

藉由平衡 Description 分佈與節點負擔提升 P2P 網路效能*

Enhancing P2P Performance by Balancing Description Distribution and Peer Loading *

林朝興 徐維廷 鄭義繼

南台科技大學 資訊管理研究所

E-mail : { mikelin, swkca, yasashi }@msrg.mis.stut.edu.tw

摘要

由於多媒體串流的風行以及 P2P 帶來了頻寬與儲存空間的優勢，使得愈來愈多的服務提供者開始在 P2P 網路上提供多媒體串流。雖然在 P2P 網路上提供多媒體串流能夠節省伺服器端的成本，但是同時也引發了許多的議題。本篇論文將針對在 P2P 網路上提供即時性多媒體串流(live streaming)服務時，所引發的議題做更進一步的探討。本論文中，我們將採用多重描述編碼(Multiple Description Coding, MDC)技術進行影片的編碼。編碼過後的影片將會產生多條品質相同的影片串流(description)，客戶端將可依照本身的需求與頻寬的限制，選擇所需要的 description 個數。透過這樣的編碼技術克服客戶端非對稱式的網路連線、異質性和變動性所衍生的問題。同時，藉由平衡每條 description 在網路上的分佈，減少對伺服器頻寬的使用，降低拒絕的人數，以達到最佳化系統擴充性。

關鍵詞：Peer-to-Peer network, live streaming, MDC, scalability

1. 前言

P2P(Peer-to-Peer)網路帶來了大量的計算與儲存資源，在初期的應用大部份著重於檔案分享上的傳輸應用[7][19][21]。因為 P2P 網路的優勢可以降低在伺服器端的成本，例如：儲存空間、頻寬…等。因此，P2P 網路在現今的應用上愈來愈受到歡迎與重視，並且逐漸取代傳統的主從式架構。在多媒體串流普及的現今，開始有許多的研究討論如何在 P2P 網路上提供多媒體串流服務[2][6][22]。相較於在傳統

的主從式架構中提供多媒體串流服務[10][13][15][23]，在 P2P 網路上可以為伺服器端節省更多的硬體成本和容納更多的使用者請求。

目前在 P2P 網路上提供多媒體串流服務可以區分成以下兩類：

- (1) **VoD(Video on Demand)**。VoD 稱之為隨選視訊，每一個使用者不論在任何時間加入系統，都希望能夠觀賞到整部完整的影片並且隨選隨看[9][11]。因此，在 P2P 網路上提供 VoD 的服務時，為了要滿足使用者的要求，所以在傳送影片內容時必須要考慮到影片播放時間的順序。
- (2) **Live streaming**。Live streaming 屬於直播性質的即時性多媒體串流服務。在即時性的多媒體串流中，使用者所在意的是最新的觀賞進度[8][25]，例如：棒球比賽或是新聞節目。因此，在 P2P 網路上提供 live streaming 時，系統並不需要提供給使用者完整影片內容，只需要提供給新進的使用者目前系統正在播放的影片內容。

在 P2P 網路上提供 VoD 服務時，因為伺服器端所提供的是完整的整部影片串流。所以，為了降低伺服器端頻寬與儲存空間的使用，伺服器端必須藉由網路上的 peer 來幫忙影片串流的儲存與轉送。通常 peer 會將所看過的影片串流存放於本身的儲存空間，再轉送給其他有需求的 peer，這種傳送方式在大多數的論文中定義為“cache-and-relay”[4][14]。

而在 P2P 網路上提供 live streaming 服務時，所要考慮的因素將會比提供 VoD 服務還要單純。因為在 live streaming 的系統中，每一個使用者所觀看的影片時間點會是一致的，所以在網路上的 peer 並不需要將所看過的影片存放於本身的儲存空間。在系統中的每一個 peer 只需要考慮本身的頻寬是否足以傳送目前所

*This work has been partially supported by the National Science Council, Taiwan, under Contract 95-2221-E-218-015-MY2

觀看的影片串流傳送給下一個 peer。

儘管 P2P 網路為許多的多媒體串流服務提供者帶來了許多的優勢，例如：大量的運算能力、較低的頻寬與儲存空間成本...等。但是在 P2P 網路帶來這些優勢的同時也產生了許多的議題。一個理想的 P2P 多媒體串流系統應該要能解決以下的議題：

- (1) **客戶端非對稱式的網路連線(Asymmetric Access Link)**。在 P2P 網路上所觀看的多媒體串流品質往往會因為傳送端的上傳頻寬而有直接的影響。因為在 P2P 網路上的使用者來自於不同的連線，例如：Ethernet、DSL、Cable Modem。即便接收端具有充裕的下載頻寬，但是其他的來源節點未必有足夠的上傳頻寬。因此，接收端在接收多媒體串流時，往往會因為傳送端的上傳頻寬過小而無法達到接收端所要求的影片品質。
 - (2) **客戶端間的異質性(Heterogeneity)**。因為在目前的 P2P 網路上存在著許多不同下載頻寬的網路連線，例如：ADSL 或是 Dial-up。在這樣的網路異質性之下，提供固定位元率的單一影片串流並無法滿足所有的 peer。如果 peer 的下載頻寬小於影片串流的播放位元率時，該 peer 將會無法觀看影片。
 - (3) **客戶端的變動性(Dynamics)**。在 P2P 網路上的每一個 peer 可以不受任何的限制自由進出系統。因為節點的行為難以掌握，而且頻繁的變動，這將導致傳送端的離開直接地影響到接收端的多媒體串流接收。所以相較於傳統的主從式架構，在 P2P 網路上提供多媒體串流有著較差的品質穩定性。
 - (4) **系統資源的利用**。因為伺服器端的能力有限，而且每一個 peer 未必能貢獻出充裕的上傳頻寬。一旦伺服器沒有善加規劃的使用這些資源，將會發生過多的資源浪費。這樣的情況將會縮短系統可接受使用者請求的時間。當系統有拒絕節點的情況產生時，該系統的生命將會結束，不會再接受其他的請求。因此伺服器端必須要能夠善加規劃使用網路上的所有資源，增加系統的服務人數，延長系統的生命。
- Layered encoding 為上述的議題提供了一種多版本影片位元率的解決方式[5][20]。在 layered encoding 中所使用的是可調式影片編碼技術。首先一部影片將會被編碼

成多條的影片串流(1~m)，在 layered encoding 中稱之為“layer”。使用者可以依照本身的下載頻寬與需求，同時從多個節點下載所需要的 layer 個數。只要所接收到的 layer 個數愈多，解碼後的影片串流品質就會愈好。

雖然 layered encoding 的編碼技術對於客戶端非對稱式的網路連線與客戶端間的異質性有了新的解決方式。但是 layered encoding 的編碼技術卻存在著串流之間相依性的缺失。不論使用者所需要 layer 個數多少，在接收 layer 時皆必須依序從編號第 1 條的 layer 開始接收，直到使用者所需要的 layer 個數。這樣的缺失將導致編號愈前面的 layer 會有較多的請求，編號愈後面的 layer 有較少的請求。因為這些不同編號的 layer 在網路上不平均的被使用者持有，所以將導致部份的節點形成『熱點』(hot spot)。熱點的產生會造成傳輸工作負擔的不平均，進而影響到整體系統的擴充性。同時，因為串流之間的相依性，當提供編號較低的串流來源節點離開系統時，接收端的影片品質將會大幅地受到影響。

CoopNet[17]與 SplitStream[3][18]對於上述的議題使用了 MDC(Multiple Description Coding)編碼技術[1][12][16][24]來提供另一種的解決方式。MDC 與 layer encoding 同樣都是屬於可調式的影片編碼技術。但是 MDC 與 layer encode 最大的差異在於編碼後的串流之間並不具有相依性。

在 MDC 編碼技術中，一部影片同樣會被編碼成多條的影片串流(1~m)，每一條影片串流稱之為“description”。同樣地，接收端可以依照本身的下載頻寬與需求同時從多個節點下載所想要的 description 個數。但是在經過 MDC 編碼後的 description 中，使用者在接收 description 時並不需要依序從編號第 1 條的串流開始接收。使用者只需要在這 m 條 description 中任意選取能夠滿足本身所需要的 description 個數即可。這樣的特性給予了使用者在選擇來源節點時更加有彈性。

在 CoopNet 與 SplitStream 中，另一項重要的貢獻，就是將客戶端的變動性所產生的影響降到最低。CoopNet 與 SplitStream 主張客戶端所需要的 description 必須來自於不同的來源節點，例如：接收端需要 3 條 description，提供給接收端這 3 條 description 的來源節點必須是 3 個不同的來源節點。這

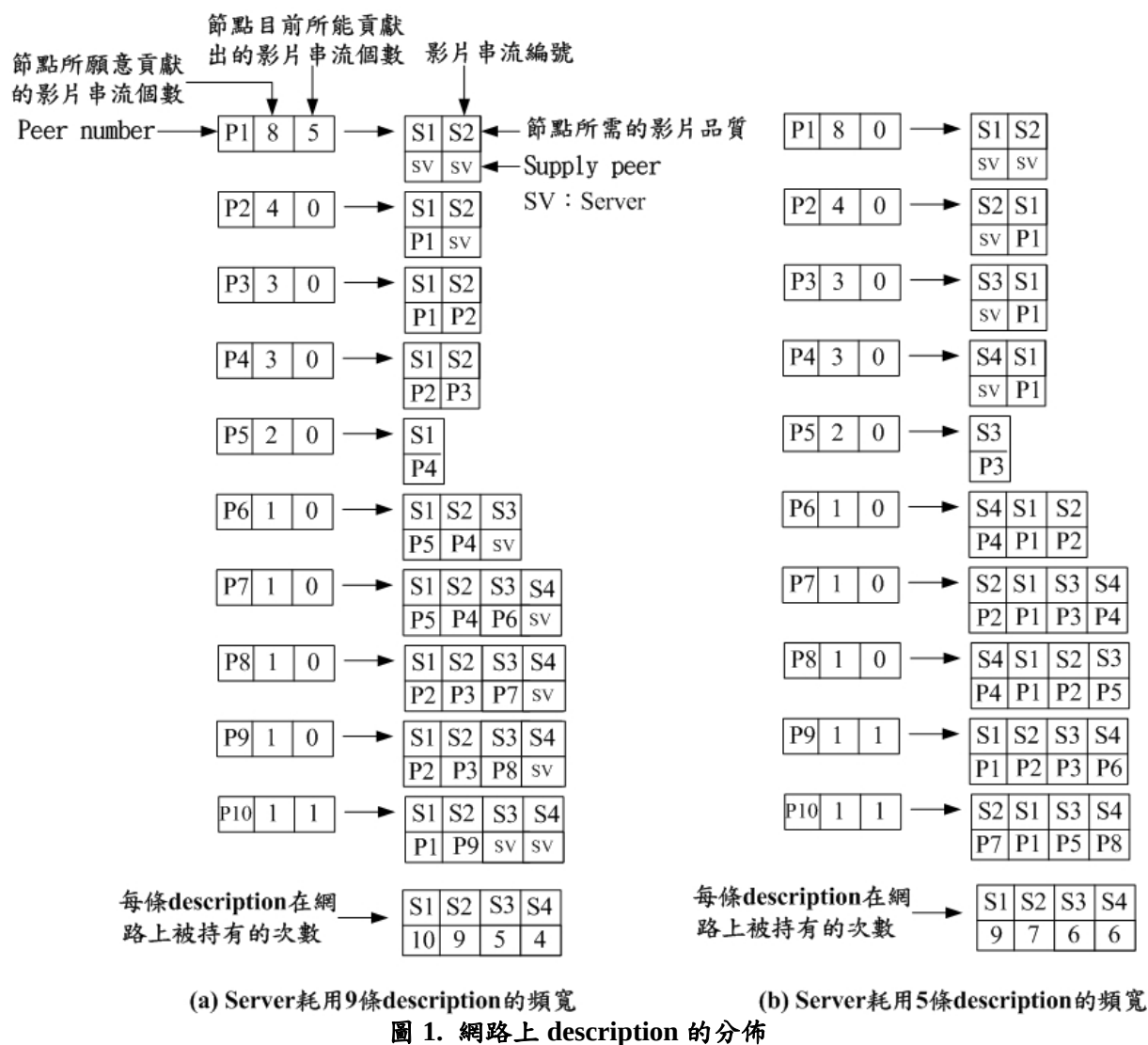


圖 1. 網路上 description 的分佈

種多重來源下載的概念，使得接收端不再因為單一來源節點的離開而導致影片完全無法觀賞。如果接收端的其中一個來源節點離開時，所受到的影響最多為損失1條description的品質。因此，使用這樣多重來源下載的概念可以提升使用者的影片容錯能力，並且提供給使用者更穩定的影片觀看品質。

雖然上述的研究都能夠充份的解決客戶端非對稱式的網路連線、客戶端間的異質性與客戶端間的變動性問題。但是這些研究卻沒有善加的規劃使用網路上的資源，也就是沒有最佳化”系統資源的利用”。在 Layer encoding、CoopNet 和 SplitStream 中，都是使用了可調式的編碼技術進行影片串流的編碼。但是在進行影片串流的分配時，卻忽略了讓網路上的使用者平均的持有這些不同編號的影片串流。在這種缺乏一套可以平均分配每條影片串流的系統中，對於整體系統的擴充性將會有直接的影響，進而增加系統的

拒絕率。當有大量的請求突然湧進系統時，系統會因為伺服器端沒有剩餘的可用頻寬，或是網路上的其他使用者，有剩餘的可用頻寬但是卻因為未持有接收端所需的影片串流，因而拒絕提供服務給新進的使用者。

我們提供了一個例子，如圖 1。在圖 1 中假設一部影片串流使用 MDC 編碼技術編碼成 4 條 description 分別為 S1、S2、S3 與 S4。在網路上存在著 10 位使用者分別為 P1、P2...、P10 且這 10 位使用者符合現今網路的非對稱與異質性網路連線。並且為了希望讓客戶端的變動性影響最小，所以每一個客戶端的所擁有的 description 來源皆是來自於不同的來源節點。

在圖 1.(a)中，每條 description 被使用者所持有的次數是非常不公平的狀況，最多被持有的 S1 與最少被持有的 S4 相差了 6 次。在圖 1.(b)中最多被持有的 S1 與最少被持有的 S4 相差了 3 次。此外，在圖 1.(a)中，在

網路上的全部所有使用者還可以提供出 6 條 description 的頻寬，伺服器耗用了 9 條 description 的頻寬。另外圖 1.(b)中，在網路上全部所有使用者還能夠提供 2 條 description 的頻寬，伺服器耗用了 5 條的 description 的頻寬。這樣的情況充份的顯示出在系統中因為每條 description 被不公平的持有。部份的節點擁有剩餘的對外頻寬，但是未持有其他節點所需的影片內容。因此，在這些節點無法被當成來源節點的情況之下，新進節點只好轉向伺服器請求服務。

有鑒於上述的問題，主因是缺乏了一套良好的影片串流分配機制來提升系統的服務人數與延長系統的生命。因此，本論文將在提供 live streaming 的 P2P 網路上，提供一個有效的影片串流平衡機制，進行影片串流的分配。首先影片串流將會透過 MDC 編碼技術進行編碼。當開始進行 description 分配時，會依據每條 description 的分佈與每個節點的可用頻寬進行分配。藉由平衡每條 description 在網路上的分佈與節點的負擔，達到提升系統整體效能的目標。

本篇論文的編排如下：第 2 節為 CoopNet 的介紹。第 3 節為描述平衡機制所需要的假設及方法。第 4 節為平衡機制的效能評估。最後第 5 節為結論與未來研究。

2. CoopNet

2.1 CoopNet 的系統管理

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“集權式”的管理。首先伺服器端會先將多媒體串流使用 MDC 進行編碼。當多媒體串流被編碼成多條不同編號的 description 之後，伺服器端會將每條 description 放置在不同棵的應用層群播樹中進行群播。當有新的使用者想要取得多媒體串流時，必須先與伺服器取得連繫，並告知本身所需要的 description 數量以及所願意上傳的 description 數量。伺服器端取得該使用的資訊之後，再統一由伺服器配置該使用者所需的 description 以及來源節點。

2.2 CoopNet 的系統特性

CoopNet 的主要特性有以下幾項：

- (1) 充份的適應客戶端間非對稱式的網路連線與異質性。CoopNet 使用了 MDC 編碼技術

進行多媒體串流的編碼，所以接收端可以依照本身的需求同時從多個來源節點接收所需的 description 數量。接收端將可以不再受到傳送端的上傳頻寬而無法最大化本身的下載頻寬。

- (2) 將客戶端的變動性影響降到最低。CoopNet 使用了多重來源下載的概念，所需要的每條 description 的來源節點皆來自於不同的來源節點。當有其中一個來源節點離開時，接收端仍然可以持續觀看影片，所損失的影片品質就只有一條 description 的品質。
- (3) 使用者快速的加入系統與錯誤的回復。因為在 CoopNet 採用的是集權式的管理方式，這樣的管理方式雖然需要依賴一個強而有力的伺服器。但是在使用者的加入與錯誤的回復上卻能夠快速且正確的提供回應給使用者。這種快速且正確的回應使用者需求，在提供多媒體串流的 P2P 網路上是必要的。因為部份的使用者可能會因為長時間的等待而放棄觀賞影片離開系統。

2.3 CoopNet 中選擇來源節點的依據

CoopNet 在來源節點的選擇上提供了兩種不同的選擇依據：

- (1) 隨機選取。當伺服器在配置新進節點的來源節點時，並不考慮任何的其他因素。系統純粹以能夠滿足新進節點的需求為優先考量。並從具有剩餘對外頻寬且持有新進節點所沒有的影片串流內容的來源節點之中隨機挑選。
- (2) 傳輸的距離。伺服器在替新進節點尋找來源節點時，會優先選擇傳輸距離較近的來源節點。這樣的考量主要目的是希望讓傳送端與接收端之間能有較短的傳輸距離。當有較短的傳輸距離，影片串流傳送到達接收端的時間就愈短。如此一來接收端就能愈快的得到影片串流。

在 CoopNet 的這兩項選擇依據中，很明顯的可以看出 CoopNet 並沒有均勻的分配讓每一個使用者持有不同的 description。雖然依照傳輸的距離為首要考量可以讓使用者用最短的時間取得多媒體串流，但是仍然無法提升系統的服務人數與延長系統的生命。

因此，本篇論文提出了一個有效平衡機制來滿足提升系統的服務人數與延長系統生命的需求。在平衡機制中將依照每條 description 的分佈與節點在網路上的資源多寡進行

description 的分配，讓每條 description 可以被使用者均勻的持有。雖然不以傳輸的距離為首要的考量，可能會出現較長的傳輸距離，而延遲接收端接收到影片串流的時間。但是這樣的延遲通常只會出現於新進使用者加入系統的初始時段，對於接收端所需要的品質並不會有所影響。

3. 平衡機制

3.1 符號與定義

在開始介紹平衡機制的假設與方法之前，首先必須先介紹在論文中所會使用到的符號表示與其代表的意義：

R_x ：表示編號第 x 條的 description 目前在網路上所剩餘的可用頻寬。例如： $R_1=4$ ，則表示編號 1 這條 description 剩餘 4 條 description 的頻寬可用。

O_x ：表示節點持有編號第 x 條 description 的來源節點。例如： $O_1=3$ ，則表示該節點擁有編號 1 這條 description 來自於節點 3。

M_j ：表示節點 j 所必須貢獻的 description 編號。例如： $M_2=1$ ，則表示節點 2 在系統中必須貢獻編號 1 這條 description。

A_j ：表示節點 j 目前所剩餘的對外頻寬。例如： $A_1=4$ ，則表示節點 1 目前剩餘 4 條 description 的可用對外頻寬。

Q_j ：表示節點 j 所需要的 description 個數。例如： $Q_1=4$ ，節點 1 需要 4 條 description 的影片品質。

S ：表示所有 description 編號的集合。 $S=\{x | x=1,2,...,m\}$

H_j ：表示節點 j 目前所持有的 description 編號的集合。

N_j ：表示節點 j 目前未持有的 description 編號的集合， $N_j=S-H_j$ 。

c ：表示伺服器選中新進節點所必須貢獻的 description 編號。

d ：表示伺服器選中要滿足新進節點所需品質的 description 編號。

在本論文中，對於影片串流配置和錯誤回復的程序，我們採用與 CoopNet 相同的中央「集權式」管理。當新進的使用者需要影片串流服務時，首先該使用者會先與伺服器取得連繫，然後在由伺服器進行影片串流的分配。在平衡機制中，為了要達到使用者所持有的每條 description 皆來自不同的來源節點。因此，系統會限制每一個使用者在系統中所有的對外頻寬將只會貢獻同一編號的 description(M_j)。這

樣的限制主要的目的是要讓客戶端的變動性影響降到最低。

3.2 決定新進節點所必須貢獻的 description 與來源節點

3.2.1 決定新進節點所必須貢獻的 description

當使用者與伺服器取得連繫的同時，使用者會告知伺服器該使用者所需要的 description 數量(Q_{k+1})，以及所願意貢獻出的 description 數量(A_{k+1})。伺服器取得該資訊後，伺服器將會開始進行 description 的分配。首先，伺服器會加總每一條 description 在網路所剩餘的可用頻寬。

Step1：加總每一條 description 在網路上所剩餘的資源

$$R_x = \sum_{j: M_j=x} A_j, x=1,2,...,m \quad (1)$$

假設一部影片串流被編碼成 4 條 description，分別為 S1、S2、S3 與 S4。同時在網路上目前存在著 4 個節點分別為 P1、P2、P3 與 P4。這 4 個節點分別貢獻不同的 description，如圖 2。當新進節點 P5 加入系統時，伺服器會加總每一條 description 所剩餘的可用頻寬，如圖 2 所示。S1 剩餘 4 條可用頻寬、S2 剩餘 5 條可用頻寬、S3 剩餘 2 條可用頻寬，最後 S4 剩餘 1 條可用頻寬。

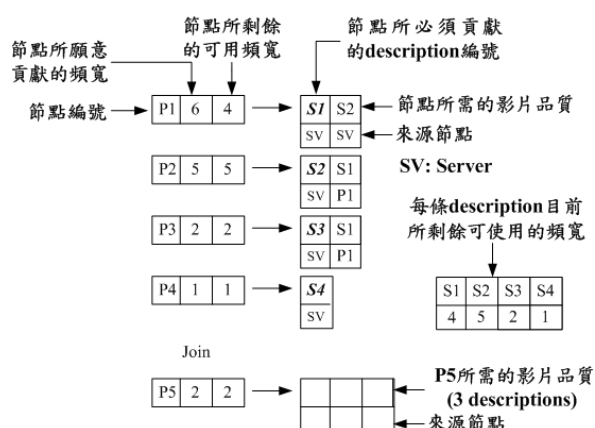


圖 2. 加總所有 description 的可用頻寬

伺服器加總完每一條 description 在網路上所剩餘的可用頻寬後，伺服器會挑選出新進節點所必須貢獻的 description。伺服器在挑選該條 description 時，所依照的準則為剩餘可用頻寬最少的 description。

Step2：挑選新進節點在系統所必須貢獻的 description(c)

$$c: R_c = \min_{1 \leq i \leq m} R_i \quad (2)$$

這樣的準則可以確保剩餘最少可用頻寬的 description，不會因為新進節點的加入，而導致該條 description 在網路上剩餘可用的頻寬減少。另一方面是為了要平衡每條 description 在網路上的可用頻寬。因為當有部份的 description 在網路上沒有剩餘的可用頻寬時，這表示後來加入的節點將有可能無法全部的從網路上的其他 peer 獲得影片串流。當發生這樣的情況時，新進的節點只能夠轉向請求伺服器提供服務。過多的 description 來源都必須仰賴於伺服器時，將會增加伺服器的負擔並且降低系統的服務人數。

決定新進節點所要貢獻的 description 之後，新進的節點將會把所有的對外頻寬拿來貢獻伺服器為該節點選擇的 description。假設新進節點 P5 需要 3 條 description 的品質，並且願意貢獻 2 條 description 的頻寬，如圖 3 所示。伺服器在加總完每一條 description 的可用頻寬後，將會選擇剩餘可用頻寬最少的 description S4。S4 這條 description 將會成為 P5 在系統中所必須貢獻的 description。

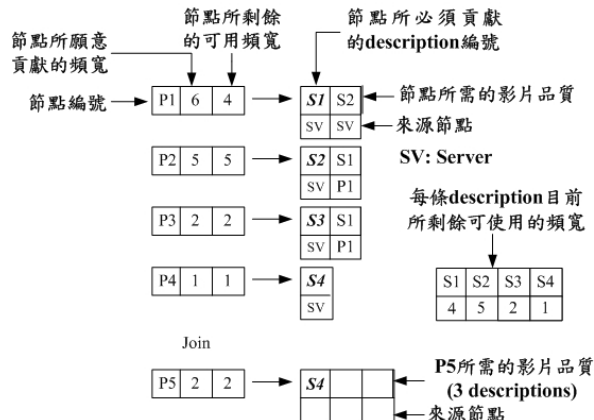


圖 3.新節點必須貢獻的 description

3.2.2 決定所必須貢獻 description 的來源節點

伺服器選定新進節點所必須貢獻的 description 之後，接下來必須決定所選定 description 的來源節點(O_c)。伺服器在選擇該 description 的來源節點時，會選擇剩餘可用頻寬最多，而且是負責貢獻該條 description 的節點來當作新進節點的來源節點。

Step3: 挑選新進節點所必須貢獻 description 的來源節點(O_c)

$$O_c = j, \text{ where } A_j = \max_{i: M_i = c} A_i \quad (3)$$

如果所選中的 description 在網路上沒有剩餘的頻寬可用時，則由伺服器負責提供。若伺服器頻寬已耗盡時，該新進節點將會被拒絕。依照這樣的選擇依據，主要的目的是要平衡網路上節點的傳輸負擔，避免部份的節點接受過多的請求而形成「熱點」。如果當部份的節點形成熱點時，該熱點離開系統將會影響到較多的節點的串流品質。

如圖 4 所示，伺服器將會為 P5 所要貢獻的 description S4 挑選來源節點。伺服器為 P5 所挑中的來源節點為 P4，因為 P4 是負責貢獻 S4 且剩餘頻寬最多。圖 5 為一個新進節點加入系統時，伺服器為尋找新進節點所必須貢獻的 description 與來源節點的演算法。

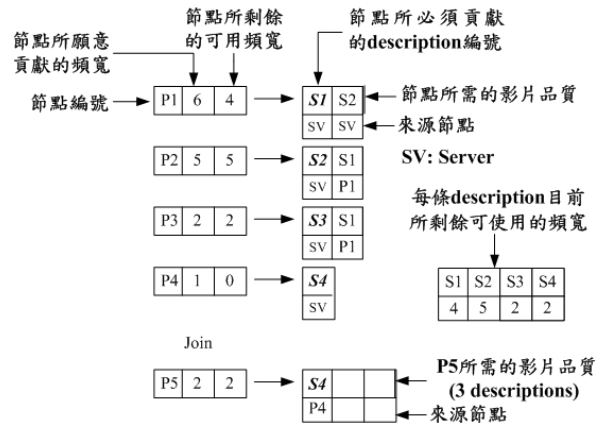


圖 4.必須貢獻 description 的來源節點

For a newly arrived peer $k+1$

1. **For** each description $x \leftarrow 1$ to m
2. $R_x \leftarrow \sum_{j: M_j = x} A_j$
3. Find a description c subject to $R_c = \min_{1 \leq i \leq m} R_i$
4. $M_{k+1} \leftarrow c$
5. **If** $R_c = 0$ **Then**
6. **If** server bandwidth > 0 **Then**
7. $O_M \leftarrow \text{server}$
8. $Q_{k+1} \leftarrow Q_{k+1} - 1$
9. $H_{k+1} \leftarrow H_{k+1} \cup c$
10. **Else** reject request
11. **Else**
12. Find a peer j subject to $A_j = \max_{i: M_i = c} A_i$
13. $O_c \leftarrow j$
14. $Q_{k+1} \leftarrow Q_{k+1} - 1$
15. $H_{k+1} \leftarrow H_{k+1} \cup c$

圖 5.尋找必須貢獻的 description 與來源節點演算法

3.3 決定其他 description 與來源節點

伺服器選定完新進節點在系統所必須貢獻的 description 與來源節點之後。接下來伺服器必須選擇其他不同編號的 description(d) 來滿足新進節點所需的 description 個數。

3.3.1 尋找其他不重覆的 description

伺服器在選擇其他不同編號的 description 時，會優先選擇在網路上剩餘可用頻寬最多，而且是新進節點所未持有的 description(H_{k+1})。

Step4: 挑選其他不重覆的 description(d)

$$N_{k+1} = S - H_{k+1} \quad (4)$$

$$d: R_d = \text{MAX}_{i \in N_{k+1}} R_i \quad (5)$$

選擇剩餘最多可用頻寬的 description 的目的，主要是為了不讓剩餘可用頻寬較少的 description 在下一個節點加入時變為 0。因為當有部份的 description 所剩餘的可用頻寬剩餘 0 時，下一個節點將有可能必須向伺服器要求服務。如此一來，將會導致伺服器有過多的請求，進而影響系統的服務人數。另外，選擇其他不同編號 description 主要的原因是因為在 MDC 的編碼之下，接收到相同的 description 並不會提升串流品質。所以必須選擇不重覆的 description。

在圖 4 中，節點 P5 未持有的 description 編號為 {1,2,3}。S2 在這 3 條 description 所剩餘可用的頻寬為最多。因此，伺服器將會優先選擇 S2 來滿足新進節點所需的串流品質。圖 6 為伺服器為 P5 選定 S2 之後的情況。

3.3.2 尋找其他 description 的來源節點

同樣地，在選擇該條 description 之後，伺服器仍然必須為新進節點選擇該條 description 的來源節點(O_d)。其所依照的準則與 Step3 相同。

Step5: 挑選其他不重覆 description 的來源節點

$$O_d = j, \text{ where } A_j = \text{MAX}_{i: M_i=d} A_i \quad (6)$$

接下來伺服器將重覆 Step4 與 Step5，直到新進節點所需的 description 個數被滿足為止。一旦系統無法滿足新進節點所需的影片

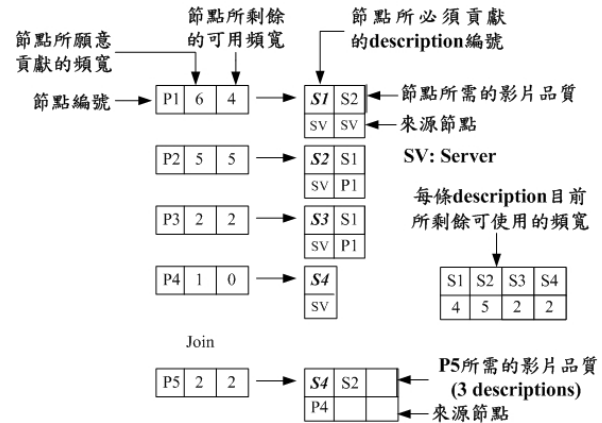


圖 6. 挑選不重覆且剩餘最多資源的 description

品質時，系統將會拒絕該新進節點。如圖 7 所示，伺服器選定 P5 所要接收的 description S2 之後，必須選擇該條 description 的來源節點。選定的來源節點為 P2，因為 P2 負責貢獻 S2，而且所剩餘的頻寬是最多的。圖 8 所表示的是節點 P5 加入系統之後，網路上所有節點與所有 description 的情況。圖 9 為伺服器為新進節點選定所必須貢獻的 description 與來源節點之後，選擇其他不重覆的 description 與來源節點的詳細演算法。

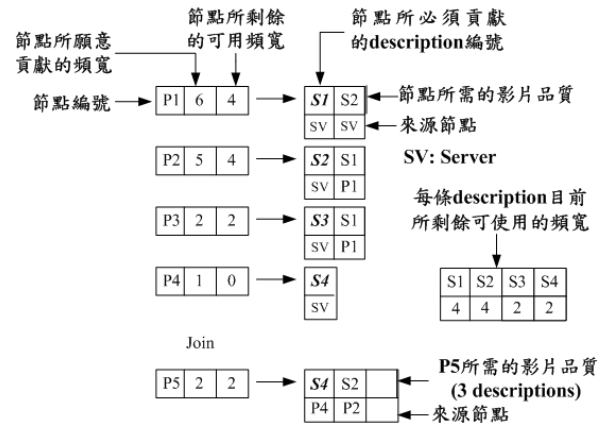
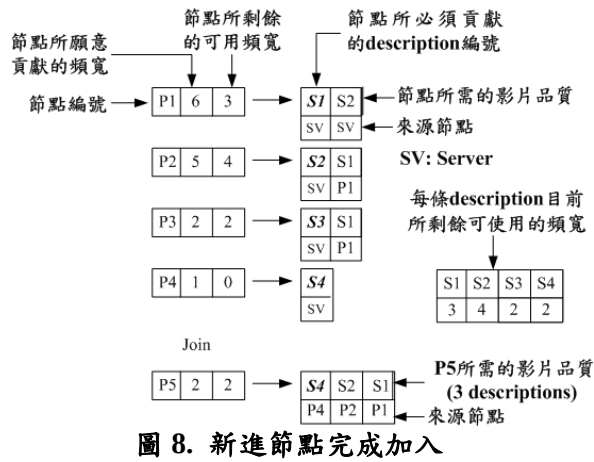


圖 7. 挑選不重覆 description 的來源節點

3.4 錯誤回復(Failure recovery)

在本研究中，將錯誤的復原分成兩類：

- (1) **被拋棄節點所必須貢獻 description 的來源節點。**當發生節點所必須貢獻 description 的來源節點(O_c)離開時，系統將必須為被拋棄的節點進行錯誤回復。因為離開的節點是負責提供被拋棄節點所必須貢獻的 description。所以當被拋棄的節點失去該條 description，同



For a newly arrived peer $k+1$

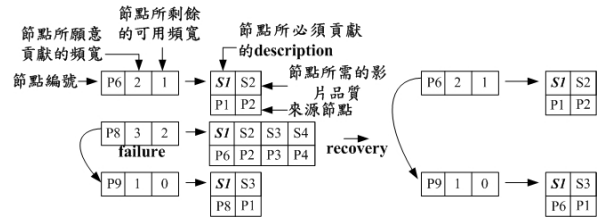
1. Repeat
2. $N_{k+1} \leftarrow S - H_{k+1}$
3. Find a description d subject to $R_d = \text{MAX}_{i \in N_{k+1}} R_i$
4. If $R_d = 0$ Then
5. If server bandwidth > 0 Then
6. $O_d \leftarrow \text{Server}$
7. $Q_{k+1} \leftarrow Q_{k+1} - 1$
8. $H_{k+1} \leftarrow H_{k+1} \cup d$
9. Else reject request
10. Else
11. Find a peer j subject to $A_j = \text{MAX}_{i: M_i = d} A_i$
12. $O_d \leftarrow j$
13. $Q_{k+1} \leftarrow Q_{k+1} - 1$
14. $H_{k+1} \leftarrow H_{k+1} \cup d$
15. Until $Q_{k+1} = 0$

圖 9. 尋找其他 description 與來源節點演算法

時也會影響到該節點的子孫節點。為了不希望讓整個系統因為部份節點的離開，而引起大範圍的變動。所以當節點失去所必須貢獻的 description 時，伺服器會執行 Step3 來為被拋棄的節點尋找新的來源節點。伺服器所尋找到的新來源節點會比被拋棄節點早先加入系統，而且是貢獻相同 description 的節點。

圖 10 說明了伺服器進行節點所必須貢獻 description 的來源節點錯誤回復後的情況。原本在系統中是由節點 P8 負責提供 S1 給節點 P9，但是節點 P8 在傳輸中途離開系統。因此，伺服器將回到 Step3 尋找其他替代節點來為節點 P9 進行錯誤回復。伺服器所尋找到的節點為 P6，同時該節點會滿足 Step3 所要求的條件。

(2) 非被拋棄節點所必須貢獻 description 的來源節點。當節點發生非必須貢獻 description



的來源節點(O_d)離開時。伺服器會執行 Step4 與 Step5 進行錯誤回復，當進行錯誤回復時，所選擇的 description 並不需要符合節點所失去的 description。進行錯誤回復的主要目的是為了讓被拋棄的節點回復到原先所持有的串流品質。所以並不需要考慮所回復的 description 內容是否與未被拋棄時的一樣。只需要考慮是否能夠回復原有的品質。

圖 11 說明了伺服器進行非節點所必須貢獻 description 的來源節點錯誤回復後的情況。節點 P1 的離開將導致節點 P2、P3 與 P5 受到影響。因此，伺服器將執行 Step4 與 Step5 進行錯誤回復。進行錯誤回復時，所選擇的 description 不需要與原先所持有的 description 一致。例如，節點 P5 原先持有 S1、S2 與 S4，但是在錯誤復原之後，節點 P5 所持有的是 S2、S3 與 S4。

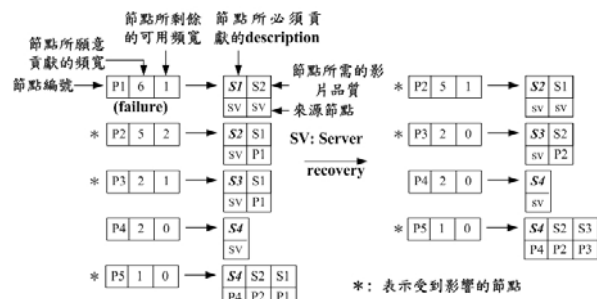


圖 11. 非必須貢獻 description 的錯誤回復

4 效能評估

4.1 實驗環境與設定

本篇論文的實驗參數設定如表 1 所示。使用者的請求會持續 60 秒，伺服器的頻寬設定為能夠提供 1000 條的 description 頻寬。一部影片串流將使用 MDC 編碼技術編碼成 16 條品質相等的 description。每秒鐘使用者的到達個數分別為 30、60、90、120、150、和 180 個節點。每一個節點所要接收的 description 會介於 2 到 16 條 description，剩餘可用的對外頻寬會介於

0~16 條 description。

每秒鐘到達的節點所要求 description 個數比例如表 2 所示。為了符合現今的網路使用者習慣與異質性的網路連線。所以，我們將大部份的節點所要求的 description 個數集中在 4~12 條 description。而在要求較少與較多 description 的請求會有比較少的使用者。如表 2 中，2 條與 16 條 description 的請求。同時，在我們所設定的環境中，每一個節點的上傳頻寬會等於下載頻寬。

表 1 實驗參數設定

模擬時間(秒)	60
伺服器頻寬(條)	1000
Description 個數(條)	16
每秒鐘節點到達個數	30,60,90,120,150,180
Q_i	2~16
A_i	0~16

表 2 每秒鐘節點要求 description 個數的比例

Description 個數	2	4	6	8	10	12	14	16
下載比例 (%)	6	13	17	17	17	17	7	6

在實驗過程中，我們將與其他兩種方法進行比較。這兩種方法同樣以 CoopNet 為基礎，每條 description 皆來自於不同的來源節點。分別為以下兩種：

- (1) **random**: random 的方式在選擇 description 與來源節點時，採用的是隨機選擇的方式並不會有任何的依據。
- (2) **sequence**: sequence 的方法中，description 的傳送會依照 description 的編號依序從 1 傳送到 Q_{k+1} 。

4.2 實驗結果與分析

以下我們將要針對本論文所提出的平衡機制，以及 random 和 sequence 這三種方法進行效能評估。評估的指標分別有，每條 description 在網路上分佈的標準差、伺服器端的使用率、系統的拒絕率和系統生命。伺服器端的使用率的計算方式為伺服器端所消耗的頻寬除以伺服器端的全部頻寬。標準差的公式為：

$$\sigma = \sqrt{\frac{1}{m} \sum (x - u)^2} \quad (7)$$

m 表示影片被編碼後的 description 數量， x 表示每條 description 實際在所有被使用頻寬中使用的次數， u 表示每條 description 的平均數。

圖 12 說明了三種不同方法的標準差。在 sequence 的方法中，因為每條 description 的分配是依照順序性的傳送，所以編號較高的 description 會有較少的請求，相反地，編號較低的 description 會有較多的請求。因此，sequence 標準差的差異會隨著節點的增加而快速增加。在 random 的方法中，因為採用隨機的分配，每條 description 未必會被均勻的分配。因此，隨著節點的增加，標準差將會小幅度的增加。在我們的平衡機制中，進行 description 的分配時，因為有考量到平衡每條 description 在網路上的資源，所以標準差會是最低的。

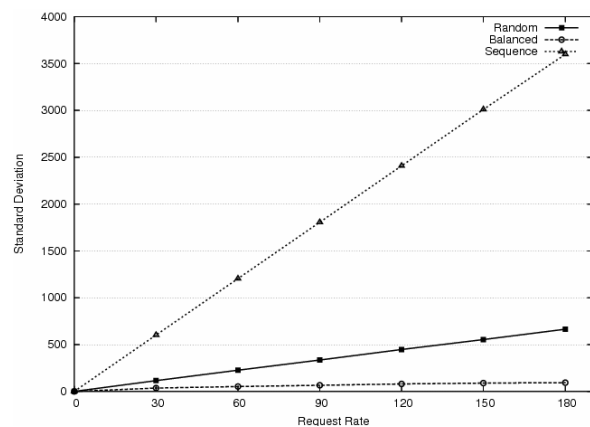


圖 12. 每條 description 的標準差, σ

圖 13 說明三種不同方法的伺服器頻寬使用率。當節點的到達率增加時，三種方法的伺服器頻寬使用率都會增加。但是在 sequence 的方法，因為每條 description 的分配極度不均勻產生過高的標準差。當要求較多 description 的節點加入時，這些節點只能夠從目前網路上的 peer 獲得編號較低的 description，其餘編號較高的 description 將必須從伺服器端來取得。所以 sequence 的方式會有最高的伺服器使用率。而在 random 的情況之下，部份的 description 也有可能產生不均勻分佈的情況，所以當有節點要求較多的 description 時，也將必須由伺服器端來支援。

在我們的平衡方法中，因為考慮到每條 description 在網路上的資源平衡，所以當有要求較多 description 的節點加入時，將可以從網路上的 peer 取得，不需要再向伺服器端請求。所以我們的方法在三種方法中，伺服器端的使用率會是最低的。

經由上述的兩項的效能評估後，我們可以發現 description 的均勻分配將可以更進一步的節省伺服器端的頻寬。接下來我們將評估 description 的均勻分配對系統的拒絕率以及系統生命的影響。

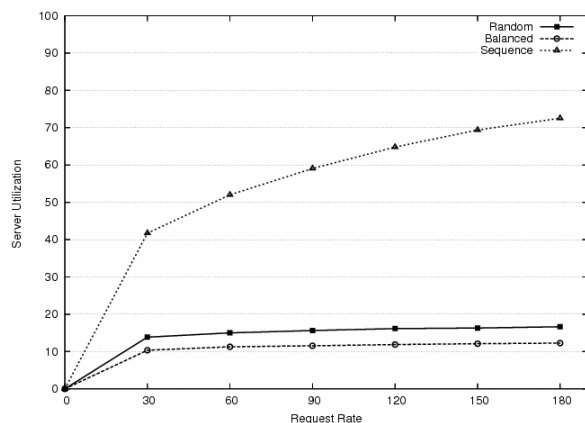


圖 13. 伺服器頻寬使用率

首先我們將伺服器端的對外頻寬設定為 120 條的 description 頻寬，節點的請求同樣持續 60 秒鐘。在圖 14 中，我們可以發現三種方法的拒絕率都會因為節點的請求率增加而升高。在 sequence 的方式中，因為過高的標準差，將導致伺服器的頻寬快速的耗盡，所以拒絕率將會大幅地升高。在 random 的方式中，因為採用隨機的挑選方式，所以 description 的標準差將會比 sequence 的方式低。相對地，標準差愈低將有更少的節點向伺服器端請求，所以拒絕率將比 sequence 低。在我們的平衡機制中，因為精準的控制每條 description 的資源，所以有著最低的標準差。因此，本論文所提出的平衡機制會有最低的拒絕率。

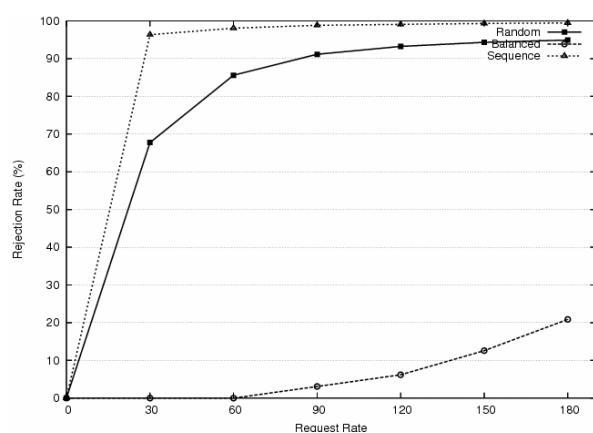


圖 14. 系統整體拒絕率之比較

圖 15 說明了在伺服器的頻寬為 120 條 description，以及節點的請求時間最多為 60 秒

的情況下，系統產生第一個拒絕節點的時間。sequence 與 random 同樣會因為過高的標準差，而導致伺服器的頻寬快速耗盡，所以會有較短的系統生命。在我們的平衡機制中，大部份的節點將可以從網路上的 peer 自給自足，不需要向伺服器端請求。所以能夠服務更多的客和延續較的系統生命。

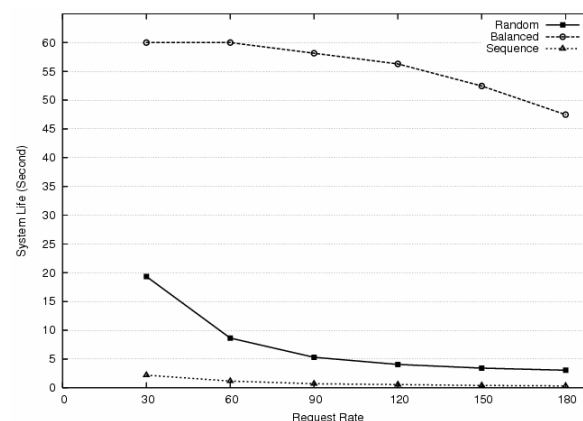


圖 15. 系統生命時間之比較

5 結論

在本篇論文中，我們首先藉由 MDC 編碼技術解決了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，我們更進一步的對系統中的資源善加利用。在本篇論文所提出的方法中，主要是透過平衡每條 description 在網路上的資源與每一個節點的負擔。當伺服器在進行 description 的分配時，將會依據 description 與節點所剩餘資源的多寡進行分配。

經由模擬實驗的結果可以得知，在目前網路上的使用者習慣與異質性的網路連線中，本篇論文所提供的方法相較於其他兩項方法，將可獲得以下的優勢：

- (1) **更低的伺服器頻寬耗用。**藉由本論文所提供的平衡機制進行 description 的分配，將可以讓每條 description 均勻的分散在網路上。如此一來，節點之間將能夠自給自足，而不需要求助於伺服器。所以對於伺服器頻寬將會有較低的請求。
- (2) **接受更多的使用者。**由於本論文所提出的平衡機制可以更進一步的節省伺服器頻寬的耗用。當三種方法的伺服器頻寬相等時，我們所提供的方法將可以接受更多的使用者。
- (3) **更長時間的系統生命。**在我們所提供的平衡機制中，因為伺服器頻寬的耗用的速度

較其他兩種方法慢，所以將能夠維持更長時間的系統生命。

在未來研究方面，我們將針對當系統無法給予節點要求的 description 數量時，每一個節點對於不同影片品質的容忍程度進行探討。當系統請求人數增多，且網路上的資源耗盡時。將對影片品質容忍程度較大的節點減少 description 的供應，以接受新進節點的請求，藉此滿足每一個節點的最低需求，並且降低系統的拒絕率。

參考文獻

- [1] Apostolopoulos, J. G. and Wee, S. J., "Unbalanced multiple description video communication using path diversity," *Proc. of IEEE ICIP*, Vol. 1, pp.966-969, 2001.
- [2] Banerjee, S., Bhattacharjee, B. and Kommareddy, C., "Scalable application layer multicast," *Proc. of ACM SIGCOMM*, pp.205-217, 2002.
- [3] Castro, M., Druschel, P., Kermarrec, A., Nandi, A., Rowstron, A. and Singh, A., "SplitStream: High-bandwidth multicast in cooperative environments," *Proc. of ACM SOSP*, pp.298-313, 2003.
- [4] Cui, Y., Li, B. and Nahrstedt, K., "oStream: Asynchronous streaming multicast in application-layer overlay networks," *IEEE Journal on Selected Areas in Communications(JSAC)*, pp.91-106, 2004.
- [5] Cui, Y. and Nahrstedt, K., "Layered peer-to-peer streaming," *Proc. of ACM NOSSDAV*, pp.162-171, 2003.
- [6] Chu, Y-H., Rao, S. G., Seshan, S. and Zhang, H., "Enabling conferencing applications on the internet using an overlay multicast architecture," *Proc. of ACM SIGCOMM*, pp.55-67, 2001.
- [7] Clarke, I., Sandberg, O., Wiley, B. and Hong, T. W., "Freenet: A distributed anonymous information storage and retrieval system," *Workshop on Design Issues in Anonymity and Unobservability*, pp. 311-320, July 2000.
- [8] Deshpande, H., Bawa, M. and Garcia-Molina, H., "Streaming live media over a peer-to-peer network," in *Submitted for publication*, 2002.
- [9] Do, T. T., Hua, K. A., and Tantaoui, M. A., "P2VoD: Providing fault tolerant video-on-demand streaming in peer-to-peer environment," *Proc. of IEEE ICC*, Vol. 3, pp.1467-1472, 2004.
- [10] Dan, A., Sitaram, D. and Shahabuddin, P., "Scheduling policies for an on-demand video server with batching," *Proc. of ACM MM*, pp.15-23, 1994.
- [11] Guo, Y., Suh, K., Kurose, J. and Towsley, D., "P2Cast: Peer-to-Peer patching scheme for VoD service," *Proc. of ACM WWW*, pp.301-309, 2003.
- [12] Goyal, V. K., "Multiple description coding: compression meets the network," *Signal Processing Magazine of IEEE*, Vol. 18, pp.74-93, 2001.
- [13] Hua, K. A., Cai, Y. and Sheu, S., "Patching: a multicast technique for true video-on-demand services," *Proc. of ACM MM*, pp.191-200, 1998.
- [14] Jin, S. and Bestavros, A., "Cache-and-relay streaming media delivery for asynchronous clients," In *International Workshop on Networked Group Communication(NGC)*, 2002.
- [15] Juhn, L. and Tseng, L., "Harmonic broadcasting for video-on-demand service," *IEEE Transactions on Broadcasting*, pp. 268-271, 1997.
- [16] Lu, Z. and Pearlman, W. A., "An efficient, low-complexity audio coder delivering multiple levels of quality for interactive application," *Proc. of IEEE Multimedia Signal*, pp. 529-534, 1998.
- [17] Padmanabhan, V. N., Wang, H. J., Chou, P. A. and Sripanidkulchai, K., "Distributing streaming media content using cooperative networking," *Proc. of ACM NOSSDAV*, pp.177-186, 2002.
- [18] Padmanabhan, V. N., Wang, H. J. and Chou, P. A., "Resilient peer-to-peer streaming," *Proc. of IEEE ICNP*, pp.16-27, 2003.
- [19] Ripeanu, M., "Peer-to-Peer architecture case study: Gnutella network," *Proc. of IEEE P2P*, pp. 99-100, 2001.
- [20] Rejaie, R., Handley, M. and Estrin, D., "Layered quality adaptation for Internet video streaming," *IEEE Journal on Selected Areas in Communications(JSAC)*, Vol. 18 pp.2530 - 2543, 2000.
- [21] Saroiu, S., Gummadi, P. K. and Gribble, S. D., "A measurement study of peer-to-peer file sharing systems," *Multimedia Computing and Networking*, 2002.
- [22] Tran, D. A., Hua, K. A. and Do, T., "ZIGZAG: An efficient peer-to-peer

- scheme for media streaming,” **Proc. of IEEE INFOCOM**, Vol. 2, pp.1283-1292, 2003.
- [23] Viswanathan, S. and Imielinski, T., “Metropolitan area video-on-demand service using pyramid broadcasting,” **ACM Multimedia Systems Journal**, Vol. 4, No. 3, pp.197-208, 1996.
- [24] Wand, Y., Reibman, A. R. and Lin, S., “Multiple description coding for video delivery,” **Proc. of the IEEE**, pp.57-70, 2005.
- [25] Zeng, M., Yu, H., Yang, F. and Zheng, W., “Dynamically optimize multicast tree structure,” **Proc. of IEEE GCCW**, pp. 143-150, 2006.