

應用於行動 P2P 串流網路之統計式多來源資料下載策略

童曉儒 林思宏

國立屏東科技大學 資訊管理系

srtong@mail.npust.edu.tw、M9956003@mail.npust.edu.tw

摘要

在行動同儕網路(Mobile peer-to-peer)環境中提供視訊串流服務面臨著許多傳輸上的挑戰，例如：高封包遺失率、網路連線不穩定與使用者網路的異質性等問題。因此本論文針對這些議題，提出了兩種創新傳輸策略：「統計式下載(Statistical Download；SD)」與「無區塊式交錯(Blockless Interleaving；BI)」，藉以配合 UEP-FEC 通道編碼，提供調適性的視訊品質。SD 是基於一個 RTT 機率模型，利用多來源的下載方式，提升封包下載成功率；而 BI 能根據來源端的網路遺失情形，提供不同的 interleaving 傳輸策略。模擬實驗結果發現，即使系統在高度異動與異質性的網路下，SD 仍然可以提供穩定且即時的資料速率，同時 BI 亦能有效提升 UEP-FEC 解碼效能，降低封包叢集遺失的影響。而這些措施能明顯提升串流影片播放的連續性，減少中斷發生機率。

關鍵詞：點對點網路傳輸、前向錯誤更正、不等量錯誤保護、交錯傳輸、調適性視訊品質、行動視訊串流。

Abstract

Provisioning of video streaming service in Mobile P2P networks is challenged by many facts in transmission, such as high data loss rate, burst loss, unstable connectivity, link and peer host heterogeneity. This thesis addresses these transport issues by proposing two novel schemes, statistical download (SD) and blockless interleaving (BI), which cooperatively work with UEP-FEC channel coding. SD improves the successful rate of data download by exploiting multi-source download according to a RTT-based probability model. BI conducts different interleaving policies of data download from respective source peers according to the associated channel loss process. Simulation results show that even the system presents high volatility and heterogeneity, when the size grows into large, SD can improve on-time data rate with moderate traffic overheads. Meanwhile, BI can improve the error recovery performance of UEP-FEC by eliminating the packet loss correlation. It turns out that the video continuity is significantly improved.

Keywords: Peer-to-peer network, forward error correction, unequal error protection, interleaving, adaptive video quality, mobile video streaming.

1. 前言

在行動 P2P(Mobile P2P；MP2P)網路中觀看串流影片，有兩項待解的議題：(1)首先，觀看串流影片時，如果採用 ARQ 的策略，必須等待 timeout 才能進行封包的重送，這容易導致資料無法即時下載，而產生播放延遲的問題。為了解決此問題，我們可以讓使用者提早下載資料，使重傳不至於失效，但是這又導致 client peer 必須事先備有大量的 buffer 來儲存暫存的資料。(2)因為實際網路傳輸是連續性的遺失情況(loss burst)，容易對 UEP-FEC 錯誤修復效能的影響，傳統上使用的 interleaving 乃是將相同 size 的 FEC codeword 組合起來，進行交錯傳送，以達到均衡錯誤的目標，但是在 MP2P 環境中，封包的來源可能來自於不同的 source peer，而每個 source peer 的網路情況也迥異，因此如何導入 interleaving 概念，以克服 packet loss burst 是一項有趣的議題。

本論文提出了兩項機制：「統計式下載(Statistical Download；SD)」與「無區塊式交錯(Blockless Interleaving；BI)」。在 Statistical Download 策略中，我們提出一個以 RTT 為基礎的機率模型，允許 client peer 向多個 source peer 同時要求某一個封包，藉以提高封包第一次下載的成功率，如此一來可以確保 UEP-FEC 錯誤修復的效能，另一方面亦可降低發生 ARQ 的機率。另外在 Blockless Interleaving 機制中，我們將傳統單一來源 interleaving 的概念，導入多重 source peer 的環境中，也就是每個 source peer 依據它的網路狀況，以不同的 interleaving 方式提供資料給 client peer 中不同的 UEP-FEC codeword，如此一來，各個 channel 中的 packet loss burst 被均勻的打散在各個 UEP-FEC codeword 中，不致過於集中，而影響錯誤修復的效能。

2. 研究方法

本論文所探討的 MP2P 系統架構如圖 1 所示。首先，Media server 將壓縮後的影片進行 UEP-FEC 通道編碼，也就是將數個 Group of Picture (GOP) 視為一個 epoch，每個 epoch 內的資料組成一個 Block of Picture (BOP)，然後編碼成一個 UEP-FEC codeword，接著把 UEP-FEC codeword 資料進行封裝(packetizer)的動作，最後將封包儲存至 cache buffer 當中，等候傳送。在 client peer 的部分，當 client peer 加入系統之後，會先向 directory server 註冊，同時傳送 cache buffer 內的 buffer map，接著 directory server 會將相關 client peer 的 buffer map 進行比對動作，然後傳送一個可供資料下載的 source

peer list 給 client peer，接著 client peer 根據本論文所提出的「Statistical Download」與「Blockless Interleaving」決定封包下載的排程，再交由 Multiple Stop-and-Wait (MSW) 傳輸模組進行資料的下載。當 client peer 接收到 source peer 回傳的封包資料時，會預先將資料暫存至 cache buffer 中，同時可提供給其它 client peer 進行下載，當某個 GOP 播放時間到期，並且它所屬的 UEP-FEC codeword 累積資料量足夠時，會將資料 depacketizer，並傳送至 UEP-FEC decoder 進行解碼，解碼後的 BOP 會重新安排成原來的 GOP 然後傳送給播放器進行影片的播放。

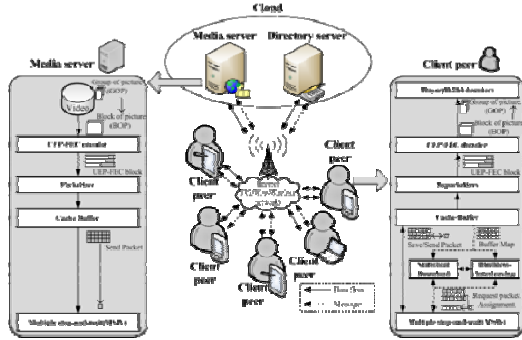


圖 1：MP2P 系統架構圖。

依據上述的系統架構，每個主要的系統模組功能歸納如下：UEP-FEC encoder/decoder 模組：將影片資料進行通道編碼，根據網路的狀況提供適當的冗餘碼，以修復錯誤。Multiple Stop-and-Wait protocol (MSW) 傳輸模組：針對 source peer list 中，各個 source peer 的能力與網路狀況，指派封包下載。Statistical Download (SD) 模組：提出以 RTT 為基礎的封包遺失率模型，並以此模型推導 client peer 應向那些 source peer 要求封包，以確保封包下載成功機率。Blockless Interleaving (BI) 模組：根據各個 source peer 連線的網路 loss burst 狀況，決定該 source peer 應採行的資料下載 interleaving pattern，以降低 loss burst 對 FEC 修復效能的負面影響。

2.1 UEP-FEC 通道編碼[5]

我們假設影片採用 Scalable Video Coding (如：H.264) 壓縮(如圖 2)，依影格(frame)的重要性，分成 q 個 layer，表示成 $L_0, L_1 \dots L_{q-1}$ ，編號越低的 layer 代表重要性越高，每個 Group of Picture (GOP) 由一序列重複的 layer pattern 影格所組成，並可獨立解碼。GOP 的開始往往是影片隨機存取的切入點，而一個 epoch 包含了 C 個 GOP，構成一個 Block of Picture (BOP)，BOP 為 UEP-FEC 通道編碼的基本單位，BOP 的資料會重新安置成 UEP-FEC codeword。如圖 3 2 所示，其中切割成 q 個區塊，編號為 $0, 1 \dots q-1$ ，區塊 i 所放置的是 layer i 所有的 frame，layer i 的每個 column 會產生出一個 $FEC(n, k_i)$ codeword，最後將每行封裝成一個 packet (亦稱為 descriptor)，共 N 個 packets，然後送至 cache buffer 等待傳送。

當 client peer 收到 k_i 個封包時，依據 FEC 的特性，layer 0、layer 1...layer i 所有的資料皆可修復。

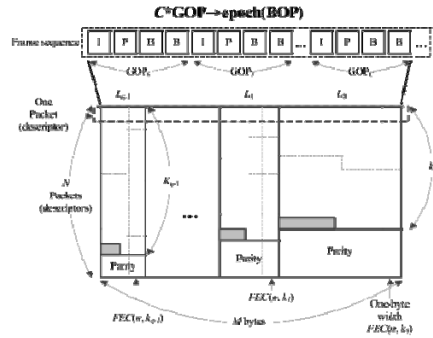


圖 2：UEP-FEC codeword 組成架構。

2.2 Multiple stop-and-wait(MSW)傳輸模組[1]

Client peer 對每個可提供資料的 source peer 建立單獨的連線(圖如 3)，而每個連線是採用 stop-and-wait 的下載模式，也就是當 client peer 發出 request 至 source peer 之後，會等待直到取得回傳的封包後，再發出下一次的 request。如果順利的於 timeout 時間內接收到封包，則會以遞增方式增加下次要求的封包數量；當逾時發生時，代表 source peer 離線或者是網路不穩造成封包遺失，則會將原本委託該 source peer 的任務重新分配至其它 source peer 執行重傳。另外在 source peer 方面，會以 First-Come-First-Serve 的方式回應封包給 client peer。

MSW 傳輸機制可以根據 source peer 的網路狀況自動調整資料下載量，當封包遺失時會自動啟動其它 source peer 的重傳機制，因此可以有效整合 P2P 系統中異質的網路與使用者，提升整體的 throughput。此特性在使用者頻寬有限的行動環境下是非常重要的。

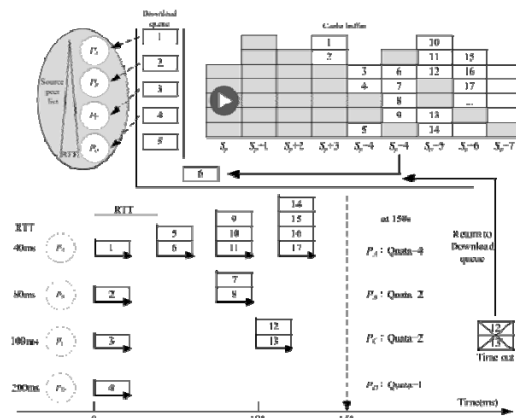


圖 3：MSW 運作圖例[1]。

2.3 統計式下載(Statistical Download)

然而在前述的 MSW 中，往往需要等到 request timeout 之後才會啟動重傳，而此時可能造成該封包

已經超過播放時間。再者 UEP-FEC 雖然提供了不同程度的保護等級，但是當封包遺失情形過於嚴重，超過了 layer 0 可修復能力時，仍會造成影片播放暫停的情形。

有鑑於此，我們打破封包來自單一 source peer 的限制，允許 packet 從一個以上的 source peer 同時下載，以提升資料下載的成功率。基本的原理是，我們制訂了一個目標(可容許)的封包遺失率 PLR_{target} ，根據此遺失率設計 UEP-FEC 保護。每個封包下載分派給一群 source peers 執行，對每個 source peer 而言，封包下載的成功率取決於該 source peer 到 client peer 之間的網路狀況，如：RTT 時間或是封包遺失率。對每個封包而言，這群 source peer 所提供的整合封包遺失率必須能使 UEP-FEC 解碼後的封包遺失率低於 PLR_{target} 。我們稱這種以機率為依據的下載策略為「統計式下載(Statistical Download; SD)」。

我們進一步說明細部設計。基本上在整個傳輸過程中，造成封包遺失的主因有兩種。

(1)逾時傳送：由於封包傳輸過程中，發生 delay 的情況，因此封包收到時已經過了它的播放的時間，而成為無效的封包。在此令 rt_s 代表 client peer 與 source peer s 之間的 Round Trip Time (RTT)， $P_{rt_s}(T_d)$ 則表示 rt_s 小於 timeout T_d 的機率，因此逾時的機率可表示為 $1 - P_{rt_s}(T_d)$ 。

(2)封包遺失：在傳送過程中，由於無線訊號的干擾造成封包遺失，在此我們以 $PLR_{ch}(s)$ 來表示與 source peer s 相關的通道遺失率。

假設封包 e 由一群 source peer S_e 集合下載，則整體 S_e 所產生的 PLR 可表示成：

$$PLR_e(T_d; S_e) = \prod_{s \in S_e} (P_{rt_s}(T_d) PLR_{ch}(s) + (1 - P_{rt_s}(T_d))) \quad (1)$$

當封包 e 的 PLR_e 高於某個容許值 β 時，我們會持續指派新的 source peer 加入 S_e 中，直到 $PLR_e \leq \beta$ ，而這個容許值 β 的訂定與 UEP-FEC 中各 layer 有關。假設 $PLR_{target,i}$ 為 layer i 所允許的封包遺失率，從文獻[6]中，我們得知對 layer i 而言，如果可容許的封包遺失率為 β_i ，則 $FEC(n, k_i)$ 解碼後的有效封包遺失率可表示成：

$$PLR_{eff}^{(n,k_i)}(\beta_i) = \frac{1}{k_i} \left(\sum_{t=1}^{k_i} t \binom{k_i}{t} \beta_i^t (1 - \beta_i)^{k_i-t} \times \sum_{j=\max\{0, n-k_i-t+1\}}^{n-k_i} \binom{n-k_i}{j} \beta_i^j (1 - \beta_i)^{n-k_i-j} \right) \quad (2)$$

因此 β_i 可表示成：

$$\beta_i = PLR_{eff}^{(n,k_i)-1}(PLR_{target,i}) \quad (3)$$

一般而言當 $i < j$ 時， $PLR_{target,i} \leq PLR_{target,j}$ ，因此 $\beta_i < \beta_j$ ，所以我們可以令所有的 β_i 皆參考 β_0 ，也就是：

$$\beta_i = PLR_{eff}^{(n,k_i)-1}(PLR_{target,i}) < \beta_0, 1 \leq i \leq q-1 \quad (4)$$

就某個封包 e 而言，當 $PLR_e > \beta_0$ 時，它會保留在 download list 中，等候指派(狀態為“等待中”)；直到我們加入某個 source peer 後， $PLR_e \leq \beta_0$ 時，我們才會停止繼續指派下載它(狀態為“下載中”)；當封包 e 成功接收時，其狀態會更改為“已接收”。

接下來我們說明如何訂定 $P_{rt_s}(T_d)$ 。根據 Chebyshev 不等式[2]，對任何一個隨機變數 X 而言，下列關係永遠成立：

$$P(|X - \mu| \geq \alpha\sigma) \leq \frac{1}{\alpha^2}, \text{ for any real number } \alpha > 0 \quad (5)$$

此處 μ 為 X 之平均值， σ 為 X 之標準差， α 為正整數，我們假設系統統計至某個 source peer s 之 RTT 的平均值為 μ_s 與標準差為 σ_s ，則當 $T_d = \mu_s + \alpha\sigma_s$ 時， rt_s 之分佈可表示成：

$$P_{rt_s}(\mu_s + \alpha\sigma_s) \geq 1 - \frac{1}{\alpha^2} \quad (6)$$

關於 α 值的大小，當 α 值越大時，timeout 時間越長，超過 timeout 時間的機率越小；反之 α 值越小時，超過 timeout 時間的機率則越大。Jacobson/Karels[3]建議 $\alpha = 3$ ，因此 timeout 時間 T_d 為 $\mu_s + 3\sigma_s$ ，也就是說封包能於 timeout 時間抵達的機率為 $P_{rt_s}(T_d = \mu_s + 3\sigma_s) \geq \frac{8}{9}$ 。在實驗中，我們會分析在不同 α 值對系統效能的影響。

2.4 無區塊式交錯(Blockless Interleaving)

雖然在「Statistical Download」能有效提升傳輸成功率，但是在實際網路中傳輸封包常發生叢集遺失(loss burst)的情況，當封包遺失過於集中在某一個 UEP-FEC codeword 中時，經常會導致遺失無法修復。因此針對這個問題，我們提出一個適用於 multisource 的「Blockless Interleaving」下載策略。

基本的原理就是將來自相同 source peer 的封包盡量打散在不同的 UEP-FEC codeword 當中，使遺失能平均分佈，當 loss burst 現象嚴重時，則打散在較多的 UEP-FEC codeword 當中；反之當 loss burst 較輕微時，則僅需分散在幾個 UEP-FEC codeword 中，即可達到平均分佈遺失的效果。我們稱分散至幾個 UEP-FEC codeword 的數量為 interleaving depth d 。因此每個 source peer 依據它的網路狀況，以最合適的 interleaving depth 提供資料給 client peer。

接下來我們說明如何為每個 source peer s 訂定合適的 interleaving depth d_s 。我們參考文獻[4]中的 IF 演算法，當 source peer s 的 loss model 為 $GE(p_s, q_s)$ 時，則 interleaving depth d_s 為：

$$d_s = \left\lceil \frac{\log(1-0.9)}{\log(1-p_s-q_s)} \right\rceil \quad (7)$$

接著我們討論「Blockless Interleaving(BI)」封包派送的排程。因為 client peer 從發出 request 到接收封包，須經過一段 RTT 時間，因此在下載串流影片資料時，我們必須考慮到發出 request 的封包是否能在 playback time 到期之前成功接收。為了達到這個目的，我們以 e_c 作為基準點，向後推算 RTT 時間後所對應的 epoch，也就是 id 為 $c + \left\lceil \frac{\mu_s + \alpha \sigma_s}{\Delta t} \right\rceil + 1$ 的 epoch，此處 Δt 為一個 epoch 的時間長度，然後對該 epoch 以後“等待中”的封包提出 request，如果這些封包都沒有遺失的話，它們皆在 playback time 前成功接收到。

因此我們從 buffer 空間內，自以下 epoch range (ER) 中，依 id 先後順序，從每個 epoch 中挑選至多一個“等待中”的封包傳送。為了避免 interleaving depth 高於 B ，因此我們取 interleaving depth d_s 與 B 的最小值，做為 ER 的上限：

$$ER = \left[e_{c + \left\lceil \frac{\mu_s + \alpha \sigma_s}{\Delta t} \right\rceil + 1}, \min \left\{ e_{c + \left\lceil \frac{\mu_s + \alpha \sigma_s}{\Delta t} \right\rceil + d_s}, e_{c+B-1} \right\} \right] \quad (8)$$

當 ER 範圍中所有 epoch 皆檢查過，或者挑選的封包數量已達到此次 request 的上限 q 時，我們則會停止該次 request 的排程，此處 q 值為：

$$q = \min \{ q^* + 1, m_c, m_s \} \quad (9)$$

其中 q^* 代表前次 request 所允許的封包下載數量； m_c 則代表 ER 中，“等待中”的封包數量； m_s 為 source peer s 所能提供的最大下載量。 m_s 的決定是為了避免某幾個 client peer 霸佔全部的下載資源，因此我們讓 $m_s = \frac{U}{n}$ ，其中 U 為該 source peer 可提供給外界的下載頻寬， n 為目前正在進行傳輸的 client peer 數量。

3. 實驗環境與結果分析

為了要了解「Statistical Download」與「Blockless Interleaving」所產生的效能影響，我們模擬四種可能的傳輸組合，分別是「Deterministic Download (DD)」與「Statistical Download+Blockless Interleaving (SD+BI)」。這兩種機制的主要差異於 transport 中 DD 是 Sequential(Single source /packet)，而 SD+BI 是 Statistical(Multiple sources/packet)，而 retransmission 的部分，DD 是 timeout 之後啟動，而 SD+BI 是當 Playback due 但是封包資料仍不足時才會啟動，最後 SD 多了 interleaving 的策略。

下面我們欲觀察下列四種效能參數：continuity index (C_{index})、total traffic (T) 與有效封包遺失率 $PLR_{eff,0}$ 在兩種傳輸機制的表現。接著針對影像品質進行探討，比較兩種傳輸機制之 PSNR 與 performance gain。最後針對 SD+BI 傳輸機制，進行不同標準差倍數 α 與 buffer size B 的分析，觀察其效

能表現。

我們實驗網路拓樸主要可分為四個部分。video server 能提供上傳與下載各 1000Mbps 的網路速度，並且其網路封包遺失模型 GE loss model 的 $PLR_{ch}=0.1\sim 0.4$ ， $q=0.2$ ；directory server 主要提供使用者註冊與 buffer map 的交換，因此無設定頻寬上傳與下載上限。在使用者方面：wireless host (WH) 占整體使用者中的 20%，網路速度為 5Mbps/1.5Mbps，GE loss model 的 $PLR_{ch}=0.1\sim 0.4$ ，而 $q=0.15$ ；另外 80% 是 mobile host (MH)，網路速度為 1.5Mbps/0.5Mbps，GE loss model 的 $PLR_{ch}=0.1\sim 0.4$ ，而 $q=0.1$ 。

模擬環境主要使用建構於 OMNeT++ 平台上的 OverSim P2P。模擬環境中總模擬時間為 900 秒鐘，我們為 WH 與 MH 分別建構兩個獨立的 churn generator 模擬兩種不同類別使用者的加入/離開，平均為 120sec，我們調整平均系統大小(同時上線的 peer 總數)與網路遺失率 PLR_{ch} 來觀察系統的行為變化。無線網路環境主要採用 Gilbert-Elliott 模型來模擬 packet loss burst 的情形，網路傳輸的 delay 時間則使用 exponential 分佈來模擬，WH 平均為 5ms，MH 平均為 20ms。我們實驗的影片來自於 Video Trace Library 的一部 NBC News，其新聞影片長度約 90 秒鐘，並挑選適用於行動裝置中觀看的高畫質 mp4 影片，bit rate 約為 253Kbps，影格速率為 30fps。同時以 UEP-FEC 編碼方式，針對 H.264 格式中的 I-P 與 B frame 進行 FEC(256, 192)、FEC(256, 224) 與 FEC(256, 240) 的編碼，因此 client peer 收到 epoch 中 75% (low)、83.5% (medium)、93.75% (high) 的封包比例時，分別可解碼回 I (Layer 0)、I+P (Layer 1) 或 I+P+B frame (Layer 2) 的視訊品質。在此，我們選用 3 個 GOP 為一個 epoch，而一個 epoch 包含了約 1.6 秒的影片資訊，平均大小約為 61 個為 1KB 的封包，而影片因透過 UEP-FEC 的編碼，所以其總資料量增加約 21.5% 的資料量，因此實際產生的 bit rate 約為 307Kbps，

3.1 Continuity index(C_{index})之討論

如圖 4 我們比較了兩種 scheme 的 C_{index} 值。兩種 scheme 中看到一些共通的現象：在 N 小的時候，它們的分享效率不佳，因此 C_{index} 一般看來較差；當 N 增加的時候，分享效果較好，因此 C_{index} 能獲得改善。就網路狀況來講，當網路狀況好的時候，分享較有效率；網路狀況較差時，容易發生封包遺失，因此 C_{index} 會降低。

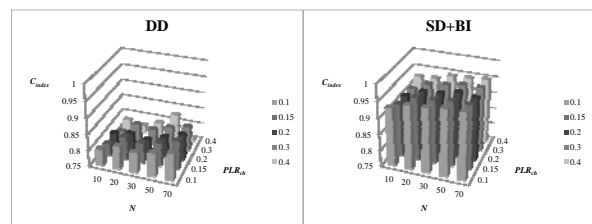


圖 4： C_{index} 之比較：(a)DD、(b)SD+BI。

我們由分析可以得知，在 DD 中無法有效解決網路 loss burst 問題，並且花費長時間等待封包重傳，造成 epoch 資料不足無法進行解碼，因此 C_{index} 較低。而 SD+BI，可以避免封包重傳延遲的問題，並且能夠提升預測的準確性，因此相較於 DD 可以提升 C_{index} 約 14%。

3.2 Total traffic(T)之討論

我們由前面分析知道了 SD 與 BI 能有效提升 C_{index} ，但在效能提升的背後所付出的代價，在此進一步的探討。如圖 5 顯示了兩種 scheme 在不同 N 與 PLR_{ch} 下 total traffic T 的變化。在兩種 scheme 下，當 PLR_{ch} 較高時(網路情況較差)，封包遺失情況嚴重，因此需要重傳大量封包，造成 T 提升。當 PLR_{ch} 較低時(網路情況較佳)，能有效傳遞封包，降低重傳發生機率，因此 T 值較低；當 N 較大的環境中，因為系統成員增加，client peer 下載人數較多，需分享資料給較多的 client peer，因此相較於 N 較小的環境， T 值會些許提升。

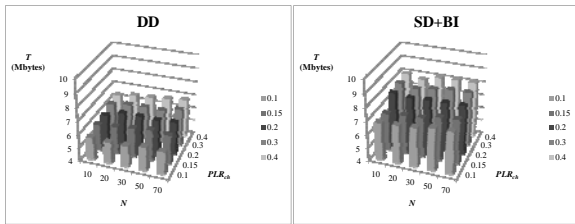


圖 5：Total traffic 之比較：(a)DD、(b)SD+BI。

由上面實驗我們可以得知，除了在 PLR_{ch} 的環境下， T 會因為封包遺失有所提升之外，DD 因為會 loss burst 的影響，增加了因 lag 進行重傳的機率，但針對一個封包只會跟一個 source peer 進行下載，因此能有效控制 total traffic。而 SD+BI，因 BI 能有效降低因 loss burst 進行封包重傳的機率，提升修復效能，因此 total traffic 能有效降低。究竟 SD+BI 增加的 traffic 量是否符合經濟效益，我們將在後面的 performance gain 進行探討。

3.3 封包遺失率 $PLR_{eff,0}$ 之討論

我們觀察實際 $PLR_{eff,0}$ 的變動情況(如圖 6)，四種 scheme 的 $PLR_{eff,0}$ 普遍受到 PLR_{ch} 變化的影響，隨著 PLR_{ch} 的增加而上升。而 N 的變化，對於 $PLR_{eff,0}$ 影響較低。

由實驗發現，在 DD 環境下，無法針對網路情況的變異即時進行調整，僅能提供給使用者固定的 UEP-FEC 保護，因此在 $PLR_{eff,0}$ 表現較差。而 SD+BI 能有效預測封包遺失率並打散網路 loss burst 的影響，並針對遺失率較高封包預先從多個 source peer 下載，因此效能表現較佳，也使得 $PLR_{eff,0}$ 趨近於 PLR_{target} ，但實際 $PLR_{eff,0}$ 仍與 PLR_{target} 有些微差距的主要因素可能是因為在封包到達 playback time 之前，

封包 e 的 PLR_e 來不及發出足夠的 request， PLR_e ，仍然無法達到 β_0 的要求因此 $PLR_{eff,0}$ 會略高於 PLR_{target} 。由上述內容可以發現 SD 與 BI 的傳輸策略有效降低 $PLR_{eff,0}$ ，因此能提升 C_{index} 。

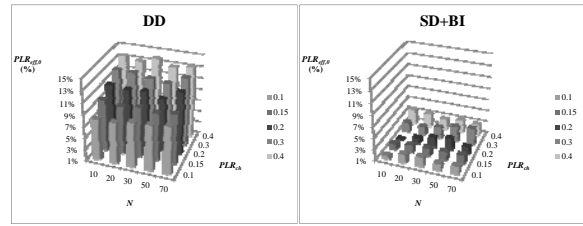


圖 6： $PLR_{eff,0}$ 之比較：(a)DD、(b)SD+BI。

3.4 影像品質比較

我們將進行兩種 scheme 的影像品質比較與 scheme 效能的 performance gain (G) 獲益比。假設在 RTT 參數 $\alpha = 2$ 、 $B = 9$ 、 $N = 50$ 、 $PLR_{ch} = 0.2$ 的環境之下，觀察兩種 scheme 差異。

Performance gain (G)：指單位傳輸成本所帶來的 PSNR 改善量，其公式定義如下：

$$G(\text{dB/Mbytes}) = \frac{PSNR_{SD+BI} - PSNR_{DD}}{T}$$

影像品質 PSNR，如圖 7 所示，我們透過 PSNR 來比較四種不同傳輸機制下的視訊品質。DD 僅能提供約為 27dB 的視訊品質。而 SD+BI 的部分，PSNR 值約 32.5dB。因為 SD 與 BI 能有效提升資料傳輸正確率與 UEP-FEC 效能，因此能有效提升 layer 1 與 layer 2 的播放時間，因此能提供給使用者較佳的視訊品質。

接著在 performance gain 的效益評估，在 SD+BI 的中，加入了 BI 的傳輸機制之後，有效提升了 UEP-FEC 解碼效能，因此能提供更好的 PSNR，並且降低 total traffic，提供給使用者較佳的系統環境，相較 DD 約能增加 2.1dB 的 PSNR。

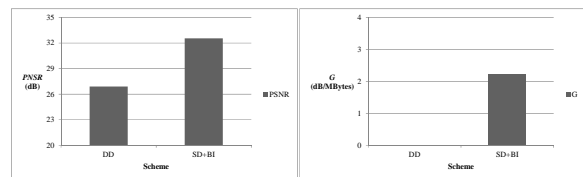


圖 7：影像品質比較：(a)PSNR、(b)Performance gain。

3.5 α 值對 SD+BI 之影響

接下來我們繼續探討 SD+BI 中，一些參數值對於它效能的影響。首先我們觀察 RTT 標準差倍數 α 值， α 值得變化主要影響封包 timeout 的時間長短，因此我們欲了解 α 值的增加是否對於封包傳輸效能有所影響，接著觀察不同 buffer size B 的變化對於系統整體效能是否有所提升。首先我們假設在 $B=9$ 、

$N=50$ 人的系統中，實驗不同 α 值在變動的 PLR_{ch} 環境中，效能的影響。

由實驗可以發現(如圖 8)，當 α 較低時，因為 timeout 時間短，封包接收機率降低，lag 發生的機會提升，所以需耗費較多的 total traffic 進行封包重傳。當 α 值的增加，透過 timeout 時間的增加，提升封包接收成功率，有效降低整體傳輸量 T ，提升視訊品質 PSNR，能用較低的傳輸量獲得較佳的視訊品質。但是由實驗中可以發現， α 值的提升，能增加 C_{index} 有限，而且當 α 值過大時，可能導致封包接收的 timeout 時間過長，當封包成功接收時，可能已經超過該封包的 playback time，因此為了避免此問題，必須選擇適當的 α 值，以確保串流資料下載的即時性。

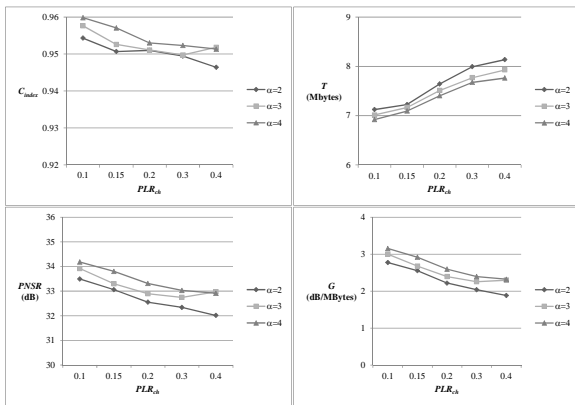


圖 8： α 值對 SD+BI 之影響：(a)Continuity index、(b)Total traffic、(c)PSNR、(d)Performance gain。

3.6 B 值對 SD+BI 之影響

在此我們將觀察在 B 值(9(≈ 15 秒)、12(≈ 20 秒)、15(≈ 25 秒)、18(≈ 30 秒)個 epoch)環境下，假設在 $\alpha = 2$ 、 $N=50$ 人、pre-cache size=3 的環境下，評估對系統影響如何。

如圖 9 我們可以發現，雖然 B 值會增加使用者的整體傳輸量 T ，但是能有效提升 C_{index} 與 PSNR 值，並且獲得較佳的 performance gain，當 B 值達到 15 與 18 時，buffer size 接近 request 所需的 interleaving depth d_s (模擬實驗所需 d_s 範圍為 5~20)，所以在 BI 封包派送排程時的 request 數量受限於 ER 範圍之中，因此對於系統效能改善有限。並且由於行動環境使用者的硬體資源有限，緩衝區空間並非能無限擴充，如行動環境使用者儲存空間有限，因此必須找到一個平衡點，能適應於各個種異質使用者。

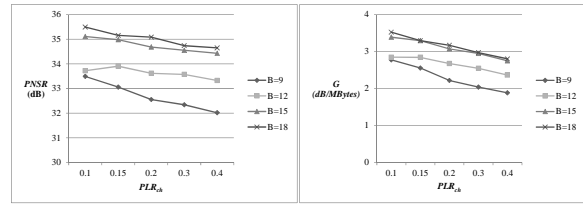
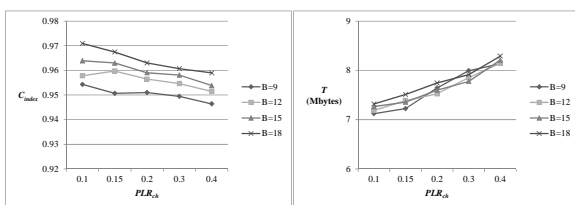


圖 9： B 值對 SD+BI 之影響：(a)Continuity index、(b)Total traffic、(c)PSNR、(d)Performance gain。

4. 結論

在本論文中，我們提出了「Statistical Download」與「Blockless Interleaving」的傳輸策略，在不同的 PLR_{ch} 環境下，能有效提供給 MP2P 使用者流暢的串流播放環境，減少 lag 中斷的時間。同時透過 UEP-FEC 的保護策略，在異質網路與異質平台的使用者，可以提供更彈性的影片品質。

實驗結果發現，在 SD 傳輸策略能有效提升資料傳輸的成功機率，使其效能優於 DD。另外結合了 BI 策略之後，能有效減少行動網路環境中的 loss burst 影響，使得 $PLR_{eff,0}$ 降低，因此可以達到較佳的影片播放流暢度(C_{index})、視訊品質(PSNR)。並且我們分析了 SD+BI 所能帶來的 performance gain 之後發現，雖然說整體傳輸成本增加，我們的成本效益事實上比 DD 高出約 2.1dB/Mbyte。因此在網路頻寬足夠的環境下，採用我們提出的機制能提供較佳的系統效能。

並且在 SD+BI 的環境下，我們分析了不同的 α 值與 B 值對系統的影響，雖然透過兩者的增加，皆能獲得較佳的系統效能，但是我們必須考慮到，串流資料即時性的問題，我們的 α 值亦不能無限擴張，避免 timeout 時間過長；並且由實驗結果發現，buffer size 接近 interleaving depth，request 數量將受制於 interleaving depth，所以再增加 B 時，能提升的系統效能有限，而且行動使用者手持裝置規格並非能無限擴張，我們仍須找到適當的平衡點，以提供高效能的視訊串流平台。

參考文獻

- [1] 童曉儒，許益晨，“為無線 P2P-UEP 串流而設計之傳輸與快取策略，” National Computer Symposium, National Chia Yi University, Dec. 2011.
- [2] J. B. Philip, “Chebyshev and Fourier spectral methods.” Courier Dover Publications, 2001.
- [3] J. Van, “Congestion avoidance and control,” ACM SIGCOMM Computer Communication Review, pp. 157-187, 1995.
- [4] S. R. Tong, H. W. Chen, C. L. Wu, and Y. J. Chang, “Adaptation of Near-Perfect Packet-Level FEC Interleaving in Mobile Media Streaming,” MINAR lab technical report, Dec. 2012.
- [5] U. C. Kozat, et al., “Unequal Error Protection Rate Less Codes for Scalable Information Delivery in Mobile Networks,” INFOCOM, pp. 2316-2320, 2007.
- [6] Wilhelmsson, Leif, and B. M. Laurence, “On the effect of imperfect interleaving for the Gilbert-Elliott channel,” Communications, IEEE Transactions, pp. 681-688, 1999.