

以分群平衡機制之動態緩衝策略提升同儕式網路隨選視訊串流效能

林朝興 黎俊達

國立台南大學

資訊工程所

Email: mikelin@mail.nutn.edu.tw

王冠升

南台科技大學

資訊管理所

Email: m9690107@webmail.stut.edu.tw

摘要—同儕式網路發展成為現今網際網路被廣泛應用的一種技術，帶來了頻寬與儲存空間的優勢，也漸漸取代傳統的主從式架構。以同儕式網路為基礎的隨選視訊服務，透過資源共享的模式，由系統內的節點提供新進節點加入時需要的影片片頭資訊，但如何有效的維持提供影片片頭的服務能力，也成為了隨選視訊系統的一大課題。本論文中，我們提出了以分群平衡機制之動態緩衝策略(Balanced Dynamic Buffering)，並採用多重描述編碼(Multiple Description Coding, MDC)技術對影片進行編碼，編碼過後的影片將產生多條品質相同的描述串流，客戶端可依自身需求或頻寬限制，選擇所需要的影片描述串流個數。本論文之模擬實驗顯示分群平衡機制之動態緩衝策略比其它類似機制更能顧全系統中的資源平衡，延長影片片頭的可服務時間，改善伺服器端的頻寬耗用，提升系統整體擴充性。

關鍵字—同儕式網路、多重描述編碼、分群、動態緩衝區、擴充性

一、前言

隨著資訊科技的快速發展，使用者對多媒體資訊的要求也愈來愈高。同儕式網路(Peer-to-Peer Network)快速發展[2][7]，採用其架構的軟體或技術也相繼推出，例如BT、Foxy或PPStream等[12][16]。此外，開始有許多的研究議題討論如何利用同儕式網路及多媒體串流傳輸技術的結合[6][15][20]，提供良好的影片觀看品質。同儕式網路不同於傳統的主從式架構(Client-Server model)[1][5][14]，其做法是透過節點間彼此在傳輸影片時的資源共享，扮演的角

色可分為「伺服器端」(Server)和「客戶端」(Client)，節點間的協同合作不僅可降低伺服器的頻寬耗用，得到更佳的系統擴充性(scalability)，也為伺服器端節省更多的硬體成本和服務更多的使用者請求。

現今常見的同儕式網路與多媒體串流傳輸技術的結合，可區分為以下兩類：

- 隨選視訊(Video on Demand, VoD)：以預錄方式且不具即時性的多媒體內容為主[13][17]。每位使用者在任何時間加入系統，皆可以觀賞到完整的影片內容，達到隨選隨看之目的。
- 即時串流(Live Streaming)：屬於直播性質的多媒體串流服務。使用者在任何時間加入系統時，由加入的時間點開始觀看最新影片內容[11]。系統不需要提供完整的影片內容，僅提供新進使用者加入系統時撥放的影片內容。

但使用者對影片觀看的需求是從影片開始至影片結束，故影片片頭的取得與維護成為首要研究的議題。一般在同儕式網路上提供隨選視訊的方法是透過節點將所看過的影片串流儲存於本身的緩衝區(buffer)，當有其它節點提出請求時，再轉送給其它有需要的節點，此種方法稱之為“cache-and-relay”[4]。

同儕式網路在多媒體串流服務上帶來許多的優勢，透過節點間的資源共享，降低伺服器端的負載。但同儕式網路仍然存在一些詬病，為達到理想化的同儕式多媒體串流服務必須解

決以下議題：

- (1) 異質性及非對稱式的同儕式網路：由於每個使用者來自於不同的連線環境，例如 Ethernet、DSL 或 Cable Modem，而在同儕式網路上觀看多媒體串流的品質往往會因為傳送端的上傳頻寬而有直接的影響。最常見的情況是接收節點有充裕的上傳頻寬，但其它的傳送節點未必有足夠的上傳頻寬，會造成傳輸時無法達到接收端所要求的影片觀看品質。
- (2) 系統資源的有效利用：在多媒體串流服務的同儕式網路環境中，節點對於影片串流的存取是連續且不中斷的，而伺服器端的上傳能力有限，若無法有效運用節點資源或維持影片片頭的服務，大量客戶端的非同步請求，將導致伺服器端負載過重，降低系統服務的擴充性。因此，我們必須善加規劃網路環境中的所有資源，以增加系統的服務人數，延長其生命週期。
- (3) 多媒體串流資源失衡：在同儕式網路中存在不同頻寬的網路連線，但多媒體串流服務提供單一固定位元率的影片，並不適用於異質性(heterogeneity)的同儕式網路[19]。故為解決上述所衍生的問題，藉由多重描述編碼(Multiple Description Coding, MDC)技術，將原始影片串流進行編碼，產生數個子串流分配給網路上其它的節點。若網路上的節點所持有的子串流數量不平衡時，新進節點無法獲得所需要的串流描述，此時將由伺服器端提供串流服務給新進節點，也增加了伺服器的負擔。

有鑑於目前在同儕式網路上所提供的服務缺乏一套完善且具彈性的動態緩衝機制。在本篇論文中，我們採用節點分群法，藉由平衡節點的描述串流分佈，以節點暫存描述串流數量進行群組分類，亦即系統內的節點所持有相同數量的影片描述串流為一群組，提出「以分群

平衡機制之動態緩衝策略」(Balanced Dynamic Buffering)。

本篇論文架構如下：第二章節為相關文獻探討，第三章節介紹動態緩衝機制在節點分群狀態下達到資源平衡概念的實例探討，第四章節介紹平衡式動態緩衝策略在多媒體串流系統環境上的應用，第五章節為動態緩衝區的效能評估，最後，第六章為本論文之結論及未來研究。

二、文獻探討

2.1 多重描述編碼(Multiple Description Coding)

MDC[10]編碼技術的原理是將一個原始影片串流進行編碼成多條品質相同的串流描述(description)，每一個description都是獨立解碼並有唯一的編號。節點只要接收到任意數量的description即可進行解碼，其特色能滿足同儕式網路上的節點提供數種不同位元率的多媒體串流。使用者可依照自身的下載頻寬與需求，像伺服器提出需求；所接收到的description數目愈多，解碼後的影片串流品質就會愈好，但重複接收同一編號的description無法增加其影片的觀看品質。

2.2 CoopNet

CoopNet[18]結合主從式及同儕式兩種網路架構。在CoopNet中，處理每一個使用者的加入或是當有來源節點離開時所進行的錯誤回復，都是採用中央集權的管理方式。首先，伺服器端將完整影片使用MDC技術進行編碼成多條不同編號的description。串流的傳輸是以結合架構式應用層群播與點對點應用層群播特性的混合式應用層群播樹(Application Level Multicast Tree, ALM Tree)為系統主體架構，由節點形成一個樹狀結構，透過伺服器端分配description到多棵樹中進行群播。當有新的使用者加入系統欲

取得影片服務，必須先與伺服器取得聯繫，並告知本身所需要的description數目以及願意上傳的description數量。伺服器端取得該使用者的資訊之後，再統一由伺服器配置該使用者所需的description與來源節點。CoopNet為延伸系統擴展性，允許節點向數個父節點請求服務，透過節點之間的資源共享使其達到提升影片觀看品質。

CoopNet雖能有效降低伺服器端的負載，且有較佳的系統擴充性，但並非是完美無缺的。建構在應用層群播樹的多媒體串流服務可能會面臨節點的服務能力將會受到緩衝區長度的限制、龐大且長時間的頻寬支援及節點連線異質性與串流資源不平衡的挑戰。為解決上述缺失，我們選擇可效度較高與數量多的description為挑選要點，並在選擇節點時，以節點的服務能力為優先考量。綜合以上挑選考量指標，進而達成可擴展與高效率的多媒體串流系統。

2.3 均等式動態緩衝機制 (Even Dynamic Buffering)

均等式動態緩衝機制[3]是當節點緩衝區處於滿載狀態時，透過丟棄緩衝區中未上傳的description，並將其所釋放的緩衝區空間均等分配給仍暫存在緩衝區中的其它description，以延長影片片頭可服務時間。

隨著動態調整節點緩衝區的程序，緩衝區暫存的description數量遞減，如圖1所示。假設伺服器端將一部3分鐘影片編碼成4條description，P0於第0分鐘時加入系統並向伺服器端請求服務。此時，P0暫存4條description且持有影片片頭的緩衝能力為3分鐘。而在第3分鐘時，P0的緩衝區呈現滿載狀態，假若P1於第12分鐘時加入請求時，P0無法服務影片片頭資訊，為了延長影片片頭服務並等待新進節點加入時的請求，可將目前緩衝區中某一條description丟棄，

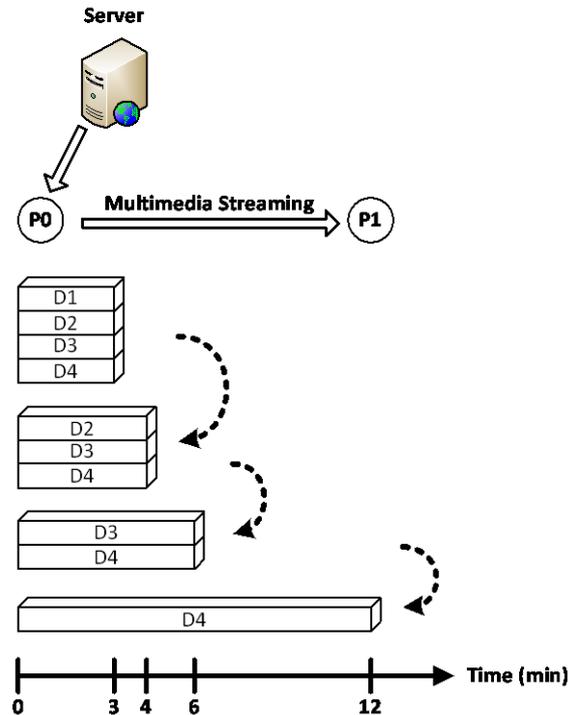


圖 1：P0 進行動態調整緩衝區程序

並將其暫用的空間釋放並平均分配給其它的description，經過均等分配後，各description的可服務時間延長為4分鐘。

以相同方法持續進行動態調整節點緩衝區程序，當緩衝區僅剩一條description時，節點所持有影片片頭的時間延長為12分鐘。當P1於第12分鐘加入時，由於P0仍持有影片片頭資訊，故P0可向P1請求服務，而剩下未滿足其觀看的影片內容則由伺服器端進行服務。

在均等式動態緩衝機制中，當緩衝區呈現滿載，漸進式的丟棄所持有的description，直到節點所持有的description僅剩1條。

2.4 伸縮式動態緩衝機制 (Elastic Dynamic Buffering)

伸縮式動態緩衝機制[3]是為了更進一步的延長節點服務新進節點的時間，改善均等式動態緩衝機制中的不足，使緩衝區空間的利用不受description狀態而有所限制，將節點的等待服

務時間延伸至最大長度。此機制是當節點在緩衝區滿載時，其可優先針對「已上傳description」的可利用緩衝區空間進行丟棄，其次為「未上傳description」。

雖然於伸縮式動態緩衝機制下考慮description的平衡，能有效降低頻寬負載，增加系統的服務能力。此機制仍無法將其效能發揮至最大，因為伸縮式動態緩衝機制在各方面雖優於CoopNet與均等式動態緩衝機制，但仍然無法有效延長影片片頭的服務。

為了有效改善伸縮式動態緩衝機制中資源平衡效能不佳之問題，我們進一步提出平衡式動態緩衝機制(Balanced Dynamic Buffering)來提升與強化對description資源平衡的性能。

三、資源平衡

為了改善伸縮式動態緩衝機制中資源平衡效能不佳，避免不正確的系統資源估計與有效降低頻寬耗用，我們提出分群平衡機制來提升系統效能，輔助緩衝區執行description丟棄的程序，以「集合(Group)」為單位使系統資源平衡化，並將系統內即將失去暫存有影片片頭能力的節點優先分派給新進節點作為父節點，善用系統內所有節點的資源，將有利新進節點選擇父節點，使其快速找到適合的父節點。

3.1 節點分群

為了避免不正確的系統資源估計與有效降低頻寬耗用，我們提出分群平衡機制來提升系統效能，將系統規劃成數個Group，每個Group是以節點暫存的description數量進行分群。

在動態緩衝區方法中，節點執行description丟棄的程序時，將優先保留目前系統上最為欠缺的description，且因為每個節點緩衝區內所暫存description數量及服務時間能力的不同，若忽略節點的特性，則將影響系統資源的平衡。

3.2 新進節點加入

當新進節點需要影片串流服務時，該節點會先與伺服器端取得聯繫，然後再由伺服器端進行影片串流的分配。當伺服器進行節點的分群時，將會依據節點提出要求description的數量來決定節點所加入的Group。

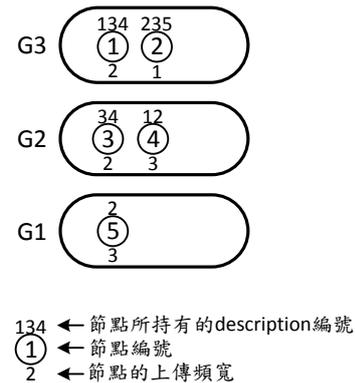


圖 2：節點是以所暫存的description 數量來進行分群

如圖2所示，假設目前有一新進節點P6(上傳頻寬為3)欲加入系統請求服務，該節點必須主動向伺服器端提供本身所需要的description數量，若需要3條description，以分群概念來分析，此新進節點會選擇加入G3。

3.3 內部群組平衡(Intra-group balancing)

當新進節點加入後，必須決定該節點要接收哪幾條description，此程序稱為「Description Selection Approach」。以圖2為例，新進節點P6需要3條description並選擇加入G3，計算該Group對每條description能提供的上傳頻寬總合，再依序從中挑選上傳頻寬最小的description作為新進節點要接收的第一條description，再決定該由哪一節點來提供此description的服務。

3.4 外部群組平衡(Inter-group balancing)

新進節點加入系統時，主動提供本身所需description 的數量及其相關資訊給伺服器，在Intra-group balancing已決定新進節點欲接收第一條description，後續仍需找到可貢獻description

服務的節點，此程序稱為「Peer Selection Approach」。為了避免節點同時服務多條的description，我們限制同一節點僅能服務一個節點1條description。以圖3為例，假設P0向伺服器端取得3條description，隨後P1加入系統請求服務，經過Intra-group balancing決定接收d1及d2，雖然P0暫存3條description，但僅能提供其中1條description 給P1，而P1所需的另一條description 需由其它節點或伺服器端取得服務。圖4為新進節點系統執行Description Selection Approach與Peer Selection Approach 的演算法。

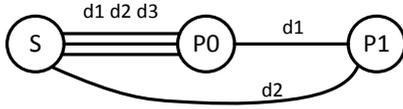


圖 3：P0 僅能服務 P1 的 1 條 description

四、平衡式動態緩衝策略

(Balanced Dynamic Buffering)

在同儕式網路上提供多媒體串流服務時，客戶端的緩衝區大小是有限的，並且對於整體系統的擴充性有決定性的影響。動態緩衝機制的提出是為了能解決節點緩衝區不足，而無法提供新進節點服務的問題。藉由動態緩衝機制的應用，可使系統中仍持有影片片頭的節點，動態的調整緩衝區內已上傳的影片所佔用的暫存區空間，以延長等待新進節點加入的時間爾後進行服務。

不同於伸縮式動態緩衝機制，平衡式動態緩衝機制的提出能更進一步的平衡網路上節點的description資源，使緩衝區空間不受description狀態而有所限制，並將節點的等待時間延伸到最大。此機制是當節點在緩衝區的description空間屬於滿載狀態時，透過丟棄緩衝區中已上傳及未上傳的description，以延長其它description的影片片頭長度。

p_{new} : a newly arrived peer

$D = \{d_1, d_2, \dots, d_m\}$: the set of total descriptions

G_i : the group in which each peer preserves exactly i descriptions

$H(p_i)$: the set of descriptions p_i preserves

$Q(p_i)$: the viewing quality required by p_i

$F(p_i)$: the available forwarding bandwidth of p_i

$S(p_i)$: the set of descriptions p_i selects

$R(p_i)$: the set of descriptions p_i receives

$U(p_i)$: the set of p_i 's parent peer

$A_i(d_j)$: the availability of description d_j in group G_i

Description / Peer Selection Approach(p_{new})

- 1 $G_i \leftarrow G_i \cup \{p_{new}\}$, where $i = Q(p_{new})$
- 2 **for** each description $d_j \in D$
- 3 $A_i(d_j) \leftarrow \sum_{p_k: p_k \in G_i, d_j \in H(p_k)} F(p_k)$
- 4 $S(p_{new}) \leftarrow \phi$
- 5 **repeat**
- 6 find a description d_x , s.t. $x = \arg \min_{j: d_j \in (D - S(p_{new}))} A_i(d_j)$
- 7 $S(p_{new}) \leftarrow S(p_{new}) \cup \{d_x\}$
- 8 find a group G_y , s.t. $y = \arg \max_{j: 1 \leq j \leq m} A_j(d_x)$
- 9 find a peer p_k , s.t. $k = \arg \max_{j: p_j \in (G_y - U(p_{new})), d_x \in H(p_j)} F(p_j)$
- 10 **if** $p_k \neq NULL$
- 11 $F(p_k) \leftarrow F(p_k) - 1$
- 12 $R(p_{new}) \leftarrow R(p_{new}) \cup \{d_x\}$
- 13 $U(p_{new}) \leftarrow U(p_{new}) \cup \{p_k\}$
- 14 **else if** $F(SERVER) > 0$
- 15 $F(SERVER) \leftarrow F(SERVER) - 1$
- 16 $R(p_{new}) \leftarrow R(p_{new}) \cup \{d_x\}$
- 17 **else if** $S(p_{new}) = D$
- 18 reject p_{new}
- 19 terminate
- 20 **until** $|R(p_{new})| = Q(p_{new})$

圖 4：Description/Peer Selection Approach 演算法

4.1 節點的管理

系統為有效的管理節點執行動態調整的程序，透過伺服器端實施中央管理，降低新進節點向伺服器端請求的次數。節點執行動態調整緩衝區的程序時，必須考慮系統內description平衡分佈的特性，以維持節點提供Full-Quality影片

串流的能力。

在動態緩衝策略中，影片串流傳輸前需以MDC編碼技術來進行編碼，將原始影片串流分割成數個相同位元率的description。每條description有固定且不重複的編號。新進節點加入系統請求服務時，必須提供其相關資訊，再由伺服器端控制並進行動態調整的相關程序，兩者執行程序如下：

- (1) Peer：當新進節點抵達系統時，必須主動提供相關資訊，包含可利用的緩衝區大小、所需的影片觀看品質及上傳頻寬。
- (2) Server：伺服器端為整體網路架構的中心，由它實施中央控制及管理。建置Cache Information Table (CIT) 管理與記錄系統內的所有節點資訊，包含節點加入時間、description狀態及數量，並同時計算節點執行動態調整的時間與剩餘頻寬。CIT內的資訊須持續的追蹤與更新，直到節點的緩衝區不再提供片頭資訊。

4.2 描述串流的丟棄

動態緩衝區方法中以緩衝區內暫存的description數量作為節點的分群準則，目的為使持有的description數量相同的節點資源均勻分布，平衡不同編號的description數量。因此，當節點執行動態調整緩衝區時，必須考慮調整後的Group中節點持有資源的數量決定description的丟棄與否。

節點在進行調整及丟棄description之前，需透過伺服器端所建置的CIT得知節點服務的子節點數量及已上傳的description，以判斷節點的description貢獻情形。若一個節點緩衝區處於滿載狀態時，且在系統中屬於葉節點，則從目前持有的description選擇一條進行丟棄。但若此節點已有服務其它節點，目前緩衝區內仍存在閒置空間可進行調整並有剩餘的上傳頻寬，則此節點的緩衝區同時存在已上傳及未上傳的

description。當節點的緩衝區是滿載的，節點會評估目前description的分布情況，找出目前系統資源中最充裕的description進行丟棄。圖5為平衡式動態緩衝策略於丟棄描述串流的演算法。

p_i : the peer about to perform the description dropping procedure
 $H(p_i)$: the set of descriptions p_i preserves in its buffer
 $T(p_i)$: the set of descriptions p_i preserves and has forwarded to other peers
 G_j : the group p_i belongs to

Description Dropping(p_i)

- 1 for each description $d_k \in (H(p_i) - T(p_i))$
- 2 $A_{j-1}(d_k) \leftarrow \sum_{p_x: p_x \in G_{j-1}, d_k \in H(p_x)} F(p_x)$
- 3 find a description d_k ,
s.t. $d_k = \arg \max_{d_n \in (H(p_i) - T(p_i))} A_{j-1}(d_n)$
- 4 $H(p_i) \leftarrow H(p_i) - d_k$
- 5 $G_j \leftarrow G_j - p_i$
- 6 if $j > 1$
- 7 $G_{j-1} \leftarrow G_{j-1} \cup p_i$

圖5：Description Dropping 演算法

五、模擬實驗

本章節中藉由模擬實驗分析比較CoopNet、Even Dynamic Buffering、Elastic Dynamic Buffering與Balanced Dynamic Buffering的效能。每分鐘到達的節點所要求description個數比例如表1，表2為本模擬實驗的相關參數設定。

表 1：每分鐘節點要求 description 個數比例

description個數(條)	2	4	6	8	10	12	14	16
要求比例(%)	21	18	18	15	10	10	5	3

表 2：實驗參數設定

Parameters	Value
影片長度(min)	120
description個數(條)	16
伺服器頻寬(description)	5000
節點到達率(request/ min), λ	1~10
緩衝區長度(min)	2~10, 2~32

5.1 伺服器端的頻寬耗用

藉由變動節點請求到達率及節點緩衝區長度，分析三種不同相異的動態緩衝機制及CoopNet下伺服器端頻寬的耗用情況。此一指標可顯示有無採用動態緩衝機制時，已存在系統內節點是否有能力服務新進節點的影片請求。換言之，其可衡量節點選擇不同熱門程度的影片(熱門電影及冷門電影)發出請求對伺服器端所造成的負載量。

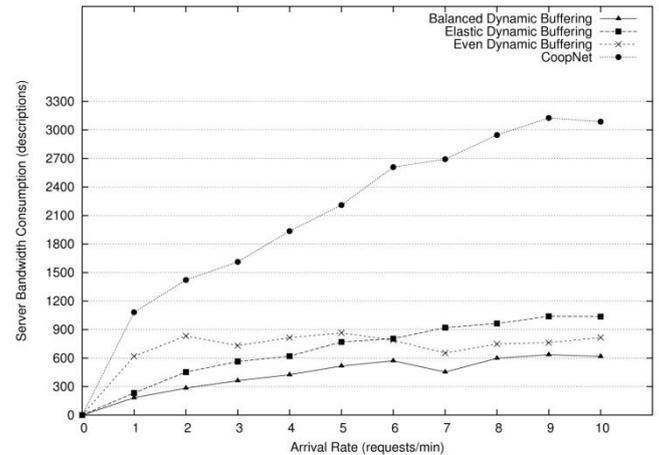


圖 6：當節點到達率為 1~10(request/min)且緩衝區長度為 2~10 分鐘

如圖6所示，由於伺服器端採用即到即服務的模式，在任一時間，只要節點發出請求都可以馬上觀賞影片。在此實驗結果中，可觀察到在CoopNet環境下，提高節點到達率，伺服器端的頻寬使用率會呈現上升狀態。

在CoopNet中，由於節點緩衝區暫存影片片頭的長度不足等待新進節點的加入，新進節點轉而向伺服器端請求服務。相較於CoopNet，動態緩衝機制透過節點動態調整緩衝區內的description資源，增加節點的可用性，降低隨選視訊服務中伺服器端負載過重的問題。這是由於MDC和動態緩衝區互相搭配的成果。

在動態緩衝區中，則可透過丟棄description，減少緩衝區中暫存的description數量，釋放出儲存空間，延伸緩衝區內可以保有影片片頭的時間。動態緩衝區用漸進式的調整緩衝區內的description數量，以延伸影片片頭可得時間，避免節點的緩衝區空間內沒有保留任何一條description片頭的情況。即使當節點到達率增加時，亦可透過節點動態調整及資源平衡，減低伺服器端的負載。

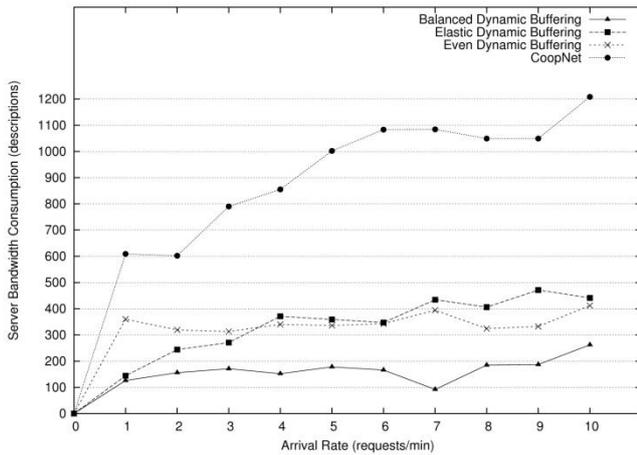


圖 7：當節點到達率為 1~10(request/min)且緩衝區長度為 2~32 分鐘

如圖7緩衝區長度為2~32分鐘，可觀察到三種動態緩衝機制與CoopNet所耗用的伺服器端頻寬有顯著改善，如同所有同儕式隨選視訊服務一樣，當節點的緩衝區長度愈長，節點會有較充裕的儲存空間暫存影片片頭，換言之，其節點等待新進節點的請求時間較長。

在CoopNet與三種動態緩衝機制中，當節點在高到達率的情況下，Balanced Dynamic Buffering更能有效延長影片片頭的服務，並有效降低伺服器頻寬的耗用。

5.2 系統內動態調整節點個數

本實驗主要統計在不同到達率 1~10(request/min)環境下，緩衝區長度固定為15分鐘，意即每個節點具有15分鐘一定的緩衝區長度。每當節點緩衝區滿載時，節點必須動態調整其緩衝區長度以延長影片片頭的服務。當同一節點受到動態調整時，統計為1次調整次數，即1個節點受到動態調整，無論被調整幾次，仍計為1次。本實驗將針對Balanced Dynamic Buffering、Even Dynamic Buffering與Elastic Dynamic Buffering三種相異動態緩衝機制作比較。因CoopNet不具有動態調整技術，故於本實驗將不適合分析及評估。

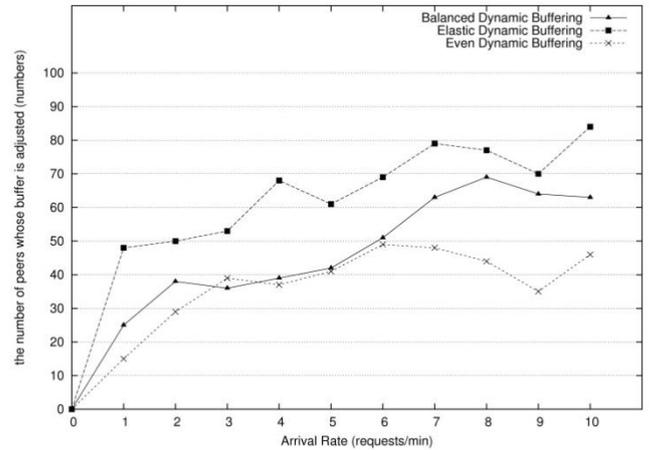


圖 8：當節點到達率為 1~10(request/min)且緩衝區長度為 15 分鐘

如圖8，藉由實驗觀察發現Even Dynamic Buffering在調整節點個數上比Balanced Dynamic Buffering與Elastic Dynamic Buffering的調整個數少，這是因為Even Dynamic Buffering動態調整緩衝區時僅調整「未上傳」之description，所以被調整的個數相對減少。然而，Elastic Dynamic Buffering動態調整時，會優先挑選description狀態為「已上傳」的來丟棄，很容易造成閒置緩衝區的空間不足，在下一秒後所有的description皆無法保有影片片頭。

Balanced Dynamic Buffering由於在Intra/Inter-balancing group即考量系統內節點的description資源與上傳頻寬，所以Balanced Dynamic Buffering在最佳化挑選description與貢獻節點狀態下，平衡動態調整「未上傳」與「已上傳」之description，節點調整個數雖低於Elastic Dynamic Buffering，但實際能有效延長影片片頭服務。

5.3 系統內節點緩衝區動態調整次數

本實驗主要統計在不同到達率 1~10(request/min)環境下，緩衝區長度固定為15分鐘，意即每個節點具有15分鐘一定的緩衝區長度。每當節點緩衝區滿載時，節點必須動態調整其緩衝區長度以延長影片片頭的服務。當1

個節點的緩衝區被調整5次，計為5次。換言之，一旦節點被調整，則遞增累計次數。本實驗將針對Balanced Dynamic Buffering、Even Dynamic Buffering 與Elastic Dynamic Buffering三種相異動態緩衝機制作比較。同樣地，因CoopNet不具有動態調整技術，故於本實驗將不適合分析及評估。

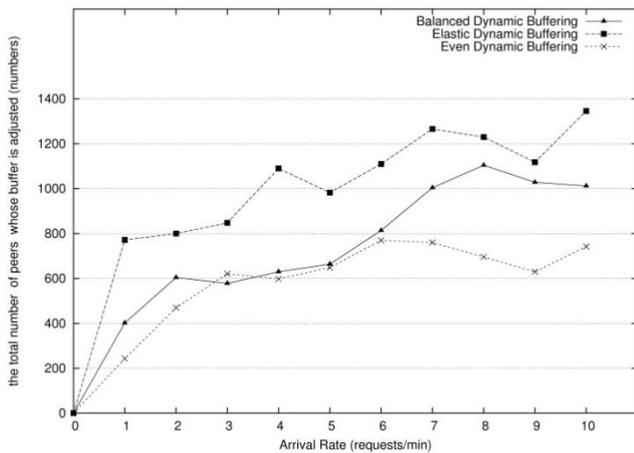


圖 9：當節點到達率為 1~10(request/min)且緩衝區長度為 15 分鐘時

如圖9，可以觀察到無論是節點調整個數或總計調整次數大小皆為Elastic > Balanced > Even。原則上節點緩衝區被調整次數愈多愈好，但Elastic Dynamic Buffering的選擇調整方式，反而會讓緩衝區可利用的空間被愈調愈少。

Even Dynamic Buffering動態調整緩衝區時僅調整「未上傳」之description，所以被調整的次數相對減少。然而，Elastic Dynamic Buffering動態調整時，會優先挑選description狀態為「已上傳」的來丟棄；而Balanced Dynamic Buffer具有一套最佳的描述串流丟棄方法來選擇丟棄「未上傳」或「已上傳」之description，即使調整次數沒有如Elastic Dynamic Buffering多，Balance Dynamic Buffering可以具體的延伸其它description的片頭保存時間。在子節點進入系統時，可以有多個可用父節點可供選取。故節點中的description被調整的次數會增加，並能有效降低伺服器端負載。

5.4 系統內節點緩衝區被動態調整的長度

本實驗主要統計在不同到達率 1~10(request/min)環境下，取樣相同緩衝區長度為15分鐘。當緩衝區長度受限時，節點執行動態調整程序的比例相對增加。使用不同的調整機制時，其節點經過動態調整後的緩衝區空間的及平衡狀況都是值得觀察的議題。每當節點緩衝區滿載時，節點必須動態調整其緩衝區長度以延長影片片頭的服務。考量調整後的Group中節點持有資源的數量以決定description的丟棄，該丟棄之description的空間為本項實驗指標，單位為分鐘。本實驗將針對Balanced Dynamic Buffering、Even Dynamic Buffering與Elastic Dynamic Buffering三種相異動態緩衝機制作比較。

一個節點持有影片片頭，且仍有剩餘的上傳頻寬即符合成為父節點的條件。從系統面來說，當可利用的緩衝區空間愈多時，代表系統對於新進節點的請求有充裕的服務能力，亦即節點可提供的緩衝區可利用空間足以處理新進節點的請求，也間接提升了伺服器端的擴展性。

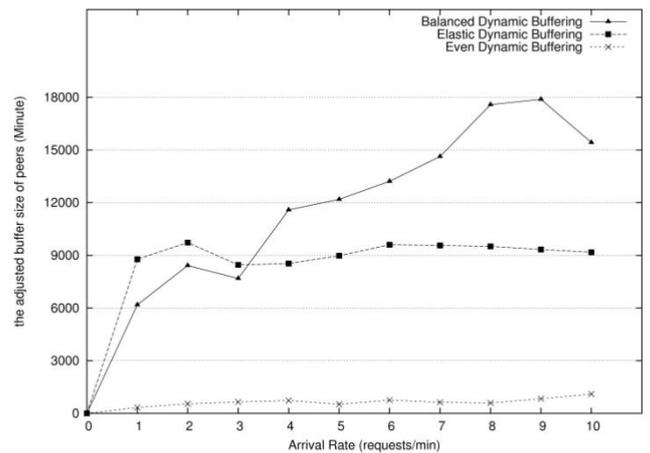


圖 10：當節點到達率為 1~10(request/min)且緩衝區長度為 15 分鐘

如圖10，Even Dynamic Buffering相較於其它方法，其調整的量也較其它兩種動態緩衝機制少許多，突顯此方法中未使用平衡機制將導

致description分布不平均之疑慮，因此系統內的節點將提早喪失影片片頭服務的能力。Balanced Dynamic Buffering中，節點與description的分布較為平均，當節點可提供的description數量較少，新進節點抵達系統時，基於Balanced Dynamic Buffering的資源平衡概念，當調整的量愈多，維持description的平衡將可滿足新進節點不同影片品質的請求。

六、結論

本篇論文提出分群平衡機制之動態緩衝策略。並採用MDC技術解決客戶端非對稱式的網路連線、客戶端的異質性以及客戶端的變動性，在解決上述議題的同時，更進一步將系統中的資源善加利用。利用平衡機制之動態緩衝策略，節點透過動態調整緩衝區內暫存的description內容，延長節點等待新進節點的時間，改善系統整體效能，亦增加系統擴充性。

未來研究方面，本論文於同儕式網路架構下所提出的平衡式動態緩衝機制，由於節點之間的資源共享是以同儕式網路架構為主體，因此父節點的離開，將會影響子節點影片的接收。為了避免節點離開對於系統所造成的影響，因此提出一個更有效率的錯誤回復機制將是本篇論文未來研究將探討的主要議題。

七、參考文獻

- [1] A. Dan, D. Sitaram, and P. Shahabuddin. Scheduling policies for an on-demand video server with batching. In Proc. of ACM Multimedia, pages 15–23, 1994.
- [2] Antony I. T. Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In Proc. of ACM Distributed Systems, pages 329–350, 2001.
- [3] C.-S Lin and Liang-Chun Wang. Improving the availability of streams by dynamic buffering on vod p2p network. In Proc. of The 5th Workshop on Grid Technologies and Applications, pages 9–16, 2008.
- [4] Chow-Sing Lin, Wei-Ting Syu, and I-Ting Lee. Improving the scalability of p2p streaming based on fine-grained balancing scheme. In Proc. of Advanced Information Networking and Application, pages 795–802, 2008.
- [5] Duc A. Tran, Kien A. Hua, and Tai Do. Zigzag: an efficient peer-to-peer scheme for media streaming. IEEE Trans. on Computer and Communications Societies, Vol.2, pages 1283–1292, 2003.
- [6] Ewa Kusmierk, Yingfei Dong, and David H. C. Du. Loopback: exploiting collaborative caches for large-scale streaming. IEEE Trans. on Multimedia, 8(2), pages 233–342, 2006.
- [7] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. Springer Berlin, 2001.
- [8] Kien A. Hua, Ying Cai, and Simon Sheu. Patching: a multicast technique for true video-on-demand services. In Proc. of ACM International Multimedia Conference, pages 191–200, 1998.
- [9] Li-Shen Juhn and Li-Ming Tseng. Harmonic broadcasting for video-on-demand service. IEEE Trans. on Broadcastcasting, 43(3), pages 268–271, 1997.
- [10] Miguel Castro, Peter Druschela, Anne-Marie Kermarrec, Animesh Nandi, Antony Rowstron, and Atul Singh. Splitstream: high-bandwidth multicast in cooperative environments. In Proc. of ACM SOSP, pages

- 298–313, 2003.
- [11] Ming Zeng, Hongliang Yu, Feng Yang, and Weimin Zheng. Dynamically optimize multicast tree structure. In Proc. of IEEE Grid and Cooperative Computing Workshops, pages 143–150, 2006.
- [12] Matei Ripeanu. Peer-to-peer architecture case study: Gnutella network. In Proc. of IEEE Peer-to-Peer Computing, pages 99–100, 2001.
- [13] Tai T. Do, Kien A. Hua, and Mounir A. Tantaoui. P2vod: Providing fault tolerant video-on-demand streaming in peer-to-peer environment. In Proc. of IEEE International Conference on Communications, pages 1467–1472, 2004.
- [14] S. Viswanathan and T. Imielinski. Metropolitan area video-on-demand service using pyramid broadcasting. In Proc. of ACM Multimedia Systems, pages 197–208, 1996.
- [15] Suman Banerjee, Bobby Bhattacharjee, and Christopher Kommareddy. Scalable application layer multicast. In Proc. of ACM SIGCOMM, pages 205–217, 2002.
- [16] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble. A measurement study of peer-to-peer file sharing systems. In Proc. of Multimedia Computing and Networking, 2002.
- [17] Santosh Kulkarni and John Markham. Split and merge multicast: live media streaming with application level multicast. In Proc. of IEEE Communications, pages 1292–1298, 2005.
- [18] Venkata N. Padmanabhan, Helen J. Wang, Philip A. Chou, and Kunwadee Sripanidkulchai. Distributing streaming media content using cooperative networking. In Proc. of ACM NOSSDAV, pages 177–186, 2002.
- [19] Xiangrong Tan and Suprakash Datta. Building multicast trees for multimedia streaming in heterogeneous p2p networks. In Proc. of IEEE Systems Communications, pages 141–146, 2005.
- [20] Yang Chu, Sanjay Rao, Srinivasan Seshan, and Hui Zhang. Enabling conferencing applications on the internet using an overlay multicast architecture. In Proc. of ACM SIGCOMM, pages 55–67, 2001.