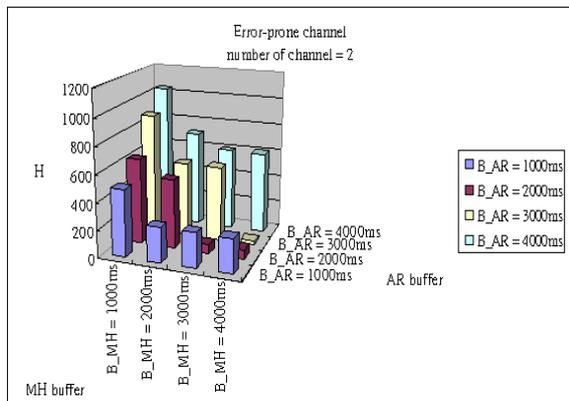


圖 3-8 Error-prone channel 下 channel 數為 2 之 hiccup H 分佈圖

Channel 數為 3 之 hiccup 分佈情形，如圖 3-9 所示，由於 channel 數增多，channels 間的傳送時程縮小，channel 能有較多的機會落於 MH buffer 所能容納範圍內，因此 buffer 能積存更多的資料量來消弭 AR synch error 所造成的影響。當有較多的 channels 所傳送的 packet 落於 MH buffer 內時，其 error recovery 的機會就增多，因而更能改善 transmission error 所造成的影響。在 $B_{AR} > B_{MH}$ 的情況，其 channel 的傳送時程將被拉開，若拉開的距離已超過 MH buffer 的存放範圍時，將使更多的 channel 落在 MH buffer 所容納範圍之外，因此將會降低 hiccup 的改善率。一般而言， $B_{MH} \geq 3000\text{ms}$ ，

$B_{AR} \geq 2000\text{ms}$ 的情況下，將能改善大部分 hiccup 所帶來的影響，達到幾近 hiccup-free 的效果。

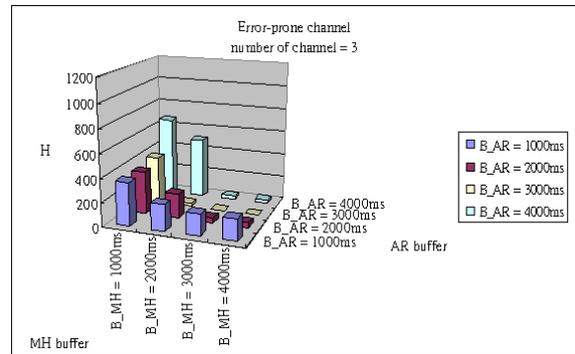


圖 3-9 Error-prone channel 下 channel 數為 3 之 hiccup H 分佈圖

圖 3-10、3-11 分別顯示 channel 數為 4、5 的 hiccup 分佈情形，channel 數為 4 時，已能大量的改善 AR synch error 所造成的影響，並且也有更多的機會讓大多數的 channels 所傳送的 packet 落於 MH buffer 內，因此能有更多的 error recovery 機會，減少 transmission error 所帶來的影響。在 $B_{AR} > B_{MH}$ 的情況下，較多的 channels 能使 channels 間的傳送時程縮小，因而有更多的機會讓大多數 channels 所傳送的 stream 落於 MH buffer 的存放範圍內，達到 hiccup 改善的效果。一般而言，在 channel 數為 4 以上時， $B_{MH} \geq 2000\text{ms}$ ， $B_{AR} \geq 2000\text{ms}$ 將能達到幾近 hiccup-free 的效果。

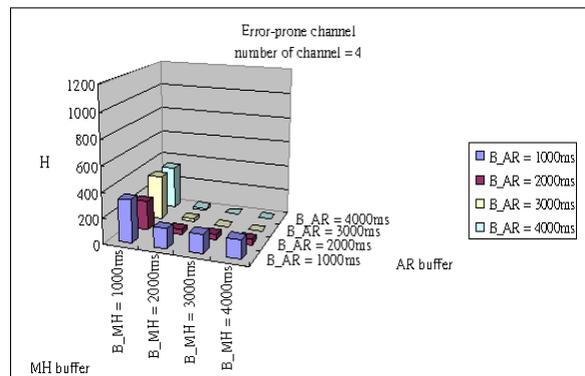


圖 3-10 Error-prone channel 下 channel 數為 4 之 hiccup H 分佈圖

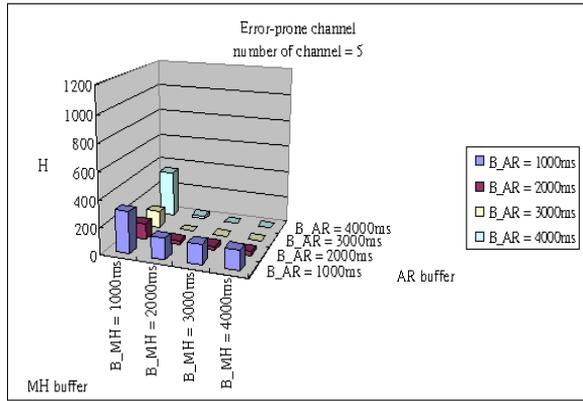


圖 3-11 Error-prone channel 下 channel 數為 5 之 hiccup H 分佈圖

從實驗二結果得知，當只有一道 channel 時，MH 換手前無法積存 data 且無法修復 transmission error，因此容易造成嚴重 hiccup。而使用 MTMC 後，MH 能從多個 channels 積存資料量且修復 transmission error，因此可大大提升資料正確性，弭平 hiccup。一般而言配置 3 道 channels 及 2~3 秒的 AR 與 MH buffer 量，即能在一般 WLAN 環境下，達到幾乎 hiccup-free 的換手效果。

3.2.3 平均錯誤分佈及錯誤改善率分析

在本章節中，我們將調整 transmission error 的影響範圍並觀察各平均 hiccup 分佈以及 hiccup 的改善率。我們定義平均 hiccup 數之其表示為 H^{avg} ，也就是根據本實驗中，AR 及 MH buffer 配置的 16 種組合變化所得的 H 結果來平均，從圖 3-12 的結果得知，當 MH 受 transmission error zone 的影響範圍愈大時，若沒有使用多個 channels 傳送時，其 hiccup 的情況將相當嚴重。而經由 MTMC 機制後，其受到 hiccup 的影響則有顯著改善。我們進而觀察 hiccup 的改善程度，假設 channel 數為 i 的平均 hiccup 數表示為 H^{avg}_i ，則 hiccup 的改善率 Imp 可表示為

$$Imp = (H^{avg}_1 - H^{avg}_i) / H^{avg}_1 \quad (3)$$

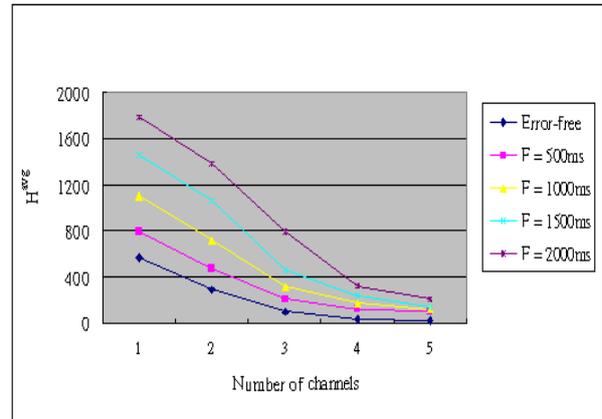


圖 3-12 平均 hiccup H^{avg} 分佈圖

圖 3-13 顯示，從 hiccup 改善率的結果得知，transmission error zone 的範圍愈大時，對於 single channel 來說，造成嚴重 hiccup。但對於 MTMC 來說，即便是大範圍的 transmission error zone (2sec)，當 channel 數為 4 時，就可修補 80% 以上的 hiccup。如果是一般的 transmission error zone 範圍 (<1.5sec)，channel 數為 3 時，就足夠修補 65% 以上的 hiccup，而在 channel 數為 4 時更能修補至 85% 以上的 hiccup。因此透過 MTMC 機制來調整 channel 數時，將能滿足在不良的通訊環境之下，保持其應有的播放品質。

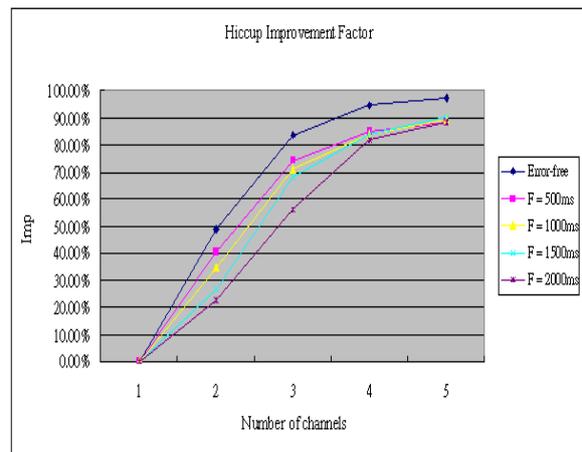


圖 3-13 Hiccup 改善率分佈圖

四、 結論

在行動網路環境下，視訊傳輸常受到訊號不良的影響，而在群播網路的傳輸模式下，視訊換手的程序將造成通訊中斷的情況，影響其播放品質，主要是因為傳輸率是由 video server 來控制，而 client 端只能採取被動接收，並且在 video server 傳送串流的過程也會因所走的路徑不同、受到延遲程度的不同而產生 AR synch error 的問題，因此在視訊換手完成後，即會產生銜接上的差異，以及造成視訊播放上的延遲及跳格而影響了播放上的品質。本研究針對無線群播網路的環境，提出 Multiple - Time - shifted - Multicast - Channel 的機制，它能增加 MH buffer 的積存量，以改善 AR synch error 以及 handoff loss 所帶來的影響。同時透過本機制，我們可以分散 burst error 所造成的影響，使 MH 能由其餘的 channel 取得遺失的封包，更能有效提升視訊品質。從實驗結果顯示 channel 數為 3、AR buffer 配置量為 2~3 秒、MH buffer 配置量為 2~3 秒，即能在一般的 transmission error zone 範圍 (<1.5sec)，達到 seamless stream handoff，提供幾近 hiccup-free 的視訊播放品質。透過 MTMC 的機制，我們可以發現，當 burst error 影響範圍不大時，我們可以減少 channel 數，以節省網路頻寬的消耗。但在 burst error 影響範圍較大時，雖然使用多個 channel 數會增加頻寬的消耗，但 multicast 所服務的對象為大量的使用者，因此對於總體的頻寬耗用成本是相當符合經濟效益。因此 MTMC 的機制非常適合運用在 multicast 的網路服務，以改善視訊播放品質。

五、 參考文獻

- [1] 楊勝雄，行動式環境中視訊換手之探討，碩士論文，屏東科技大學資訊管理系，民國九十三年七月。
- [2] Banerjee, N., Wei, W., Das, S.K.,

- “Mobility Support in Wireless Internet,” IEEE Wireless Communications, pp. 54-61., Oct. 2003.
- [3] Dutta, A., Fajardo, V., Famolari, D., Kodama, T., Ohba, Y., Schulzrinne, H., Taniuchi, K. and Van den berg, E., “Dynamic Buffering Control Scheme for Mobile Handoff”, Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on pp.1- 11., Sept. 2006.
- [4] Guo, M. and Ammar, M.H., “Scalable live video streaming to cooperative clients using time shifting and video patching”, INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, Vol.3, pp.1501-1511., 2004.
- [5] Johnson, D., Perkins, C. and Arkko, J., “Mobility Support in IPv6,” Internet Draft, Jan. 2003.
- [6] Lai, J. and Liao, W., “Mobile Multicast with Routing Optimization for Recipient Mobility”, IEEE Transactions on Consumer Electronics, Vol.47, No.1, pp.199-206., Feb. 2001.
- [7] Lee, J.Y.B., “Channel folding - an algorithm to improve efficiency of multicast video-on-demand systems”, IEEE Transactions on Multimedia, Vol.7, No.2, pp.366-378., Apr. 2005.
- [8] Lee, S., Kim, K. and Ahmad, A., “Channel Error and Handoff Compensation Scheme for Fair Queueing Algorithms in Wireless Networks ”, IEEE International Conference on Communications, Vol.5, pp.3128-3132., May 2002.
- [9] Lin, Y. and Chlamtac, I., “Wireless and Mobile Network Architectures”, Wiley Computer Publishing, 2001.
- [10] Loyka, L., “Channel Capacity of MIMO Architecture Using the Exponential Correlation Matrix,” IEEE communications letter (2.2.3.1/MIMO),

Vol.5, No.9, Sep. 2001.

- [11] Madani, M. and Light, J., "Multi-level priority queue scheduling algorithm for critical packet loss elimination during handoff", Communication Networks and Services Research Conference, Proceedings of the 3rd Annual 16-18 in 2005, pp.275-281., May 2005.
- [12] Pahlavan, K. and Krishnamurthy, P., "Principles of Wireless Networks," Prentice Hall Upper Saddle River, 2002.
- [13] Perkins C., "IP Mobility Support", RFC 2002, Network Working Group, October 1996.
- [14] Sharma, S., Zhu, N. and Chiueh, T., "Low-Latency Mobile IP Handoff for Infrastructure-Mode Wireless LANs," IEEE JSAC, Vol.22, No.4, pp. 643-652., May 2004.
- [15] Tong, S. and Yang, S., "Buffer Control for Supporting Seamless Stream Handoff in WLAN Employing Bi-cast Streaming in FMIPv6," IEEE Transactions on Wireless Communications, under revision, Dec. 2006.
- [16] Wu, I., Cen, W., Lial, H. and Young, F., "A seamless handoff approach of mobile IP protocol for mobile wireless data networks", IEEE Transactions on Consumer Electronics ,May 2002.