

利用二階分割法於多通道無線環境中  
廣播排程之問題

Using Two-Way Partition Method for Multi-Channel  
Broadcasting in Mobile Environment

研 究 生：張 元 安

Student :Yuan-Ann Chang

指 導 教 授：吳 光 閔

Advisor : Dr. Guang-Ming Wu.



南 華 大 學  
資 訊 管 理 學 系(所)  
碩 士 論 文

A Thesis

Submitted to Department of Information Management  
College of Management  
Nan-Hua University

in partial Fulfillment of the Requirements

for the Degree of  
Master of Business Administrator

in  
Information Management

June 2005

Chaiyi Taiwan, Republic of China.

中華民國 九十四年 六月

# 南 華 大 學

資訊管理研究所

碩 士 學 位 論 文

利用二階分割法於多通道無線環境中廣播排程之問題

研究生：張元安

經考試合格特此證明

口試委員：收吟秋  
邱宏林  
吳光國

指導教授：吳光國

系主任(所長)：

口試日期：中華民國 九十四 年 六 月 十七 日

## 誌 謝

論文的完成要感謝很多人，尤其是指導教授 吳光閔老師，感謝老師對於論文的指導，引導我在研究的路上有明確的方向，使得論文能夠如期完成，感念師恩。

感謝 邱宏彬老師，江明朝老師於百忙之中審閱我的論文，並於口考期間給予提供寶貴的修改意見，使得論文更加完整，元安感恩在心中。

在南華求學階段，有幸受業於 邱宏彬老師、邱英華老師、周志賢老師。等諸位老師。老師於課堂教學認真，對於對待人處事更是重視，使得元安受益良多，感謝老師的諄諄教悔。

於求學階段，感謝家母的支持，使我可以無後顧之憂在校求學。感謝同學的陪伴，使得求學生活多采多姿。更感謝摯友啟源、子軒、妙謙在我低潮時的支持與鼓勵。僅以此文，獻上個人由衷的感謝與敬意，並將這份榮耀與所有關心我的人一起分享。

張元安 謹識

於南華大學 成均館

民國九十四年六月

# 利用二階分割法於多通道無線環境中廣播排程之問題

學生：張元安

指導教授：吳光閔 博士

南華大學資訊管理學系碩士班

## 摘 要

無線技術的快速發展，行動用戶端隨時隨地的存取想要的資料。在無線環境之中，由於行動裝置頻寬的限制，所以廣播是一個用來節省頻寬的技術。在廣播通道的數量上，有單一通道與多通道之分別，而在多通道的廣播環境中，廣播資料可以平均分配到數個廣播通道中，如此可以減短廣播的週期，來減少用戶端存取資料的時間。當一個用戶端所請求的資料集合中，若資料集合被分配到不同的通道時，將可能發生存取的資料在相同時間廣播，這樣的情況稱為資料競爭。當發生此情形時，用戶端必須等待到下一個週期，如此便增加用戶端存取資料的時間。本文提出資料分割演算法應用在多通道上的資料分割，利用資料分割的結果，來減少在廣播排程上，用戶端請求資料所花費的廣播週期次數。最後針對我們所提出的方法進行效能的評估：比較初始化與使用本文所提出之演算法對於競爭次數的影響。實驗結果顯示本文所提出的演算法可以有效地減少資料競爭的發生。這樣的結果預期對用戶端在存取資料時可以有效地減少廣播週期的次數。

關鍵字：無線廣播、單一通道、多通道、資料分割

# **Using Two-Way Partition Method for Multi-Channel Broadcasting in Mobile Environment**

Student : Yuan Ann Chang

Advisors : Dr. Guang-Ming Wu

Department of Information Management  
The M.B.A. Program  
Nan-Hua University

## **ABSTRACT**

Owing to the advance of wireless communication techniques, a mobile user can access the desired data anytime anywhere. Due to the limit of mobile device, data broadcasting is an efficient method for save bandwidth. In the multi-channel, the data can be equally distributed to different channels for reduce the broadcast cycle and reduce the client which access data item. When a mobile user submits an issue for a data set that it distributed to different broadcast channels, the state called data competition. When the numbers of data competition increase, the data collision increase also, therefore, the mobile user must wait for next broadcast cycle, and thus the mobile user needs more time to access data set. The state called data collision. In this article, we propose the Two-Way Partition Algorithm and apply it to data partition of multi-channel. We use the result of data partition to reduce the data competition. Finally, we evaluate the performance of the algorithm and compare the data competition times that influence with initial states' and using our algorithm. The result shows that our approach significantly reduces the data competition times.

Keyword: Wireless Broadcast, Two-Way Partition, Single-Channel 、  
Multi-Channel

# 目錄

書名頁	ii
論文指導教授推薦函	iii
論文口試合格證明	iv
誌謝	v
中文摘要	vi
英文摘要	vii
目錄	viii
表目錄	xi
圖目錄	xii
第一章、緒論	1
第一節研究背景	1
第二節動機	2
第三節研究目的	3
第四節研究架構	4
第二章、文獻探討	5
第一節資料傳輸模式	7
第二節廣播環境介紹	11
一、廣播時效性	11
二、廣播通道數量種類	12
三、用戶端請求資料項之數量	15
四、資料範圍	16

五、廣播資料重覆性	16
六、廣播排程策略	17
1. First Come First Served	17
2. Earliest First Request (EFR)	17
3. Most Request First (MRF)	18
4. Longest Wait First (LWF)	18
5. R*W	18
6. QEM	18
七、提昇用戶端效能的相關技術	19
1 索引技術(Indexing)	19
1.1(1, m) 索引技術	20
1.2 分散式索引技術	21
2 快取技術(Caching)	22
八、效能評估	23
第三章、問題描述	25
第一節相關環境	25
第二節符號定義與說明	27
第三節資料競爭之範例	29
第四節演算法目標	33
第四章、二階分割演算法	35
第一節基本想法	35
第二節基本假設	36
第三節二階分割演算法	37
第五章、實驗結果	49

第一節實驗環境說明	49
第二節模擬參數	49
第三節均勻分佈	50
一、廣播資料項數量對競爭次數之影響	50
二、選擇率對競爭次數之影響	51
第四節常態分佈	53
一、資料項數量對競爭次數之影響	53
二、選擇率對競爭次數之影響	54
第六章、結論與未來方向	56
參考文獻	57



## 表 目 錄

表3.1:問題描述使用之相關符號	27
表4.1:演算法相關符號	37
表4.2:演算法中相關變數	39
表5.1:系統模擬參數	49
表5.2:均勻分佈下:資料項數量對競爭次數之影響	51
表5.3:均勻分佈下:選擇率對競爭次數之影響	52

## 圖目錄

圖1.1:無線網路架構圖	1
圖2.1:Ad hoc基本運作示意圖	6
圖2.2:需求模式之廣播架構圖	7
圖2.3:廣播架構示意圖	9
圖2.4:單一廣播通道之結果	13
圖2.5:多通道廣播排程結果:各個通道資料平坦式分配	14
圖2.6:通道廣播排程結果:各個通道資料非平坦式分配	15
圖2.7:相同廣播週期內使用Flat廣播模式	17
圖2.8:相同廣播週期內使用Non-Flat廣播模式	17
圖2.9:(1, m)索引技術之索引樹	21
圖2.10:廣播排程結果:利用(1, m)索引資訊	21
圖2.11:廣播排程結果:使用分散式索引技術	22
圖2.12:存取時間與傾聽時間之範例	24
圖3.1:資料碰撞示意圖	26
圖3.2:用戶端之請求所產生的資料競爭關係	30
圖3.3-a:資料競爭關係圖	31
圖3.3-b:廣播排程結果	31

圖3.4-a:處理後之資料競爭關係圖 .....	32
圖3.4-b:處理後之廣播排程結果 .....	32
圖3.5:處理後之廣播排程結果 .....	33
圖4.1:資料存取關係圖 .....	36
圖4.2:資料競爭次數關係圖 .....	38
圖4.3:搬動頂點對資料競爭次數影響關係圖 .....	39
圖4.4:二階分割演算法之虛擬碼 .....	42
圖4.5:初始化廣播資料 .....	43
圖4.6:初始化資料競爭次數 .....	44
圖4.7:搬動最大 $Gain_{V_i}$ 且符合其它條件之 $V_i:V_1$ .....	45
圖4.8:搬動最大 $Gain_{V_i}$ 且符合其它條件之 $V_i:V_6$ .....	45
圖4.9:搬動最大 $Gain_{V_i}$ 且符合其它條件之 $V_i:V_2$ .....	46
圖4.10:搬動最大 $Gain_{V_i}$ 且符合其它條件之 $V_i:V_5$ .....	46
圖4.11:搬動最大 $Gain_{V_i}$ 且符合其它條件之 $V_i:V_3$ .....	47
圖4.12:搬動最大 $Gain_{V_i}$ 且符合其它條件之 $V_i:V_4$ .....	47
圖4.13:演算法處理後，相關變數資訊 .....	48
圖5.1:均勻分佈:廣播資料項對競爭次數之影響 .....	51
圖5.2:均勻分佈:選擇率對競爭次數之影響 .....	53

圖5.3:常態分佈:廣播資料項對競爭次數之影響 ..... 54

圖5.4:常態分佈:選擇率對競爭次數之影響 ..... 55

# 第一章、緒論

在本節中，我們描述本文研究之環境、動機、研究目的與本文架構。

## 第一節 研究背景

無線通訊技術的快速發展，行動用戶端可以於任何的時間、地點存取相關的資料。在無線環境當中，用戶端有電力與頻寬的限制，所以用戶端關心的是如何消耗最少的電力來達成快速取回資料。在圖 1 的架構圖中，描述一個無線網路的架構圖。該圖中包括三個主要的元件，分別是行動用戶端，廣播伺服器與有線(Wired)網路線路。在無線環境當中，行動用戶端，如筆記型電腦、行動電話與個人數位助理(PDA)。這些行動裝置可以利用無線的通訊協定(Protocol)來取得自己需要的資訊。而廣播伺服器負責處理要被廣播的資料項，並利用廣播的傳輸方式傳送給某一區域的行動用戶端。在二個廣播伺服器之間，因為伺服器之間距離過遠的原故。所以利用有線的網路線路讓二個伺服器之間可以直接的連線。

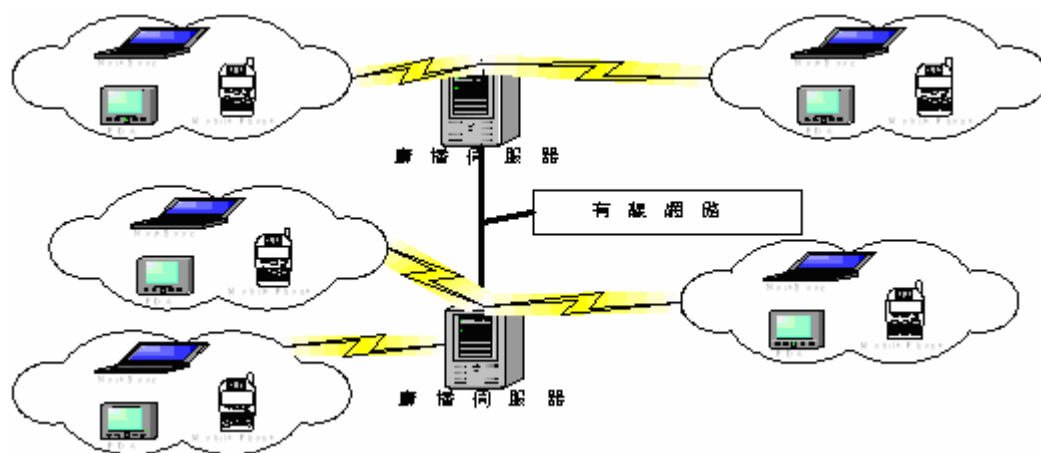


圖 1.1:無線網路架構圖

從行動用戶端與廣播伺服器的觀點來看，行動用戶端想要消耗最少的電力來取回自己所需要的資料項。廣播伺服器在處理要廣播的資料時，關心的是如何有效的將資料進行排程後傳送給行動用戶端，這二者之間存在著相互影響的關係。當廣播伺服器廣播出來的資料項順序對所有用戶端在存取資料為最佳化的排程時，行動用戶端即可快速的取回資料並消耗最少的電力。在廣播排程的問題上，有相當多的學者[8,9,13]提出一些有效的方法在處理廣播排程的問題。為了增加用戶端在存取資料的效率上，一些學者提出了快取(Caching)[12,26,50]和索引(Indexing)[38,49,51]的方法，來幫助行動用戶端存取資料之效率。在廣播排程的問題上，學者提出根據資料的存取頻率與heuristic的演算法，來決定資料廣播排程的問題。而以上所提的方法，不外乎是為了提昇用戶端和伺服器在運作時的效率。這些相關的研究方法我們在第二章中明確描述其中的運作細節。

## 第二節 動機

在無線環境當中，因為行動裝置的特性，所以節省電力消耗和有效地利用網路頻寬(Bandwidth)是相當重要的議題。在電力消耗方面，因為行動裝置所攜帶的是一個小型的電力裝置，所以他所運作的方式必須要相當的節省電力，才能夠有較長的待機時間。無線網路的頻寬因為在該環境之中，相對於有線網路的網路資源相對較低。所以充份的利用網路資源對伺服器而言，是相當重要的。

對行動裝置而言，傳統的 Client-Server 架構中的資料傳輸模式：因為頻寬充裕，用戶端可以對伺服器提出一對一的請求資料。而這樣的模式已經不再適用於無線的通訊環境當中，因此使用廣播的傳輸方式可以有效的處理大量的行動用戶端對於資料請求的模式。

對廣播伺服器來說，廣播排程是屬於NP-Complete[34]的問題。所以許多的研究者提出一些演算法來處理該問題。而在這些方法中，想要去取得一個最佳化的廣播排程結果。而一個最佳化的廣播排程對用戶端而言，將可以快速且用最省電的方法去取回需要的資料。

### 第三節 研究目的

在無線廣播的環境當中，研究者將研究的重心放在單一通道(Single Broadcast Channel) [10,18,22]與多通道(Multiple Broadcast Channel)[8,16,19,20,25,34,35,37,41,49]的廣播環境當中。對於請求的資料項而言，有單一請求資料(Single Data Item)[34,37]與多資料(Multiple Data Items)[32]的請求方式。在多通道廣播環境當中，研究者未考慮到一個情況：當在多通道的廣播環境當中，伺服器所廣播的資料被分配到各個廣播通道上進行廣播時，當用戶端需要的資料集合被分配在不同的廣播通道上廣播中，對用戶端而言，請求的資料項將有機會在相同的時間被廣播。但用戶端在相同的時間點上，只能存取多通道中的某一個通道的廣播資料。當請求的資料被分配到不同的通道時，將產生一個問題，相同的用戶端請求的資料項在廣播排程上將有機會於相同的時間被廣播出來，我們稱這樣的情況為資料競爭。當廣播排程的結果，對於同一筆用戶端所請求的資料於相同的時間點上廣播出來時，我們稱該情況為資料碰撞。當該情況發生時用戶端必須等待到下一個廣播週期才能將資料取回，如此便增加用戶端的存取時間。當資料碰撞發生的次數增加時，對於全體用戶端而言，存取的時間勢必增加。

在本文當中，我們解決在多通道的廣播環境當中，用戶端提出多個資料請求時所發生資料競爭的問題。為了解決這個缺點，在本文當中，我們應用"二階分割演算法"來處理這樣的問題。主要的精神為將

廣播資料做分割的動作，來達到最小化資料競爭的次數，而分割廣播資料的方式即是根據全體用戶端請求後所產生的存取關係圖為依據，並利用本文所提的演算法，對資料進行分割動作。而分割後產生的群體表示成一個廣播通道的廣播資料。而分割的目的是讓資料競爭的次數達到最小，讓伺服器對資料進行廣播排程時，可以產生最少資料碰撞的情況。

本文所提出的演算法主要的目的為：減少行動用戶端在多通道的廣播環境之中，請求多個資料時，降低用戶端存取資料所花費的廣播週期次數。在使用本文所提之演算法後可以有效地減少該情況的發生，這樣的結果，預期可以增加全整用戶端在存取資料時的效率。

#### 第四節 研究架構

本文架構如下：在第二章描述在無線廣播環境之中相關的研究。第三章中定義本文所要解決的問題。第四章提出本文所使用的演算法。實驗結果呈現在第五章中。最後，第六章提出結論並說明未來的研究方向。



## 第二章、文獻探討

在無線環境當中，運作架構主要可以分成二大類，分別是隨意網路(Ad Hoc)與無線廣播(Wireless Broadcast)二種。在Ad Hoc的架構中[30,46]，主要包括的是行動主機(Mobile Host)。每一個行動主機可以直接的與鄰近的行動主機進行連線，而無透過任何的中央主機來進行通訊的動作，因此每一個行動主機都扮演著資料傳輸(Replay)的功能。在該環境當中，因為行動主機的特色，所以出現網路繞徑(Routing)[30]與電力(Power)平衡[46]的議題。在網路繞徑的問題上，因為行動主機可以任意的移動，所以網路拓樸也是動態的改變。在電力平衡的問題上，在該環境中的行動主機，每台主機所擁有的電力不盡相同，若要處理相同的工作，對於擁有較大電力的主機而言，尚可負擔。但對於小電力的主機而言，於一段時間內，電力將消耗過多，導致無法正常運作，而這將影響整體的運作。在圖 2.1 中，我們說明 Ad Hoc環境中的基本運作模式。在該圖形中，行動主機把接收到的資料往鄰近的主機傳送，當接收到的資料為目的地時，則收下該資料。否則往目的端的路徑傳送。對於繞徑的問題，因為用戶端可以任意的進入該環境當中，當某一個用戶端離開後，若網路拓樸沒有動態的跟著改變，則資料將無法順利傳輸到目的端的電腦中。電力負載問題也是相當的重要，因為在相同的環境之中，每台主機負責相同的工作，如何有效動態的平衡主機的工作量，來增加整體服務的時間，在該環境中也是相當重要的議題。

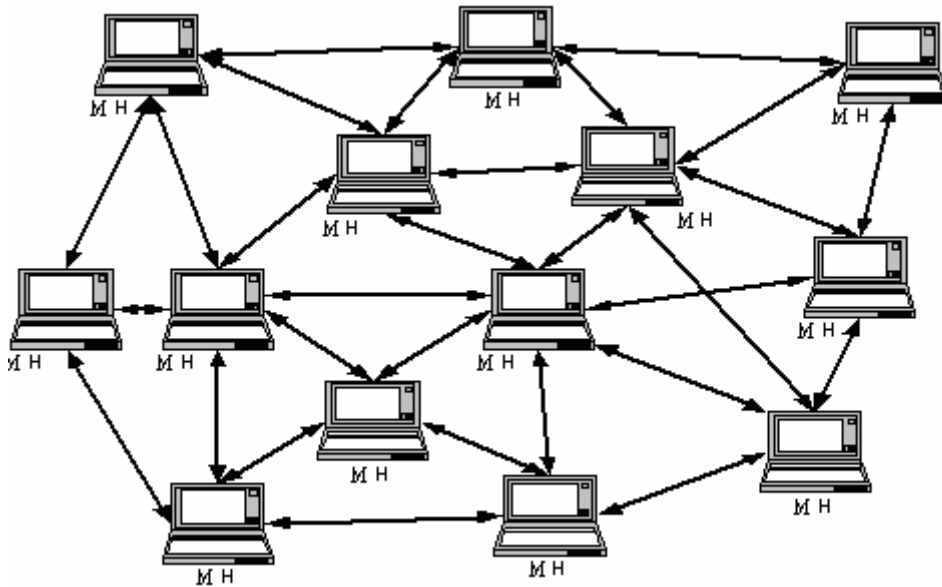


圖 2.1:Ad hoc 基本運作示意圖

在無線環境中，另一個環境為無線廣播。而該環境為本文所要探討的領域，因此我們將清楚地介紹該環境中的細節。無線廣播最主要的運作模式為伺服器連續且重複的廣播資料給行動用戶端。當伺服器廣播出來的資料為行動用戶端所需要的資料時，行動用戶端接收該資料項，直到所有請求的資料接收完畢。這樣的傳輸模式主要的目的是為了有效利用伺服器的頻寬。為了讓行動用戶端花費較少的時間來存取資料。早期一些研究者提出廣播磁碟[2,3](Broadcast Disk)的觀念。主要想法為利用磁碟之間轉速的不同來做區別，將伺服器要廣播的資料項中，區分成熱門資料(Hotter Data Item)與冷門資料(Colder Data Item)，將熱門的資料放在高轉速的磁碟之中。因此在該磁碟中的資料項被廣播的次數將較為頻繁，相對來說冷門的資料被廣播的次數較低。對用戶端而言，被廣播較頻繁的資料項被存取到的機率將較高。

而在無線廣播的環境當中，資料傳輸的模式主要可以分為以下三種模式，以下我們說明其中的運作流程。

## 第一節 資料傳輸模式

在無線廣播的環境中，有三個主要的傳輸模式：分別是廣播模式 [1,2,14,16,40,45,51](Pure Broadcast)、需求模式 [5,11,15,38,42,47](On-Demand)與混合模式[4,6,7,8,31,40,43] (Hybrid)三種。在需求模式當中，用戶端可以透過上傳的通道，將請求的訊息傳給廣播伺服器，而伺服器根據用戶端所請求的資料傳輸給用戶端。該模式的架構如圖 2.2所示。

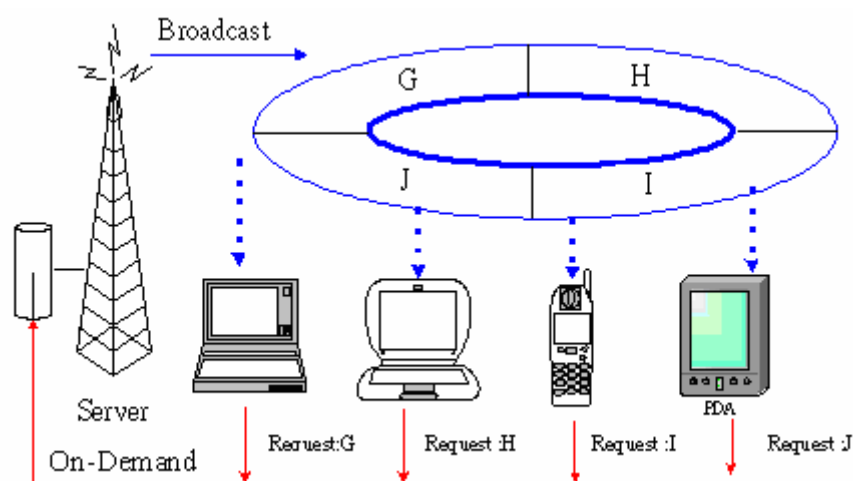


圖 2.2:需求模式之廣播架構圖

需求模式也被稱為拉式(Pull Based)的廣播模式。在圖2.2的架構圖中，行動用戶端分別對廣播伺服器提出資料的請求，而廣播伺服器根據用戶端所提出的請求來廣播資料給行動用戶端。

在該模式中，因為用戶端對伺服器提出資料的請求必須透過上傳的通道，但在無線的架構中，資料傳輸是非對稱式(Asymmetric Communication) [47]的通訊方式。因此，當有大量的用戶端對廣播伺服器進行資料的請求時，對廣播伺服器而言，將產生資料負載(Overload)的情況。針對這樣的問題，在[52]中，作者考慮到一些廣播環境中的因素，如資料項本身的大小、廣播伺服器所廣播的資料量、

以及用戶端請求資料項的熱門程度去進行分析，並利用heuristic演算法來決定資料廣播的優先順序。利用該文章所提的演算法，可以有效地增加上傳通道(Upload)的使用效率。

在需求模式當中，請求的資料通常必須要考慮到時間限制的問題。而在[15]中，作者所提出的演算法，可以確保用戶端所提出之請求可以在期限(Deadline)內處理完成。在評估該模式的重要衡量指標為回應時間(Response times)。而回應時間指的是從用戶端提出請求到伺服器開始提供服務所間隔時間。該時間愈小表示伺服器的服務效率愈高。

在[5]中，作者所提出的方法對於個別資料項最差回應時間的情況與整體用戶端的平均回應時間一併考慮，來達到一個平衡的狀態。

在廣播 (Pure Broadcast) 的模式當中 [1,2,8,10,14,16,32,36,40,45,51]，廣播伺服器對於要廣播的資料項重覆且持續地(Continuously and Repeatedly)進行廣播的動作。而用戶端對廣播伺服器只是進行傾聽(Listening)的動作。當伺服器廣播的資料項(Data Item)是用戶端感到興趣的資料項時，用戶端利用下載(Download Channel)的通道取回該資料項。該模式之架構如圖2.3所示。在圖2.3中有二個主要的組成元件，分別是廣播伺服器與行動用戶端。廣播伺服器主要的工作是將資料庫中所提供的資料持續且重覆的廣播給負責區域中的行動用戶端，而行動用戶端傾聽廣播通道，並將所有想要的資料取回，完成存取的动作。廣播模式也被稱為推式(Push Based)的傳輸模式。在該模式當中，當伺服器所要廣播的資料量增加時。這對全體用戶端在存取資料時，將增加存取資料的時間。

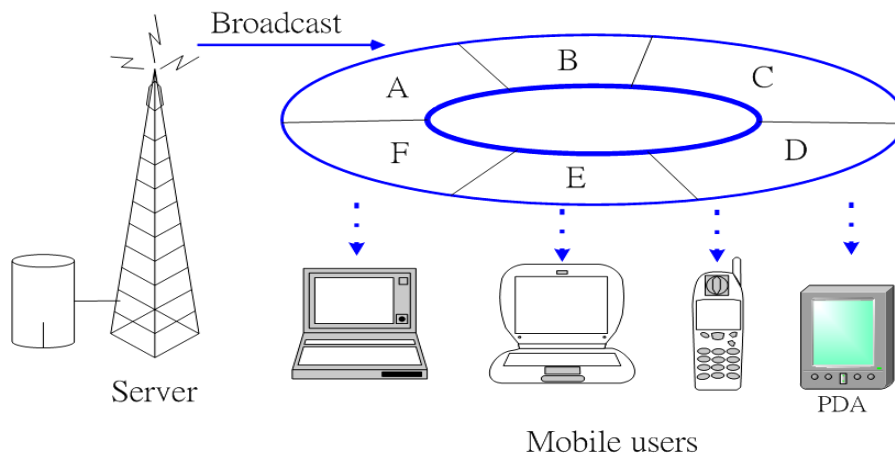


圖 2.3:廣播架構示意圖

廣播排程已被證明為NP-Complete[34]的問題，所以相關的學者[8,10,13,32]分別提出一些演算法來處理該問題。

為了讓行動用戶端在存取資料的時間達到最小化的狀態，在[13]中，作者利用貪婪法(Greedy Method)，去處理廣播排程的問題。此方法我們將於第六節，清楚說明該方法的運作流程與核心的想法。

在[10]中，該文作者對QEM[13]演算法進行修改，在該文中，額外利用資料採礦(Data Mining)中的關連式法則(Association Rule)，利用該方法找出最大項目組(Large Item Set)。而最大項目組表示成全體用戶端存取較熱門的Pattern，對於這些熱門的請求將優先處理，這對於整體的存取效率將是有助益的。

作者[8,36]，利用動態規劃(Dynamic Program)的方法在多通道的廣播環境中的各別通道，進行最佳化的排程配置。多通道的廣播環境將於第二節中說明。

在[48]中，該文考慮到資料的存取頻率(Access Frequency)與二個資料之間的關聯程度。作者的主要想法是將所有要廣播的資料轉成一個圖形，每一個資料在圖形中表示成一個頂點。圖形上的邊由二個資料之間的存取頻率所構成。而廣播排程的順序由圖形上邊的權重值最

大的邊所組成的資料項，優先進行廣播動作。

當結合廣播模式與需求模式，則該模式稱為混合(Hybrid)模式[4,7,8,31,40,43]。在該模式當中。用戶端不僅可以傾聽(Listening)廣播通道，而且可以對廣播伺服器進行請求資料的動作。一般的廣播模式。大部份行動使用者所請求的資料為靜態的(Static)，正因如此，伺服器重覆且連續的廣播資料尚可滿足全體用戶的需求。對於靜態的廣播資料，學者提出利用統計分析的方法，對廣播資料進行存取頻率分析。對於分析之結果，將請求熱門的資料用廣播的方式處理。因為當某一個資料被大量的請求時，對於上傳通道的負擔將增加。若將該資料利用廣播的模式傳輸給行動用戶端，如此一來將可減少上傳通道的負擔。而廣播出去的資料對全體用戶端將有最大的效能。對於廣播頻寬的使用將可大大的提昇。而對於較冷門的資料，因為要求的用戶端人數較少，故利用需求模式去各別的處理。

對於廣播伺服器負載的問題，在[7]中為了避免此問題，提出 $R * W$ 的處理模式。在該模式中，考慮二個因素來取得中間的平衡點。在該文當中， $R$ 代表的是用戶端的請求(Request)。 $W$ 為一段時間週期內，尚未被廣播出去之資料項所等待的時間。利用二個資料結構體去儲存該二個變數，並將二個變數值相乘。將所得之乘積，取最大值對映到相對映的資料項中，並做為廣播資料的順序。

在現實的生活中，用戶端所需要的資料項是會動態的改變。根據這樣的特性，有些演算法即是根據這些特性在設計。如[36]，該文作者考慮到用戶端所請求的資料項與廣播伺服器當下的負載情況，動態的決定二者之間的最佳平衡點。該演算法減少上傳通道的負載，並且廣播最有價值的資料項並傳輸給行動用戶端。

在設計混合模式的演算法中，通常考慮廣播模式與需求模式的相關特性。如[40]中，作者設計出一個動態規劃的演算法。該演算法考慮所有要廣播的資料並找出所有廣播資料最佳的延遲時間，來決定特定的時間點上要廣播的資料項。

於[43]中被提出一個有效的混合式演算法，該文中將熱門的資料利用廣播的方式進行處理。而冷門的資料利用需求模式的演算法去處理。在[23]中，作者也考量上述二種因素。但在[43]中，該演算法應用在單一通道的廣播環境之中。而[23]應用於多通道的廣播環境當中。對於熱門的資料項有固定的廣播通道負責處理，冷門的資料利用拉模式在專門處理資料的傳輸。

## 第二節 廣播環境介紹

在本文中，因為所研究的環境是屬於廣播的範圍，因此以下我們將詳細的描述該環境之中相關的环境特性。

### 一、廣播時效性

在無線廣播的環境當中，對於請求時間的考量也是相當重要的議題。當用戶端所請求的資料必須於特定期限(Deadline)內完成時，該模式被稱為及時性(Read-Time)[9,29,39]的廣播模式。在該環境當中，用戶端請求一個相當緊急的服務，伺服器必須在特定的時間內快速的給予回應，否則該請求將變成一個無效的請求。在該模式中典型的例子為軍事上的應用。如二軍飛機於空上交戰時，一方發射飛彈攻擊另一方，被攻擊方提供資訊的伺服器接受到被攻擊飛機所提出的緊急請求，要求伺服器提供相對映的處理訊息。若伺服器無法在特定的時間點上提供訊息，則被攻擊之飛機勢必會被擊落。

在混合模式的環境中，[29]考慮每一個資料所被廣播出去所需要

之頻寬、廣播的歷史資訊與廣播伺服器的廣播週期，三種因素去決定出優先權較高的資料項，並在期限內處理完成。在[9]中，利用EDF(Earliest Deadline First)演算法去處理工作指派(Assignment Scheduling)的問題，並在一個及時性的廣播環境當中，取得一個最佳化的廣播排程。

在[8,10]中，該文的環境並無考慮時效的問題，在此二篇文章當中，假設廣播伺服器可以收集行動用戶端所提出的請求，並對歷史的廣播資料進行分析，並利用各式的演算法去配置出一個最佳化的排程結果，利用該結果對資料進行廣播的動作。

## 二、廣播通道數量種類

### 2.1 單一廣播通道(Single Channel)

在無線廣播的環境當中，伺服器將資料廣播給行動用戶端，當需要的資料出現在廣播通道時，用戶端將需要的資料取回。在無線的環境當中，若伺服器在廣播資料，只利用到一個廣播通道在處理時，該環境稱為"單一通道"。在該廣播模式中，研究者[10,18,22]把焦點放在如何利用有效的演算法來取得最佳的廣播排程，使得用戶端在存取資料時，所花費的存取時間達到最小。

以下，我們利用簡單的範例來說明該模式的運作流程，並將結果顯示於圖 2.4 中。

假設伺服器使用單一廣播通道而且所負責廣播的資料項集合包括： $\{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8\}$ ，在不考慮廣播排程的順序的之下，將產生圖 2.4 的結果。



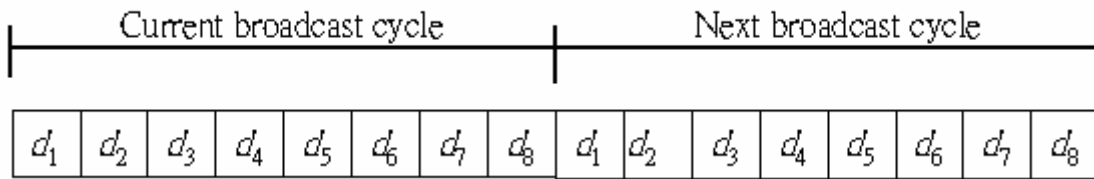


圖 2.4:單一廣播通道之結果

在該圖中，當伺服器廣播這些資料項給予用戶端時，用戶端傾聽 (Listing)廣播通道目前所廣播的資料項。在此模式中，用戶端任意的進入廣播通道中傾聽目前所廣播的資料項，伺服器在特定時間點上只有廣播一個資料項。正因為用戶端進入的時間點是無法預計的，所以當伺服器廣播的前一個資料項是目前正進入通道傾聽之用戶端所想要的資料項，則該用戶必須等待到下一個廣播週期的來到且伺服器正在廣播該資料項時。用戶端才能取回該資料項。

## 2.2 多通道廣播環境(Multi-Channel)

當廣播伺服器廣播資料，利用一個以上的通道在處理時，則稱該模式為多通道廣播環境。[8,16,19,20,25,34,35,37,41,49]。在這樣的環境當中，各個通道資料配置的種類可分為二種：一種是平坦式[16]的分配方式，另一種是非平坦式[16,36,41,49]的分配方式。平坦式的模式對於廣播的資料項而言，將平均的分配到各個通道中，而非平坦式的方式，則將特定的通道分配較少的資料項，某些通道分配較多的資料項。其中的優點是分配較少資料的通道上，在相同的廣播週期內，資料將重覆較多次。這對用戶端來說，將有更多的機會在廣播通道上接收到想要的資料，多通道排程範例，利用以下範例說明。

假設伺服器廣播的資料項集合為： $\{d_1、d_2、d_3、d_4、d_5、d_6、d_7、d_8\}$ ，並且利用二個平坦式廣播通道在處理廣播資料。將產生圖 2.5 的結果。在圖形中，廣播資料平均分配在二個通道之中，每一個廣播

週期內資料項出現的機率均相同。

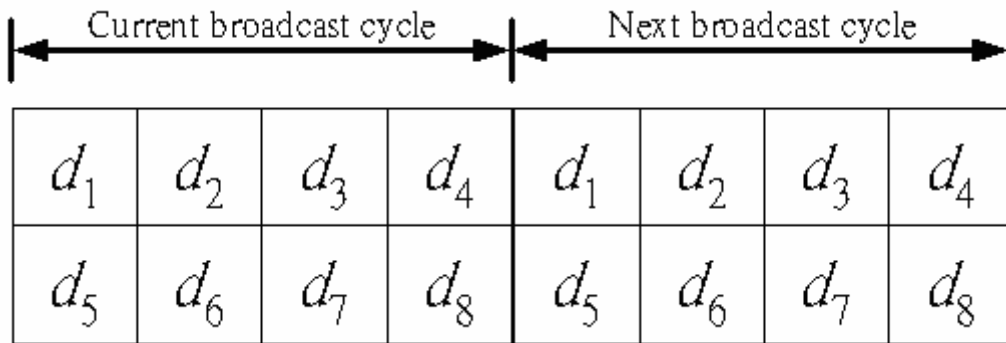


圖 2.5:多通道廣播排程結果:各個通道資料平坦式分配

對於非平坦式廣播方式，學者[36,49]，也提出相關的方法。例如根據廣播資料的屬性，動態的對各個通道的廣播數量進行調整。例如利用樹狀結構，去對映到多通道的廣播環境。這樣的結構表示在樹狀結構當中，樹的每一層對映到一個廣播通道上。並對每一個通道進行最佳化的動作。

在[36]中，作者將所有要廣播的資料項，依據資料被存取的頻率，將資料項被存取的頻率進行排序的動作。並將不同頻率的資料項，分配到不同的廣播通道上，而作者也考慮到當資料的存取頻率改變時，對映的樹狀結構也隨著改變。而這樣的設計對於整體用戶的存取時間，將有效的減少。

樹狀結構在多通道中被廣泛的使用。在[49]中，作者也考慮利用樹狀結構去建構多通道的廣播排程。該作者利用VF (Variant Fan-Out) 去產生階層式廣播排程，再利用稱為  $VF^k$  的heuristic演算法，去取得一個最佳化的廣播排程，來降低整體的存取時間。

典型的非平坦式方式，我們用圖2.6來簡單說明。在圖2.6中資料項  $d_1$  與  $d_2$  在相同的廣播週期內，資料項出現次數較多，而其它的資料項出現的機率均相同。

Current broadcast cycle						Next broadcast cycle					
$d_1$	$d_2$	$d_1$	$d_2$	$d_1$	$d_2$	$d_1$	$d_2$	$d_1$	$d_2$	$d_1$	$d_2$
$d_3$	$d_4$	$d_5$	$d_6$	$d_7$	$d_8$	$d_3$	$d_4$	$d_5$	$d_6$	$d_7$	$d_8$

圖 2.6:通道廣播排程結果:各個通道資料非平坦式分配

### 三、用戶端請求資料項之數量

從用戶端的觀點來看，當用戶端對伺服器所提出的資料項，只有請求單一資料項時，將它稱為"單一資料"請求(Single Data Item) [34,37]。當請求的資料項大於1個資料項時稱"多個資料"請求[32,36]。

在真實的環境當中，請求單一資料是不切實際的，因此在[32,36]作者提出請求"多個資料"的方式。這樣的模式對於真實環境而言，是比較可行的方案。舉例來說:當張三想要知道自己擁有的股票，分別有 IBM, Intel 與 Oracle 的股票價格。在請求單一資料的模式中，張三必須對廣播伺服器提出三次資料請求的動作。這對伺服器上傳的資源相對消耗較多。而在請求多資料的模式當中，張三只須對伺服器提出一次的請求即可要求伺服器提供所有資料的服務，對上傳資源消耗較少。

從用戶端請求資料數量的觀點看來，當用戶端請求較多的資料項與單一請求資料的模式之下，在相同的排程結果之下將花費較多的存取時間。

在[32]中，作者修改了QEM的方法，並考慮資料存取頻率與請求資料項重覆(Overlapping)的比例。對於要新增到廣播通道的資料考慮較多的因素。故在整體的存取效率上將有較大的提昇。

在本文中，我們考慮到用戶端所請求的資料項為 "多個資料" 請求。廣播通道的數量為 "多通道" 廣播模式。

#### 四、 資料範圍

在地理資訊系統(GIS)的應用上，用戶端搜尋在地圖上的某一範圍的特定資料。在這樣特殊的請求上，為了要節省頻寬，廣播伺服器廣播具有範圍的資料來節省頻寬。這樣的特性，也廣泛的應用在資料庫系統上。舉例來說：資料庫系統提供公司股價查詢的服務，某一使用者想要查詢股票從10元到50元的公司。在這樣的情況下，若伺服器將所有股價公司的資料全部廣播與只廣播該範圍的資料來比較，將浪費更多的頻寬。

作者[51]利用空間分割演算法(Space Partition Algorithm)於地圖上作空間的切割。並利用D-Tree的資料結構去處理具有地域性關係(Location-Dependent)的廣播模式。

#### 五、 廣播資料重覆性

在一個廣播的週期內，當所有要廣播的資料項在該週期內沒有重覆地被廣播，這樣的廣播模式被稱為 "Flat"。若資料項有在該週期內被重覆的廣播超過一次，則稱為 "Non-Flat"。在Non-Flat的模式當中，最大的優點是重覆廣播的資料對用戶端來說，有較高的機率可以被廣播。但重複廣播特定資料的結果，對於廣播週期來說，相對的變長。在過錯想要的資料時，必須付出較多的等待時間。來收到想要的資料。為了比較二者的差異，在圖2.7中說明Flat的廣播結果。圖2.8中說明Non-Flat的廣播結果。在圖形2.8中，資料項 $d_2$ 在同一個廣播週期內被廣播了3次，所以所有的用戶端對於 $d_2$ 的存取機率將增加。對於二種模式的廣播週期，Non-Flat的模式明顯的較長。所以要選擇二種模

式中的任一個模式，必須考慮到用戶所請求的資料類型來決定選擇的廣播模式。在本文中我們所使用的為Non-Flat的廣播模式，並利用於我們的演算法當中。



圖 2.7:相同廣播週期內使用 Flat 廣播模式

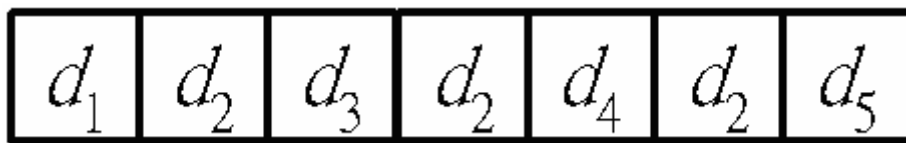


圖 2.8:相同廣播週期內使用 Non-Flat 廣播模式

## 六、廣播排程策略(Scheme Policy)

在本小節中，將介紹一些著名的廣播排程策略。以下將描述這些方法的基本精神與運作流程。

### 1. First Come First Served

該策略廣播的順序是根據被請求資料到達伺服器的時間為依據。為了達到公平的目的，該方法假設每一個資料項被請求後傳輸至伺服器的時間均相同故將傳輸遲緩的因素排除在外。

### 2. Earliest First Request (EFR)

該排程策略與作業系統工作排程的想法相同，在作業系統的運作方法是依據工作被請求的順序為優先順序。而在廣播的應用上，即是將最先被用戶端請求的資料項做為廣播的優先順序。這樣的廣播方式缺點為若先被請求的資料項的大小較大時，伺服器將花費較長的時間廣播該資料項。對於其它的用戶端必須等待較長的時間。

### 3. Most Request First (MRF)

在該方案中，假設被請求的資料項，伺服器可以事先知道各個資料被行動用戶端請求的次數。伺服器將請求最多次的資料項優先廣播。該方法的優點是熱門的資料項，用戶端可以較快的收到。但對於冷門的資料項而言將等待較長的時間，才能接收到資料項。

### 4. Longest Wait First (LWF)

在該策略的基本方法為：伺服器記錄每一個資料項已經被請求後，但尚未被廣播出去之資料項的等待時間(Wait Time)。當等待最久的資料項被廣播後，伺服器接著挑選次高的等待時間並將它廣播。此流程持續的進行。直到所有的資料項都處理完畢。在伺服器負載方面，當廣播資料項增加時，因為伺服器要記錄所有資料項的狀態，如此一來對於伺服器的負載將隨著資料量的增加而增加。

### 5. R\*W

在混合模式中，R\*W被廣泛的討論，在此方法中考慮到MRF與LWF二種排程的精神。利用二個序列(Queue)將被請求與等待最久的資料項分別存入序列當中。再對二個序列中的元素值相乘，將乘積中取最大值對映到相對的資料項。將該資料項優先廣播。考慮到等待時間與請求次數二種因素。讓用戶端等待的時間與廣播的效率可以取得平衡。

### 6. QEM

QEM[13]這個方法應用於用戶端請求"多個資料"的模式。該方法以用戶端所發出的請求為出發。每處理一個請求中的資料時利用(Greedy Method)的想法，在二個資料之間利用此方法會產生最佳的

Query Distance(QD)，來達到較佳的存取時間。但在該文中沒有考慮到二個請求之間。可能會有重覆(Overlap)請求的資料，所以在處理廣播的排程上面，將更加的困難。因此學者[10]，提出從用戶端的請求與該請求被讀取的機率。將被讀取機率較高的請求優先處理，這樣的處理可以滿足大多數的用戶端。對整體的存取時間將會大大的減少。

QEM最主要的想法為：在處理每一個資料時，必須先決定該資料項是否已被廣播。當尚未被廣播時，則將它安排在廣播排程之中，若為處理過後資料，則跳過不予處理。在此演算法當中，其中有三個主要的精神整理如下：

- 使用貪婪法(Greedy Method)。
- 目前所處理的請求，在安排廣播排程時不能影響之前所安排的結果。
- 每加入一個新的資料項，必須配置到有最小的存取時間。

## 七、提昇用戶端效能的相關技術

### 1. 索引技術(Indexing)

為了增加用戶端存取資料項的效率與節省電力的目的，使用索引技術[19,21,24,27,28,34,38,49,51]是一個有效的方法。在介紹索引技術之前，首先說明二種用戶端在無線環境當中運作的模式。分別是行動模式(Active Mode)與休眠模式(Doze Mode)。行動模式是用戶端必須要傾聽目前廣播伺服器中所廣播的資料，在無索引技術的方式之下，因為用戶端沒有任何訊息可以知道廣播伺服器下一個要廣播的資料為何，故要進行傾聽的動作。當傾聽到的資料為用戶端所需的資料項

時，才將它收下。在該模式之下，用戶端傾聽廣播通道時，電力消耗比休眠模式來得大。而使用索引技術時，索引資訊可以讓用戶端知道伺服器廣播資料的順序，進而可以進入休眠模式來達到省電的目的，並提醒用戶端何時廣播的資料項為用戶端所需要，而進入行動模式來收取資料。在使用索引技術之前，首先說明索引訊息與廣播資料之間的關係。因為索引資料使用樹狀結構來表示，所以要知道某一個資料的位置時，必須先從樹狀結構中的根(Root)開始搜尋，這表示當用戶端要使用索引資訊前，須要先從廣播通道中，將索引訊息取回。方可利用索引資訊將廣播資料取回。

在索引技術相關的研究，可以分成二種方式：分別是(1, m)索引與分散式索引，以下將分別說明運作流程與資料結構。

### 1.1 (1, m) 索引技術

(1, m)索引的處理方式是將要廣播的資料，在廣播資料的最前面的放置一份完整的索引資訊。索引資訊中，包括廣播資料的時間點。此技術如圖2.9所示。在圖中，是基本的(1, m)索引技術的樹狀結構圖。圖形中包括樹狀結構中的根(Root)I，分枝與葉結點。分枝表示的是資料的索引訊息，葉結點指向的是的伺服器所廣播的資料項，在該圖中可以清楚地看到目前伺服器廣播的資料項 $d_1$ 至 $d_9$ 。在圖2.10當中，Data 1、Data 2, …Data m。表示目前伺服器所廣播的資料集合。Index為所有資料項的索引資訊，在一個廣播週期之中，伺服器首先將索引資訊廣播出去，緊接廣播資料集合。對於用戶端而言，可以任意的進入廣播通道進行傾聽的動作。當進入的時間點上，伺服器在前一個時間點上，正在廣播索引訊息，則用戶必須等到下次的週期中的索引訊息出現，收下該訊息後，方能進行資料存取的行為。正因為用



戶端進入廣播的時間是任意的，為了減少用戶端探索(Probe)索引訊息的時間。學者提出了改良的索引資訊。該結構被稱為分散式索引，下一節當中，將清楚地說明。

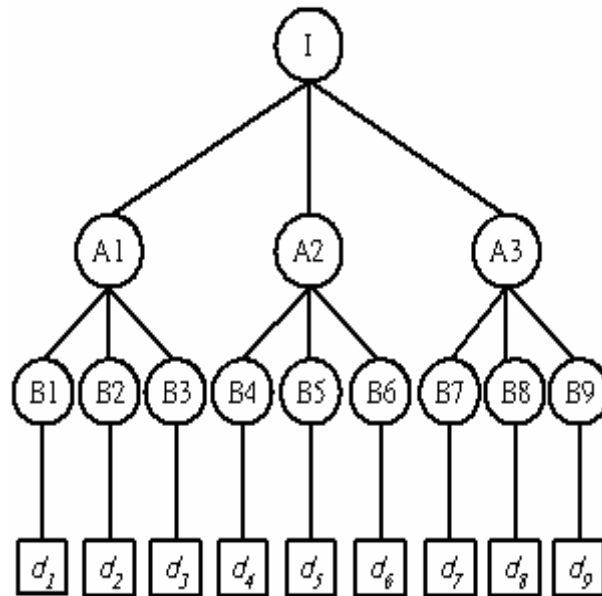


圖 2.9:(1, m)索引技術之索引樹

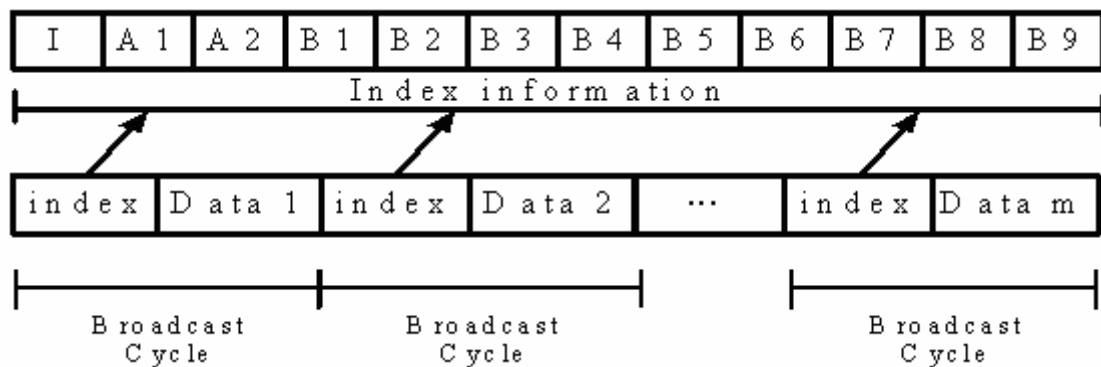


圖 2.10:廣播排程結果:利用(1, m)索引資訊

## 1.2 分散式索引技術

分散式索引技術為(1, m)技術的改良。因為在(1, m)索引技術當中，對於索引訊息的取得時間在整體考量上要花費較多的時間去探索(probe)索引訊息。而分散式索引技術將資料分成若干個區段，索引的訊息只需要索引到資料區段中資料項的部份即可。這樣的作法節省

了用戶端錯過索引訊息後，所需要等待到下一次索引訊息出現的時間。分散式索引技術的架構圖如圖2.11所示。在圖中，索引訊息只包括到該段的資料項的部份。相對(1, m)索引技術而言。此方法在廣播週期與索引訊息的probe時間，相對的減少。



圖 2.11:廣播排程結果:使用分散式索引技術

## 2. 快取技術(Caching)

資料快取的技術[3,12,26,33,44,50]，可以有效的增加行動用戶資料讀取的效率。快取技術是利用在用戶端本身的儲存裝置當中。預先將一些資料放在儲存裝置中。當需要該資料時，即從儲存裝置當中讀取即可，而不用透過廣播伺服的廣播來取得資料。而快取資料的決定，常用的方法是預先將熱門程度高的資料項，預先儲存於行動裝置當中。這樣的處理方式，最大的好處是可以減少熱門資料取回的時間。然而在行動裝置的特性中，裝置儲存的空間是有限制的。為解決這樣的問題，[12,50]提出了利用合作式快取(Cooperative Caching)的方式，來增加用戶端的快取空間容量限制。該方法最主要的想法是每一個用戶端知道鄰近點(Neighbor)的快取空間之資訊，再將用戶端所在位置附近的用戶端之快取空間各自分享，來增加快取的空間。該方法的運作流程為:用戶端想要獲得某一項資料，先從自己的快取空間當中去搜尋(Search)，發現沒有在快取空間時，便向鄰近的用戶端提出請求。若資料在鄰近點中找到時，便取回資料。若無發現，最後向廣播伺服器請求該資料項。

在快取技術的應用上，因為儲蓄裝置空間有限，所以資料取代(Replacement)的方法也是相當重要的議題。當用戶端所請求的資料改變時，預先快取的資料已是無效的訊息。這會造成快取失敗(Caching Miss)。因此[26]提出了利用索引技術去管理用戶端儲存裝置中的快取資訊。在快取技術的效率評估指標，最常見的為快取命中率(Hit Ratio)來評估。該值的計算方式為：在相同的快取空間條件上，請求資料於快取中取得的次數除請求資料的次數。當該值愈大時表示快取策略的效率愈高。

## 八、效能評估

在無線廣播的環境中，對於廣播效能評估有二個重要的效能指標，分別是存取時間(Access Time)與傾聽時間(Tuning Time)。存取時間指的是用戶端向伺服器開始提出請求到完整的取回自己需要的時間週期。傾聽時間討論的是用戶端花費在傾聽廣播伺服器的時間上。在二者的指標當中，存取時間指的是伺服器廣播排程的結果對用戶端存取資料的效能。傾聽時間是在使用索引技術後，用戶端可以利用索引資訊去決定進入休眠模式，進而減少傾聽廣播通道的時間。

於圖 2.12 中，我們利用圖形去說明二者之間的計算方式。在圖形中廣播排程使用索引技術。假設該訊息廣播花費一個時間單位，伺服器廣播的資料包含 $D_1$ 至 $D_4$ 且每一個資料項廣播花費一個時間單位。用戶端請求的資料項包括 $D_2$ 及 $D_3$ ，進入廣播通道的時間點為I。

從圖中，可以計算出存取時為 4 個時間單位。在傾聽時間上因為利用索引資訊，用戶端在伺服器廣播資料廣播於廣播 $D_1$ 時可以進入休眠模式，來節省電力。

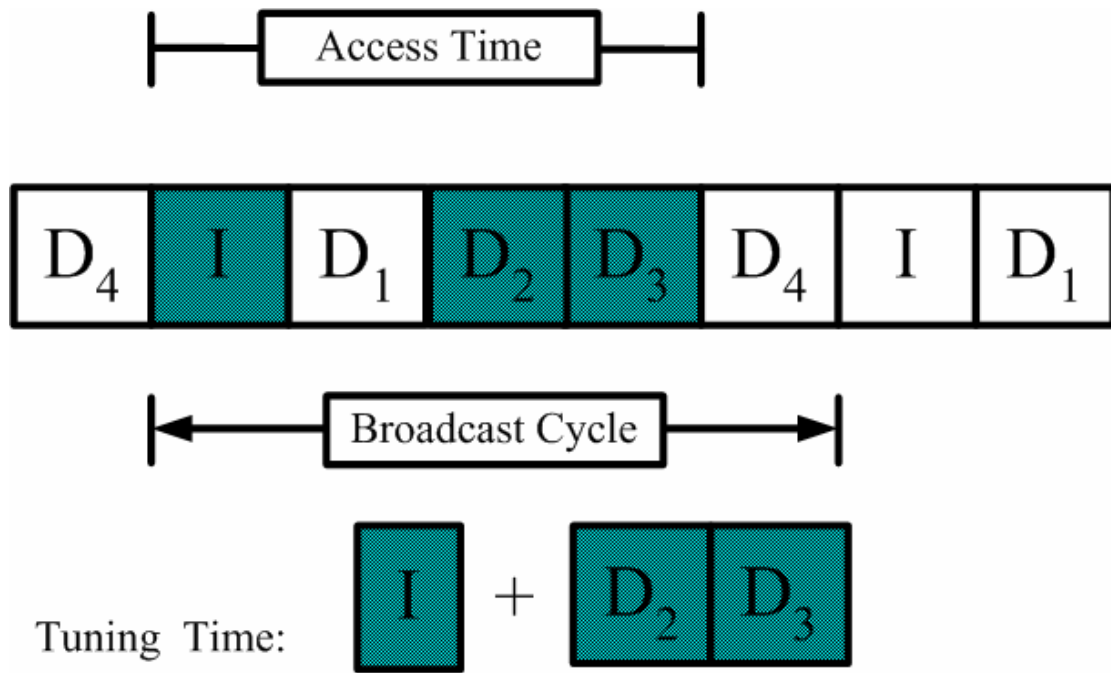


圖 2.12:存取時間與傾聽時間之範例

## 第三章、問題描述

在過去學者所提出的演算法中，處理的廣播環境有單一通道與多通道。用戶端請求的資料數量有單一資料與多個資料請求的模式。而這些演算法的目的，都是在取得最佳化的廣播排程，讓所有的用戶端在取回廣播資料時，可以利用最少的存取時間。但在多通道的環境當中，用戶端提出多個資料請求時，對於同一筆用戶端所請求的資料被分配於不同的廣播通道時，所面臨到資料競爭的問題。當這樣的情況增加時，於廣播排程上，將可能造成同一筆用戶端請求資料集合中，伺服器將請求的資料集合於相同時間點上廣播出來，這會造成資料碰撞的情形。當資料碰撞發生時，該用戶端必須等待到下一個廣播週期的到來，方能取回資料項。如此便增加用戶端請求資料的存取時間，這樣的結果對用戶端而言是所不樂見的。在本章中，描述在無線廣播環境當中，資料競爭與資料碰撞對用戶端在存取資料時所花費的廣播週期次數之影響。並定義相關的符號，最後舉例說明本文要解決的問題。

### 第一節 相關環境

在無線廣播的環境中，本文所要處理的是"多通道"的廣播環境，用戶端請求的資料是"多個資料"的請求方式。在多通道的環境當中，當用戶端請求的資料集合被分配於不同的廣播通道中時，對用戶端而言，將有機會造成伺服器把用戶端所請求的資料項集合於相同時間點上廣播出來，我們稱這樣的情況稱為資料競爭。當廣播排程的結果，發生同一筆請求的資料項於相同時間且不同通道中被廣播出來時，我們稱為資料碰撞。當資料碰撞發生的情況增加時時，對全體用戶端而言，勢必增加存取資料的廣播週期次數，這對用戶端而言是所不樂見

的。以下說明資料碰撞對用戶端在存取時間上的影響。

假設伺服器廣播的資料項包含 $\{d_1, d_2, d_3, d_4, d_5, d_6\}$ ，伺服器通道數量有2個廣播通道，分別是通道1與通道2。通道1負責廣播資料項為 $\langle d_1, d_2, d_3 \rangle$ ，通道2負責廣播資料項為 $\langle d_4, d_5, d_6 \rangle$ 。在通道中的資料項對廣播而言是有順序性的且資料是持續且重覆的廣播。用戶端提出請求的資料為 $\{d_1, d_4\}$ ，而用戶端提出的資料在存取順序上並無順序的關係。當廣播的資料為用戶端所需要時，便取回該資料項。用戶端在 $d_1$ 的時間上進入廣播通道進行傾聽廣播通道的動作，而廣播排程如圖3.1所示。對用戶端而言，進入的時間點伺服器廣播的資料項為 $d_1$ 與 $d_4$ 。這對用戶端而言，二者均為需要的資料項，但在同一時間點上，用戶端只能存取二個廣播通道中，其中一個廣播通道所廣播的資料項，另一個資料項必須等到下一個週期的到來時，才能存取該資料項並且完成資料存取的動作。

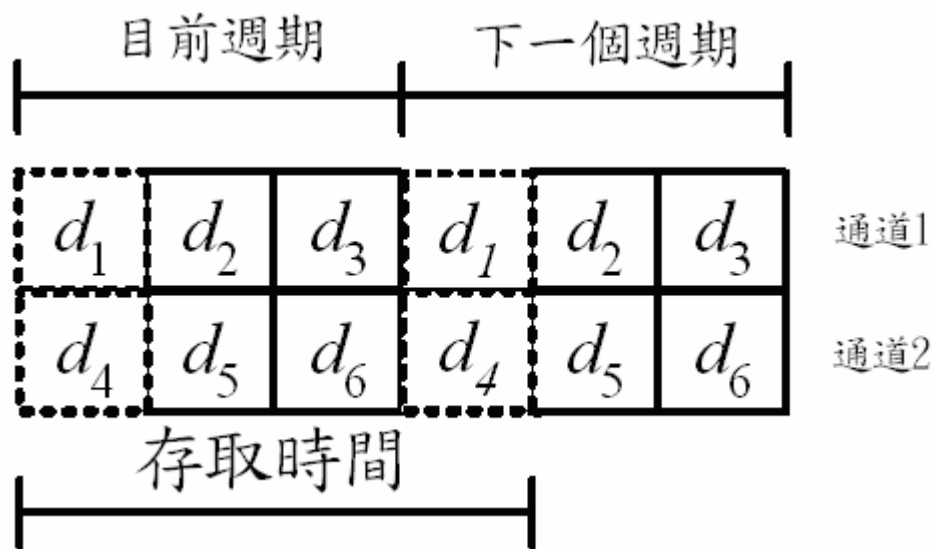


圖 3.1:資料碰撞示意圖

在本文中，我們減少所有用戶端請求資料後所產生的資料競爭次數，並達到最小的情況。這樣的結果對於在廣播排程上所產生的資料

碰撞次數，將可以有效的減少。如此一來，對用戶端在存取資料時所花費的廣播週期次數，將可以有效的降低。

## 第二節 符號定義與說明

為了方便說明本文所解決的問題，在表 3.1 中將列出一些相關的符號，並說明該符號所代表的意義。

表 3.1: 問題描述使用之相關符號

符號	說明
$d_i$	廣播環境中最基本的廣播單位。
$D$	廣播伺服器所有廣播的資料集合。 $D = \{d_1, d_2, \dots, d_m\}$
$ D $	伺服器廣播資料之數量。
$q_r$	在廣播環境中，提出資料請求的行動用戶端。
$Q$	行動用戶端的集合。 $Q \in \{q_1, q_2, \dots, q_k\}$
$ Q $	行動用戶端集合的數量。
$DS(q_r)$	用戶端 $q_r$ 所提出之請求中，包含的資料項集合。 $1 \leq r \leq k; DS(q_r) \subset D$
$ DS(q_r) $	用戶端 $q_r$ 提出請求的資料項所包含的資料個數。 $ DS(q_r)  \geq 2$
$C_i$	廣播通道之編號。 $i \in \{1, 2, \dots, h\}; 1 \leq i \leq h$

$C(d_i \approx d_j)$	<p>單一筆用戶端請求的集合中，資料項<math>d_i</math>與<math>d_j</math>在廣播排程上，發生資料競爭的情況。</p> <p>if <math>d_i \in C_a, d_j \in C_b</math>  <math>d_i, d_j \in DS(q_r); a \neq b</math>      <math>C(d_i \approx d_j) = 1</math>  otherwise      <math>C(d_i \approx d_j) = 0</math></p>
$DS(C_i)$	廣播通道 $i$ 所廣播之資料集合。 $DS(C_i) \subset D$
$ DS(C_i) $	<p>廣播通道 <math>i</math> 分配之資料項數量。</p> $1 \leq  DS(C_i)  \leq m$
$B$	<p>各個廣播通道之間分配到的資料數量的差距。</p> $\frac{ D }{h} * (1 - B) \leq  DS(C_i)  \leq \frac{ D }{h} * (1 + B)$ $0 \leq B \leq 1$

在表 3.1 當中， $d_i$  是無線環境中廣播的最基本單位。對用戶端而言，在同一個時間點上，僅能取回一個廣播資料。 $D$  為伺服器所負責處理的廣播資料並且廣播給用戶端的資料集合。 $q_r$  指的是在無線環境中，向伺服器提出請求資料的使用者。 $Q$  是廣播環境中，所有提出資料請求的用戶端。 $|Q|$  為所有提出資料請求用戶端的總數量。 $DS(q_r)$  包含用戶端  $q_r$  向伺服器請求的資料項集合。 $|DS(q_r)|$  為用戶端  $q_r$  請求資料項數量。在本文所提的演算法中，請求之資料項數量大於 2 個單位。 $C_i$  是廣播通道的編號。 $C(d_i \approx d_j)$  是判斷單一用戶提出的請求當中，二個資料項之間分配於廣播通道中的情況。判斷的方式為取得二個資料項於通道上的編號，當二者相同時表示資料項位於相同的廣播通道



中，無發生競爭情況，該值為零。若不相等時，表示資料項被分配到不同的廣播通道上，則該值為 1。 $DS(C_i)$  為各別的廣播通道中，通道所負責廣播的資料項。 $|DS(C_i)|$  是各別的廣播通道中所廣播的資料項數量。參數  $B$  表示，各個廣播通道之間所負責廣播的資料項數量之間的差距。在我們的演算法中，若搬動的資料項一直從特定的通道中搬移，則該通道負責廣播之資料項將減少，而搬移到的另一通道的廣播資料數量將會增加。這對於廣播排程將造成不平坦的結果，這對於資料在排程時，將有機會增加用戶端請求的資料發生碰撞的機會，這樣的結果是我們所不樂見的。而該參數即在控制演算法在搬移資料項時，控制不均勻結果產生的參數。

### 第三節 資料競爭之範例

為了明確地定義本文所提出的問題，本小節舉例說明資料分割的結果，對於用戶端在存取資料時，產生資料競爭的情況。

假設伺服器所廣播的資料為  $\{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8\}$ 。 $DS(q_1)$  為  $\{d_2, d_3\}$ ， $DS(q_2)$  為  $\{d_1, d_6\}$ 。廣播伺服器使用二個廣播通道。在圖形 3.2 中，以曲線分隔廣播通道的資料項，而資料項對廣播排程而言並無順序關係。每一個曲線為一個廣播通道所廣播的資料項集合。資料項與資料項之間的連接，表示用戶端請求該二個資料項的次數。當用戶端所請求的資料被分配於相同與不同的廣播通道時，對於分割結果，將會產生二種不同的結果，該結果顯示於圖 3.2 中。在圖 3.2 中，顯示的是  $q_1$  與  $q_2$  所提出的請求，對於資料競爭產生的結果。

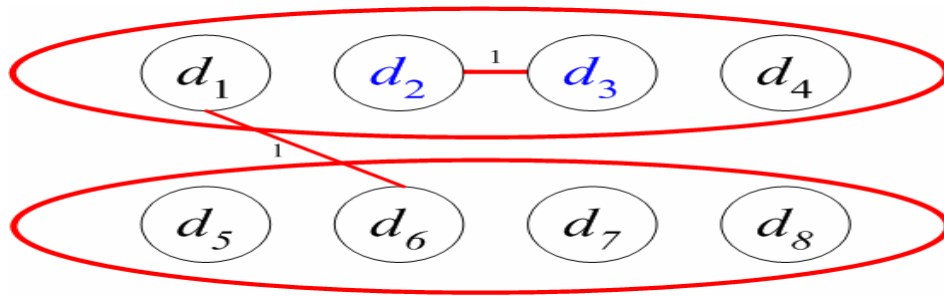


圖 3.2:用戶端之請求所產生的資料競爭關係

在圖3.2中，我們可以看出每一個用戶端發出請求後，所產生資料競爭次數的值。以 $q_2$ 為例，因為請求的資料被分配於不同的通道上，故產生1個單位的競爭次數。而 $q_1$ 請求的資料被分割於相同的通道上故無資料競爭的產生。在本文所提的演算法當中，將使用戶端請求資料後，所產生資料競爭的值達到最小的目的。這樣對於資料碰撞次數的產生，將會達到最少的情況。以下將舉例說明資料競爭對資料碰撞的影響。用戶端與請求的資料項集合如下所示： $DS(q_1):\{d_1、d_4\}$ ， $DS(q_2):\{d_1、d_3、d_4\}$ ， $DS(q_3):\{d_1、d_5\}$ ， $DS(q_4):\{d_3、d_5、d_6\}$ ， $DS(q_5):\{d_3、d_5\}$ ， $DS(q_6):\{d_1、d_5\}$ ， $DS(q_7):\{d_1、d_2\}$ ， $DS(q_8):\{d_1、d_4、d_5\}$ 。利用上述的用戶端請求的資料項，可以建立資料競爭關係圖，如圖3.3-a所示。在圖形中，曲線部份將資料分割成二個集合，在利用 $C(d_i \approx d_j)$ 可以計算出上述用戶端所請求的資料項將產生10個單位的競爭次數。在圖3.3-b中，廣播通道1負責廣播 $\langle d_1、d_2、d_3 \rangle$ ，通道2負責廣播 $\langle d_4、d_5、d_6 \rangle$ 。在使用圖3.3-a的分割結果進行廣播排程。假設所有的用戶端進入廣播通道為 $d_1$ 的時間點進入。在3.3-b的排程結果之下，對所有用戶端的請求，在廣播 $d_1$ 的時間點上將造成3個用戶端必須等待到下一個廣播週期才能取得另一個資料項。在廣播 $d_2$ 的時間點上，並無發生資料碰撞的情況。於廣播 $d_3$ 的時間點上，將發生1個用戶端的請求必須於下一個廣播週期才能取回另一資料項的情形。在圖3.4-a中，當不同的資料分割結果，對於競爭次數將會有不同的結果。從圖中利用

$C(d_i \approx d_j)$ 可以計算出發生6個單位的競爭次數。而圖3.4-b中，使用3.4-a的分割結果進行廣播排程。這樣的結果對於用戶端在存取資料時產生的競爭情況如下:在廣播 $d_1$ 的時間點上時，對所有的用戶端將產生1次的碰撞次數，於廣播 $d_4$ 的時間點上將產生1次的碰撞次數，在廣播 $d_5$ 的時間點上發生1次的碰撞次數。從圖3.3-a與3.4-a二種分割結果產生的碰撞次數來看，當較高的競爭次數發生時，對於在廣播排程上，將會有較高的機會發生資料碰撞的情況。而本文演算法的目的，即是在針對競爭次數的產生，利用演算法的處理來達到最小化的情況。這樣對於資料碰撞將可以有效的減少。對於用戶端在存取資料時，所花費的廣播週期次數將會降低。

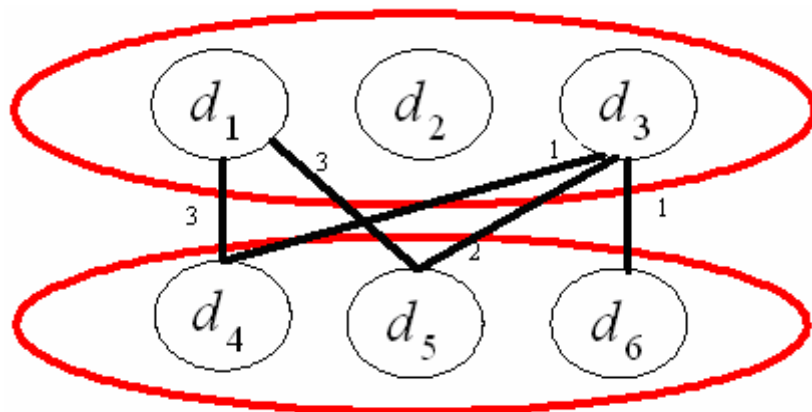


圖 3.3-a: 資料競爭關係圖

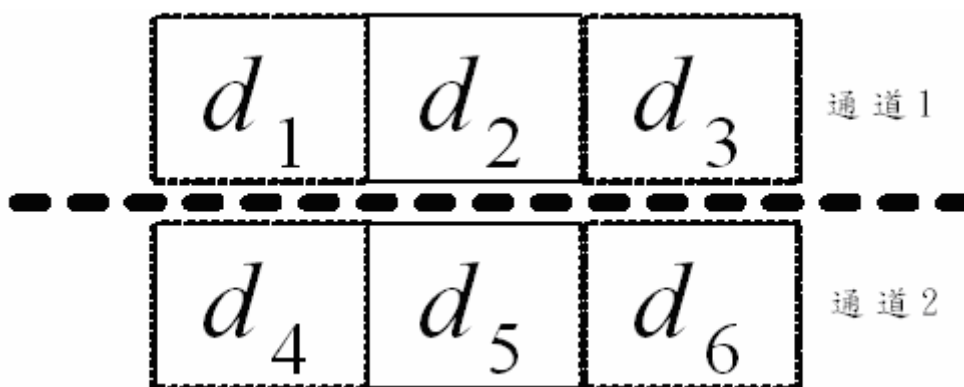


圖 3.3-b: 廣播排程結果

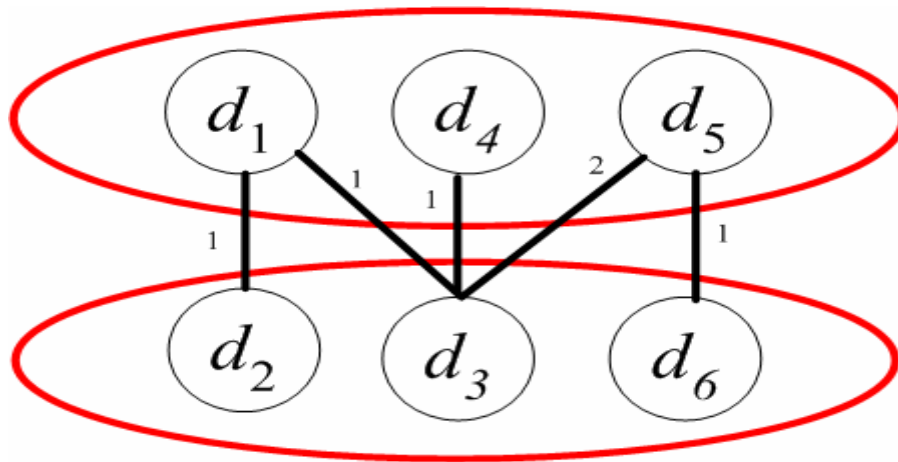


圖 3.4-a: 處理後之資料競爭關係圖

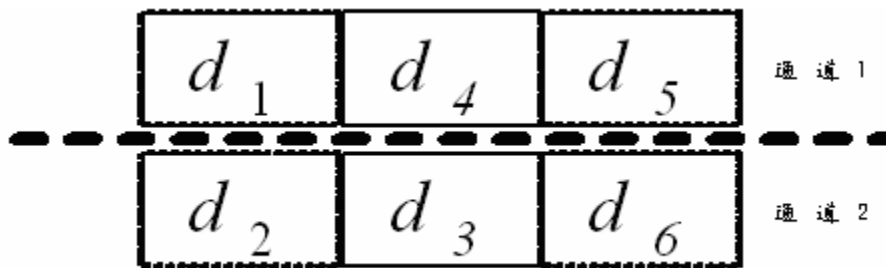


圖 3.4-b: 處理後之廣播排程結果

為了計算請求資料後所產生的資料競爭的次數，我們定義二個公式，分別是公式1與公式2。公式1為計算單一筆用戶端提出之請求所產生的競爭次數。公式2為衡量所有用戶端的請求所產生的競爭次數總數。以下舉例說明利用公式1與公式2計算競爭次數的方法。

當有二個用戶端 $q_1$ 與 $q_2$ 分別提出請求資料的服務時， $DS(q_1):\{d_1、d_3、d_5、d_7\}$ ， $DS(q_2):\{d_2、d_3、d_6、d_7\}$ 。而廣播伺服器廣播的資料項為 $\{d_1、d_2、d_3、d_4、d_5、d_6、d_7、d_8\}$ ，資料分割的結果如圖3.5所示。在圖中，每一個曲線表示成單一廣播通道所廣播的資料項，且資料項之間沒有廣播順序的關係。在利用公式1後可以計算出用戶端 $q_r$ 請求之資料項所產生的競爭次數數量，公式2考量到全體用戶端的請求，對於競爭次數產生的總數量。在公式1中， $C(d_i \approx d_j)$ 表示單一筆請求資料項之間所產生的競爭次數關係。舉例來說： $q_1$ 的請求可以產

生 $d_1 \approx d_3$ ， $d_1 \approx d_5$ ， $d_1 \approx d_7$ ， $d_3 \approx d_5$ ， $d_3 \approx d_7$ ， $d_5 \approx d_7$ 的資料存取關係。利用 $C(d_i \approx d_j)$ 去判斷請求關係中，二個資料項對於競爭次數產生的結果。當二個資料項於被分配於相同的通道時，碰撞次數為0，反之為1。在圖中，邊的權重值均為1，故 $q_1$ 的資料請求，將產生4個單位的競爭次數(實線部份)。公式2可以考量到全體用戶端的請求，對於競爭次數所產生的總次數(實線部份與虛線部份的加總)。

$$F(q_r) = \sum_{i=1}^{|DS(q_r)|-1} \sum_{j=i+1}^{|DS(q_r)|} C(d_i \approx d_j) \quad \text{公式 1}$$

$$\sum_{r=1}^{|Q|} F(q_r) \quad \text{公式 2}$$

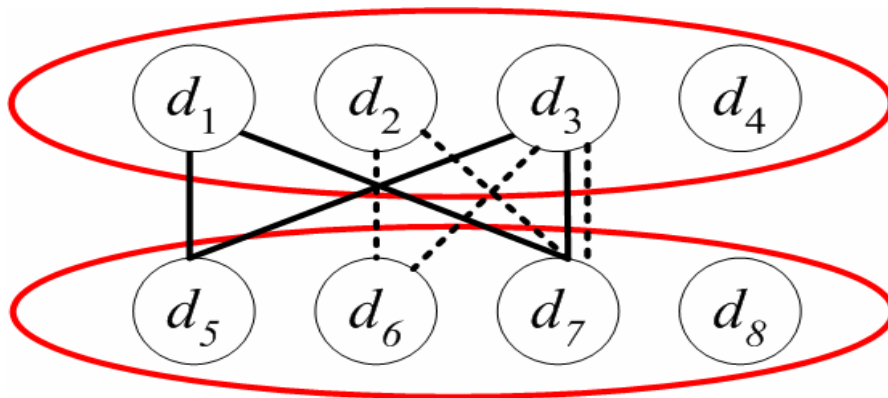


圖 3.5:用戶端請求資料所產生之資料競爭關係圖

#### 第四節 演算法目標

在第三節中，我們舉例說明資料競爭的問題且說明計算該值的方法。在本小節中，說明本文主要的目標，最小化資料競爭次數。為了達成這樣的目標。本文應用"二階分割演算法"[17]來達成該目標。

在公式1中，該公式表示計算單一筆用戶端的競爭次數。在處理

完一個用戶端後，必須再處理其它用戶端的請求。利用公式2即可累計所有用戶端提出請求的競爭次數。而本文的目標，即是利用本文所應用的演算法將公式2的值減少到最小的情況。而這樣的結果，在廣播排程上，對於用戶端所請求的資料在資料碰撞的次數上，將可以有效減少，對用戶端而言，這樣的結果在存取資料時，所增加的廣播週期的次數可以減少。

## 第四章、二階分割演算法

在本章中，將說明本文所使用的演算法，並說明演算法的基本想法與運算流程。

### 第一節 基本想法

在第 3 章中，描述本文所要解決的問題。在本節中，說明本文處理資料競爭的想法，利用圖形的方法，將單一個廣播資料對映到圖形上的一個頂點。由用戶端所請求的資料集合中，將二個資料項之間的存取關係，建立成圖形上的邊。在處理所有用戶端所請求的資料集合後，可以產生"資料存取關係圖"。並在圖形上建立虛擬線，來表示各個通道的廣播資料。而演算法的目的即是在讓各個通道上資料項之間的競爭次數透過演算法的處理達到最小的情況。以下使用範例說明用戶端請求之資料項所產生的資料存取關係圖。假設用戶端與請求之資料分別為： $DS(q_1):\{d_1, d_2\}$ ， $DS(q_2):\{d_2, d_3\}$ ， $DS(q_3):\{d_2, d_3\}$ ， $DS(q_4):\{d_1, d_4\}$ ， $DS(q_5):\{d_2, d_6\}$ ， $DS(q_6):\{d_1, d_5\}$ ， $DS(q_7):\{d_2, d_3\}$ 。利用上述請求的資料項，可以建立資料存取關係圖，如圖 4.1 所示。在圖形中， $V_1$ 表示成廣播資料項 $d_1$ 。虛線部份表示成各個通道所負責廣播的資料項。在圖中表示分割成二個通道，分別為通道 1 與通道 2。對於邊上的權重值表示成 $e_{ij}$ ，該值表示用戶端的請求對於二個資料項之間的存取次數，而圖形上的邊，在演算法處理的過程中，可以分為二種情況，分別為資料項的存取關係為相同通道與不同通道之別。當存取關係為相同通道時，稱為對內存取關係，當不同通道時，稱為對外存取關係。在處理完所有用戶端的請求後，將可以得到用戶端存取次數的加總。

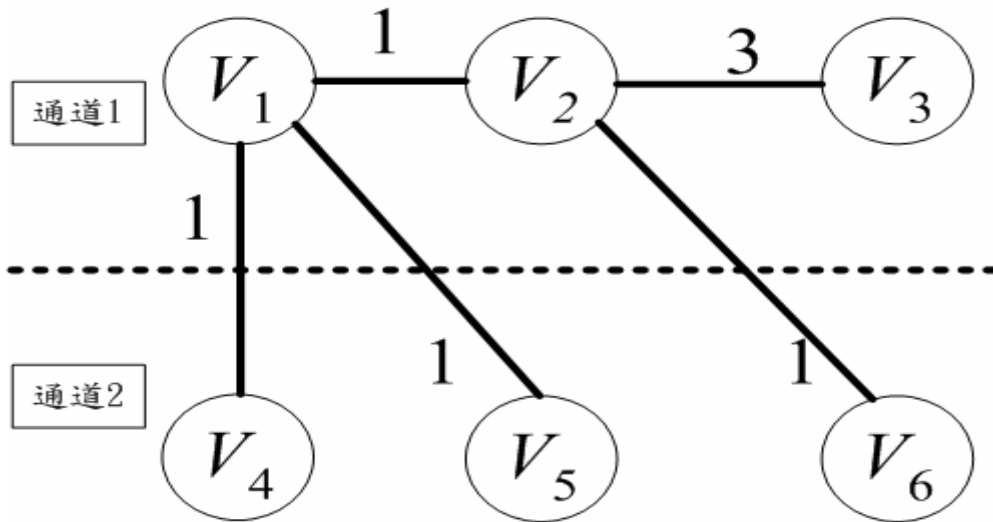


圖 4.1:資料存取關係圖

## 第二節 基本假設

在本小節中，將說明本文環境中的基本假設。從資料項的觀點來看，每一個資料項的大小均相同，所以伺服器在廣播資料時花費相同的時間在處理資料項。而用戶端在請求資料時，所花費的存取時間為一個時間單位。在圖形中對邊的表示為計算二個資料項之間的存取權重，當被存取一次時，則 $e_{ij}$ 值加1。從用戶端的觀點來看，在存取資料是沒有順序性的。也就是說當廣播資料出現用戶端請求的資料集合時，就取回該資料項。而請求的資料數量是屬於"多個資料"請求模式。於第三節中，將詳細說明本文所提出的演算法運算流程。



### 第三節 二階分割演算法

為了解決資料競爭的問題，本文應用[17]所提出在積體電路設計上所使用的方法，稱為F-M演算法，來處理在無線廣播中，多通道且多個資料請求的環境，所產生資料競爭的問題。在廣播的通道上設定為二個廣播通道。為了說明上的方便，以下定義演算法中相關的符號。如表 4.1 所示。

表 4.1:演算法相關符號

符號	說明
$Gain_{V_i}$	搬動 $V_i$ 對競爭次數的增減。
$RS(V_i)$	與 $V_i$ 有關聯的頂點集合。
$C[V_i]$	頂點 $V_i$ 被分配到之通道編號。

在表4.1中，我們定義了相關的符號於演算法的處理步驟並分別說明各個符號代表之意義。

$Gain_{V_i}$ 為搬動資料項 $V_i$ 後對競爭次數影響的結果，該值於演算法當中，占有相當重要的角色。 $RS(V_i)$ 是與 $V_i$ 有關連的頂點集合。當搬動 $V_i$ 時將與它相關聯的頂點的競爭次數也將被影響。該變數的目的為讓演算法中，可以更新到相關 $V_i$ 的 $Gain_{V_i}$ 值。 $C[V_i]$ 是頂點 $V_i$ 被分配到某個廣播通道的資料。該變數是決定如何計算 $Gain_{V_i}$ 的資訊。利用 $Gain_{V_i}$ 產生的競爭次數關係圖如圖4.2所示。在圖中方塊虛線部份表示搬動 $V_i$ 對競爭次數的影響。當與 $V_i$ 有關聯的頂點為對外存取關係時，表示搬動 $V_i$ 後對競爭次數為減少，若為對內存取關係時，表示搬動 $V_i$ 後對競爭次數為增加。以 $V_1$ 為例:在搬動 $V_1$ 後對競爭次數將減少1

個單位，計算的方式為判斷與 $V_1$ 有關聯的頂點集合與 $V_1$ 的存取權重關係 $e_{ij}$ 。以 $V_1$ 為例:與它相關聯的資料項為: $\{V_2、V_4、V_5\}$ ，而 $\{V_4、V_5\}$ 與 $V_1$ 為對外存取關係，對競爭次數的影響為減少2個單位的競爭次數。而 $V_2$ 與 $V_1$ 為對內存取關係，對競爭次數的影響為增加1個單位的競爭次數。累計 $Gain_{V_1}$ 後該值為1。表示搬動 $V_1$ 後對競爭次數將減少1個單位。而搬動 $V_3$ 後將增加3個單位的競爭次數。其它頂點的競爭次數，同理可推。

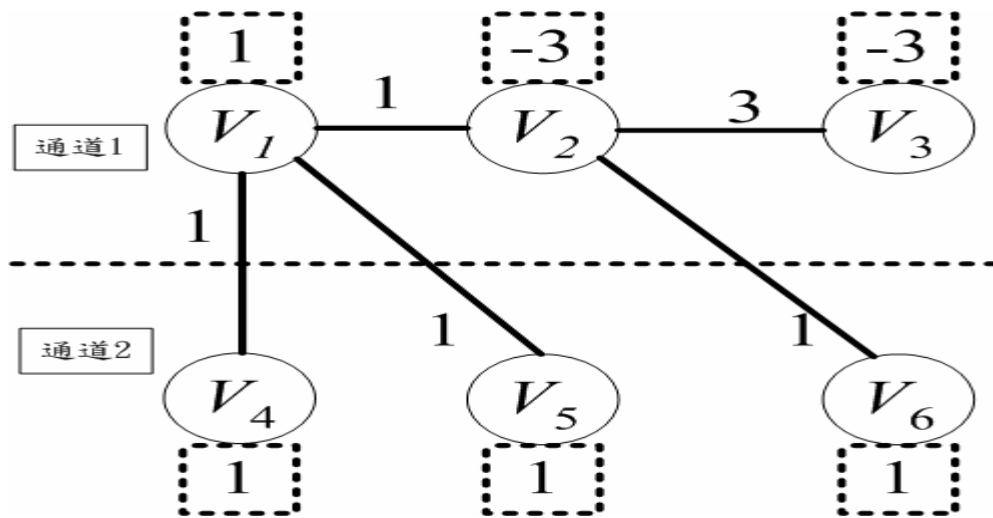


圖 4.2: 資料競爭次數關係圖

而 $RS(V_i)$ 表示在搬移頂點 $V_i$ 後，將影響與 $V_i$ 有關聯的頂點集合的競爭次數。在圖4.2中當搬移 $V_1$ 之後，將影響到資料集合 $\{V_2、V_4、V_5\}$ 的 $Gain_{V_i}$ ，更新 $Gain_{V_1}$ 結果如圖4.3所示。以 $V_4$ 為例:在搬動 $V_1$ 後， $V_1$ 與 $V_4$ 的存取關係將改變。對於有關聯的頂點之 $Gain_{V_i}$ 的計算方法為，當為對內存取時， $Gain_{V_i}$ 值將增加2倍 $e_{ij}$ 的競爭次數。當為對外存取時， $Gain_{V_i}$ 值將減少2倍的 $e_{ij}$ 的競爭次數。 $V_4$ 為例，在搬動 $V_1$ 後， $V_1$ 與 $V_4$ 的存取關係更改為對內存取關係，對於 $Gain_{V_4}$ 將更新為-1。對於有關連的 $V_i$ ，在更新時將利用相同的方式去計算新的 $Gain_{V_i}$ 。

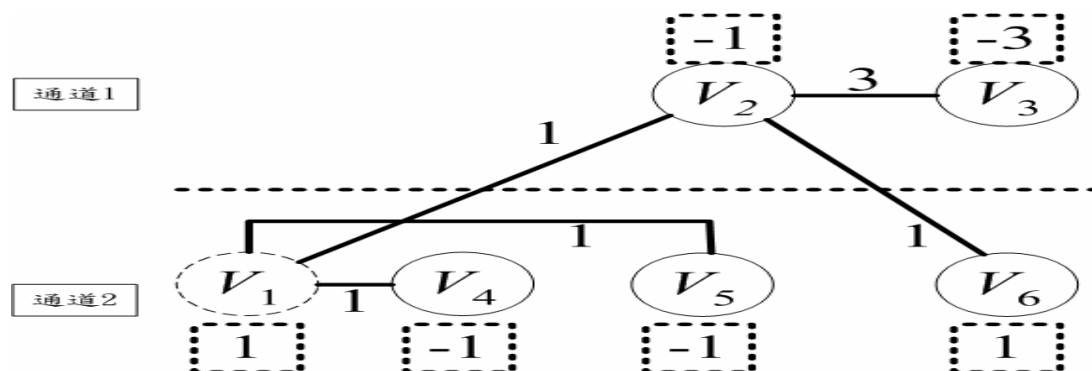


圖 4.3:搬動頂點對資料競爭次數影響關係圖

在圖4.3中，當頂點以虛線表示時，代表該頂點於當時情況的競爭次數中，搬動該頂點將可以減少最多的競爭次數。而該頂點之資料競爭次數將不給予做更新的動作。而 $C[V_i]$ 的值，由原本的通道1更改為通道2。該值為對 $RS(V_i)$ 相關的頂點在更新 $Gain_{V_i}$ 的重要資訊。

以下說明整個演算法對資料項的處理步驟，為了說明上的方便，於圖4.4中列出演算法的虛擬碼(Pseudo Code)，並在每行中增加行號來說明。並在表4.2當中，列出演算法運算過程中，相關的變數並說明該用途。

表 4.2:演算法中相關變數

符號	說明
$L[V_i]$	記錄 $V_i$ 是否搬移過。
$array\_gain[V_i]$	記錄搬移 $V_i$ 後所降低的競爭次數。
$order[V_i]$	儲存搬移 $V_i$ 的順序。
$max\_gain$	搬移到 $V_x$ 後可減少最多的競爭次數。

在表4.2中，說明演算法中，相關變數於演算法中的用途。 $L[V_i]$ 記錄廣播資料項在演算法的處理過程中，該資料項是否已被搬動。

$array\_gain[V_i]$ 該陣列記錄每次搬動 $V_i$ 對於競爭次數的增減情況，該變數是為了找到搬動到 $V_x$ 後可以減少最多的競爭次數( $max\_gain$ )。而 $order[V_i]$ 記錄每次演算法的處理所搬動的頂點順序，此陣列的目的為在演算法的處理過程中，在找到減少最多競爭次數的頂點 $V_x$ 後，必須將 $order$ 陣列中 $V_x$ 之後所搬移的頂點還原到搬動前的初值。並進行下一步的演算流程。

在圖4.4當中，行1、2為初始化廣播通道資料的結果，演算法處理的廣播通道為二個，故先將 $D$ 中一半的資料 $C[V_i]=0$ ，一半之 $C[V_i]=1$ 。 $L$ 陣列儲存 $V_i$ 是否被搬移的值，在初始化時，首先將該值設定為 $Unlocked$ 。第4行表示先計算所有 $V_i$ 的 $Gain\_V_i$ 。行5至8表示：從所有的頂點中找出未被標記為 $Unlocked$ 且為最大 $Gain\_V_i$ 的頂點，當搬動 $V_i$ 後必須判斷搬動 $V_i$ 後是否對資料平衡造成影響。若無影響則將該頂點的 $Gain\_V_i$ 依順序將它存在於 $array\_gain$ 的陣列之中，並將 $V_i$ 存入 $order$ 陣列當中。將 $V_i$ 屬於通道編號的值變更，並將 $V_i$ 設定為 $Lock$ ，表示該頂點已被搬移過。行10到13是更新與 $V_i$ 有關聯的頂點集合。更新的原因為在搬動 $V_i$ 時將影響與它有關聯的競爭次數。行10表示所有與 $V_i$ 有關聯的頂點集合做更新的動作但要排除 $V_k$ 為 $Unlocked$ 的頂點，在該集合中必須將 $V_i$ 的 $C[V_i]$ 與 $RS(V_i)$ 中的 $V_k$ 的 $C[V_k]$ 做比較。當二者相同時，則執行第12行否則執行第13行。第14行是將 $temp\_max$ 、 $max\_point$ 、 $p$ 的值歸零。該三個變數的用途為找出搬動所有的 $V_i$ 後，在搬動第 $p$ 個頂點時，所減少的競爭次數達到最多的情況。行15到19為評估在搬動的頂點中於第 $max\_point$ 個的頂點在搬動後可以減少最多的競爭次數。在行16中計算步驟是根據 $array\_gain$ 的順序。依順序取出 $array\_gain$ 中的值累加到 $temp\_max$ 。行17在每次找到最大的 $temp\_max$ 時，則執行第18與19。行19中的 $max\_point$ 即為搬動到第 $p$

個頂點後可以減少最多的競爭次數。當在處理完 $array\_gain$ 中的元素後，當 $max\_gain$ 大於0時，表示搬動第 $p$ 個之前的頂點對於碰撞次數的減少將有幫助，故將第 $p+1$ 至 $m$ 個頂點的 $C[V_i]$ 還原到搬動時的初值，但 $order$ 陣列中的第1個到 $p$ 個之頂點不變更。再將所有頂點的狀態更改為 $Unlocked$ 。在第23行表示成在 $max\_gain$ 的值為零時，表示搬動第1個到第 $p$ 個的資料項對碰撞次數並無影響所以停止搬移的動作。

```

1  for  $k=1$  to  $m/2$  {  $C[V_k]=0; L[V_k]=Unlocked$  } //initial
2  for  $j=(m/2) +1$  to  $m$  {  $C[V_j]=1; L[V_j]=Unlocked$  }
3  Repeat
4  Calculate the  $Gain_{V_i}; \forall V_i \in D$ 
5  For  $i=1$  to  $m$ 
6  Find  $V_i$  with Max  $Gain_{V_i}$  and  $V_i$  is Unlocked
   and moving  $V_i$  the balance is true
7  array_gain[ $i$ ]= $Gain_{V_i}$  ; order[ $i$ ]= $V_i$ 
8  If  $C[V_i]=1$  then  $C[V_i]=0$  ; If  $C[V_i]=0$  then  $C[V_i]=1$ 
9  Lock the  $V_i$ 
10  For all  $V_k \in RS(V_i)$  and  $L[V_k]$  is Unlocked
11  If  $C[V_i]=C[V_k]$  then
12   $Gain_{V_k}=Gain_{V_k} + 2*e_{ik}$ 
13  else  $Gain_{V_k}=Gain_{V_k} - 2*e_{ik}$ 
14  temp_max=0 ; max_point =0 ; p=0 ;
15  For p=1 to m
16  temp_max=temp_max+array_gain[p]
17  if(temp_max>max_gain) then
18  max_gain=temp_max
19  max_point=p
20  If max_gain> 0 then
21  For  $i=p+1$  to  $m$  if  $C[V_i]=0$  then  $C[V_i]=1$  else  $C[V_i]=0$ 
22  Free all Locked  $L[V_i]$ 
23 Until max_gain=0

```

圖 4.4: 二階分割演算法之虛擬碼

#### 第四節 演算法運算流程範例

在本小節中，利用實例並搭配虛擬碼來說明整個演算法的運算流程與各個變數的數值。用戶端與請求資料項如下所示。 $DS(q_1):\{d_1, d_2, d_4\}$ ， $DS(q_2):\{d_2, d_3\}$ ， $DS(q_3):\{d_2, d_3\}$ ， $DS(q_4):\{d_1, d_4, d_5\}$ ， $DS(q_4):\{d_2, d_6\}$ ， $DS(q_5):\{d_1, d_5\}$ ， $DS(q_6):\{d_2, d_3, d_5\}$ 。

首先對資料的請求建立資料存取關係圖如圖 4.5 所示。並對所有的頂點進行初始化的動作行 1 到行 2： $V_1$ 到 $V_3$ 的 $C[V_1]$ 至 $C[V_3]$ 為 0， $V_4$ 到 $V_6$ 的 $C[V_4]$ 至 $C[V_6]$ 為 1，L 陣列初始化為 *Unlocked*，L 陣列記錄頂點是否被搬移的資訊。虛線部份將頂點分割成二個通道，分別是通道 0 與通道 1。各個通道所廣播的資料如圖所示。

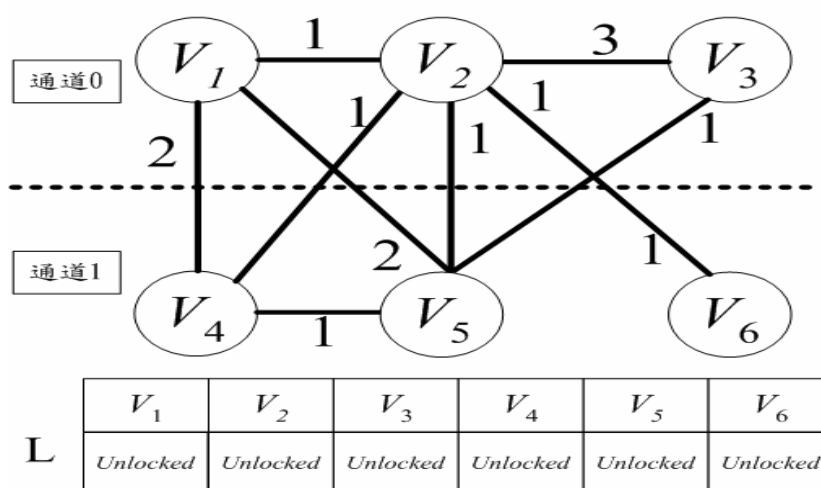


圖 4.5: 初始化廣播資料

當初始化完成後，必須執行虛擬碼中的行 4 到 11。行 4 表示對所有的頂點計算競爭次數，計算的結果如圖 4.6 中方塊虛線所示。方塊虛線的部份，表示搬動頂點後，該頂點對於競爭次數的影響。在搬動 $V_1$ 後對資料競爭次數將減少 3 個單位。而搬動 $V_3$ 後將增加 2 個單位的競爭次數。其它頂點的競爭次數，同理可推。

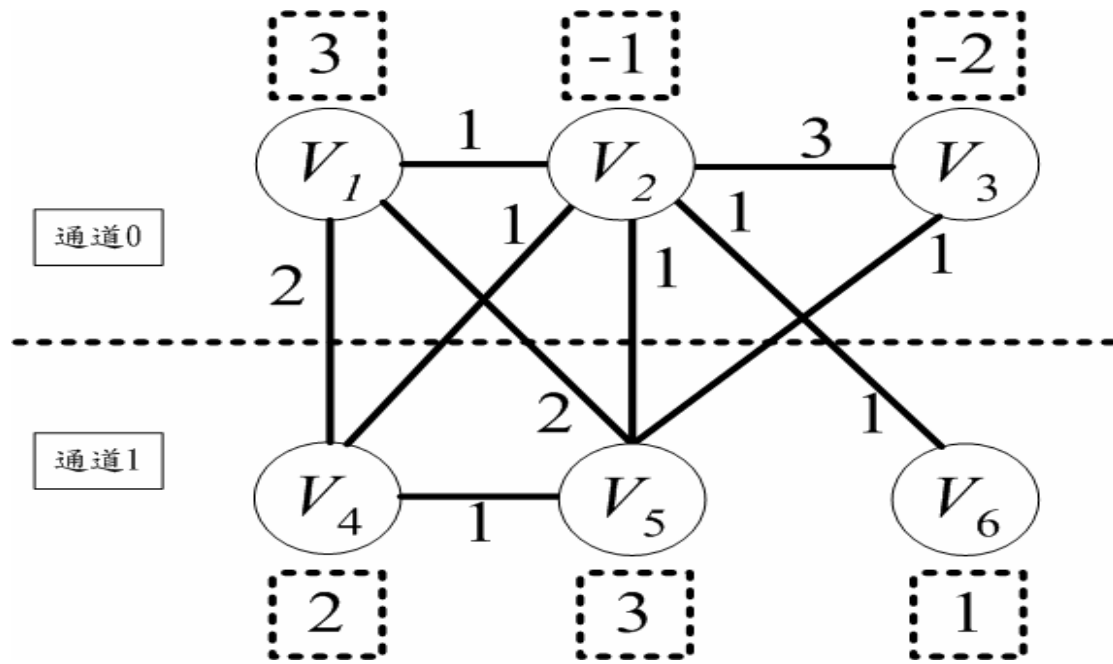
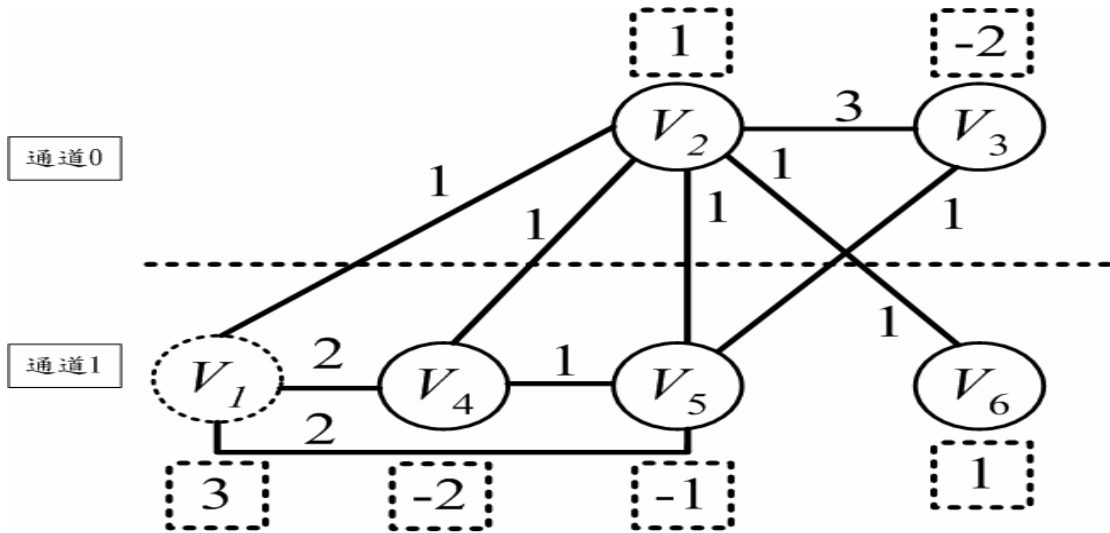


圖 4.6:初始化資料競爭次數

行 6 表示找出搬動  $V_i$  後可以減少最多競爭次數的頂點，且該頂點是在 L 陣列中，是被標記為 *Unlocked*，且搬移該頂點後沒有破壞廣播資料的平衡率。結果顯示於圖 4.7 當中。在圖形中，虛線之頂點表示在該情況之下，搬動該頂點將可以減少最多的資料競爭次數的頂點，在該狀態之下，將搬動  $V_1$ 。array\_gain[1]:1 表示搬第 1 次的頂點時，對於競爭次數增減的情況，在該次的搬動對競爭次數有減少一個單位。order[1]: $V_1$  表示搬動第 1 次時，搬動的順序為  $V_1$ 。對搬動的頂點，必須更改該頂點於通道的位置，在該次搬移中，頂點由通道 0 被搬移到通道 1。對  $RS(V_1)$  而言，與它有存取關係的頂點有  $\{V_2、V_4、V_5\}$ ，對於該集合內的頂點必須判斷該頂點是否為 *Unlocked*，當為 *Unlocked* 時將更新該頂點的競爭次數，否則省略不處理。在處理完相關變數後，進行第二次的資料搬移，在考慮平衡率的情況下且找出最大的  $Gain_{V_i}$  為  $V_6$ ，相關流程與變數訊息顯示於圖 4.7 至 4.12 之中。





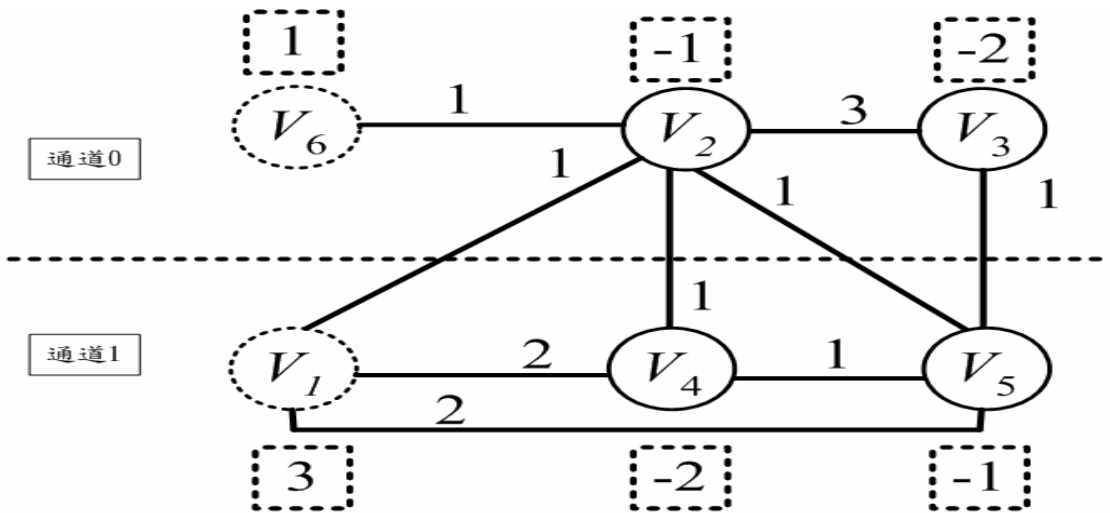
**L**

$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$V_6$
<i>Lock</i>	<i>Unlocked</i>	<i>Unlocked</i>	<i>Unlocked</i>	<i>Unlocked</i>	<i>Unlocked</i>

array\_gain[1]:3     $C[V_1]=1$

order[1]:  $V_1$      $RS[V_1]=\{V_2, V_4, V_5\}$

圖 4.7:搬動最大Gain\_ $V_i$ 且符合其它條件之 $V_i:V_1$



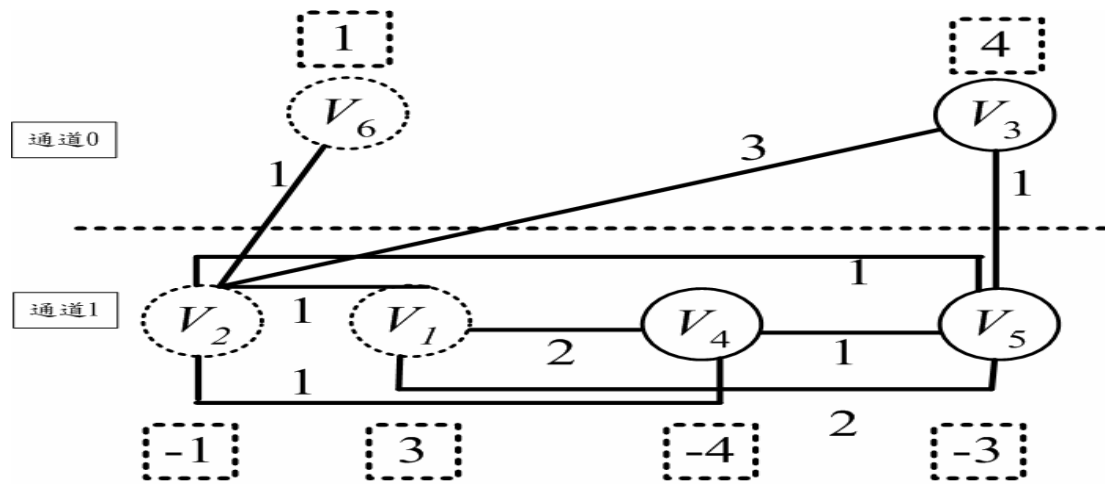
**L**

$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$V_6$
<i>Lock</i>	<i>Unlocked</i>	<i>Unlocked</i>	<i>Unlocked</i>	<i>Unlocked</i>	<i>Lock</i>

array\_gain[2]:1     $C[V_6]=0$

order[2]:  $V_6$      $RS[V_6]=\{V_2\}$

圖 4.8:搬動最大Gain\_ $V_i$ 且符合其它條件之 $V_i:V_6$



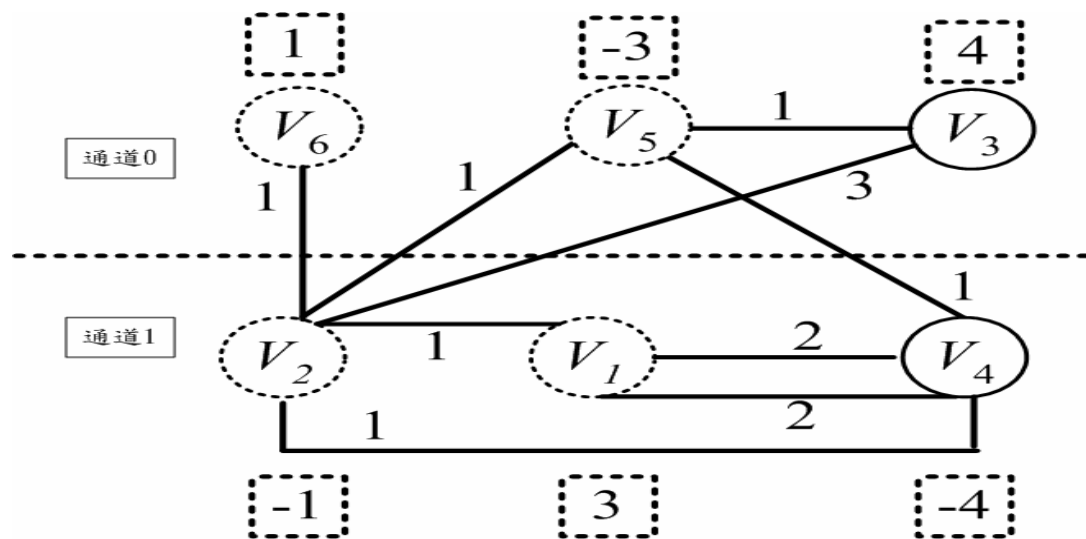
**L**

$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$V_6$
<i>Lock</i>	<i>Lock</i>	<i>Unlocked</i>	<i>Unlocked</i>	<i>Unlocked</i>	<i>Lock</i>

array\_gain[3]: -1     $C[V_2]=1$

order[3]:  $V_2$      $RS[V_2]=\{V_3, V_4, V_5\}$

圖 4.9: 搬動最大  $Gain_{V_i}$  且符合其它條件之  $V_i: V_2$



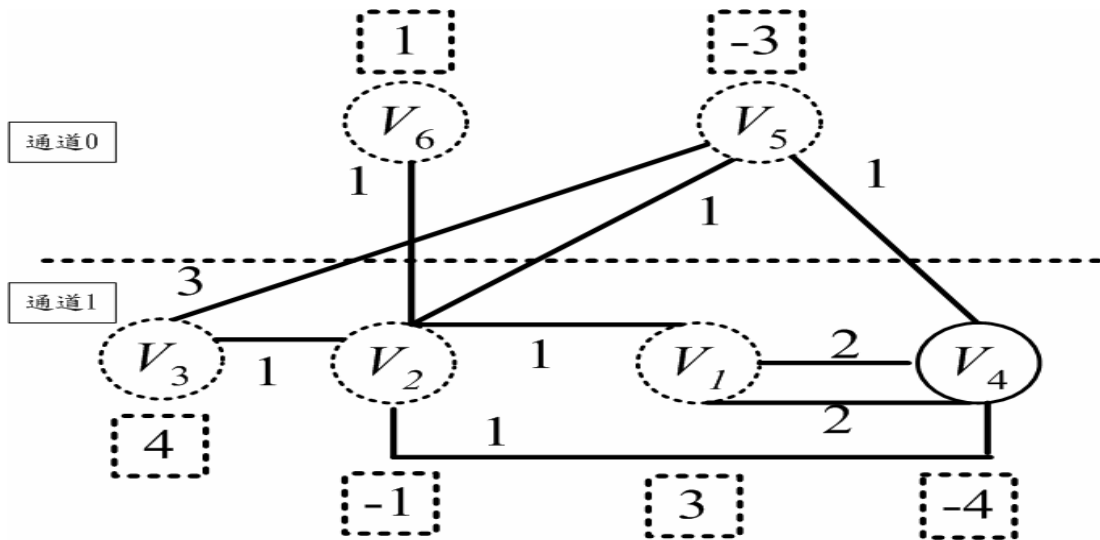
**L**

$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$V_6$
<i>Lock</i>	<i>Lock</i>	<i>Unlocked</i>	<i>Unlocked</i>	<i>Lock</i>	<i>Lock</i>

array\_gain[4]: -3     $C[V_5]=0$

order[4]:  $V_5$      $RS[V_5]=\{V_3, V_4\}$

圖 4.10: 搬動最大  $Gain_{V_i}$  且符合其它條件之  $V_i: V_5$



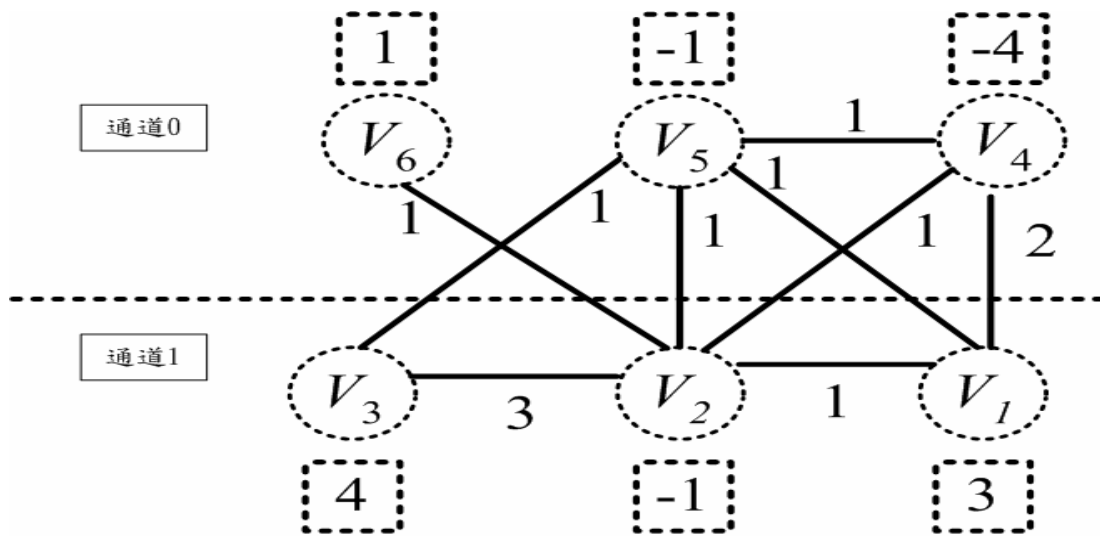
L

$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$V_6$
Lock	Lock	Lock	Unlocked	Lock	Lock

array\_gain[5]:4       $C[V_3]=1$

order[5]:  $V_3$        $RS[V_3]=\{\}$

圖 4.11:搬動最大Gain\_ $V_i$ 且符合其它條件之 $V_i:V_3$



L

$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$V_6$
Lock	Lock	Lock	Lock	Lock	Lock

array\_gain[6]:-4       $C[V_4]=0$

order[6]:  $V_4$        $RS[V_4]=\{\}$

圖 4.12:搬動最大Gain\_ $V_i$ 且符合其它條件之 $V_i:V_4$

在處理完行5至13後，將產生array\_gain與order陣列的資訊。而行15至19即是要找出在搬動各個頂點後，將減少最多的競爭次數。如p所示。在圖4.13中，array\_gain記錄每次搬動 $V_i$ 後，對競爭次數增減的情形。而order陣列記錄每次搬動 $V_i$ 的順序。利用此二個陣列元素及temp\_max與max\_gain二個變數，可以找出搬動第p個 $V_i$ 後將可以減少最多的競爭次數。而temp\_max累加array\_gain陣列中的值，當temp\_max大於max\_gain時，則p值將記錄所累加的 $V_i$ 。從範例中可以找出搬動的頂點從order陣列中，在搬移到第5個頂點後，將可以減少最多的競爭次數。在第20至22行判斷當max\_gain大於0時表示，搬移的頂點均可以減少競爭次數，故重覆上述的流程。直到max\_gain為零時，則中斷程式。若max\_gain大於零時，行21至22便將order中的值將p+1後的頂點的 $V[d_i]$ 值還原到搬移之前。而1到p的頂點將不變更。因為在搬移到第p個頂點後對競爭次數的減少，將有幫助。行22將所有資料的在L陣列中的值更改為Unlocked。並繼續執行第5到13的動作，直到max\_gain為0時，則中斷程式。

```

array gain [3, 1,-1,-3, 4,-4]  order [V1, V6, V2, V5, V3, V4]

temp_max=3  max_gain=3  P=1
temp_max=4  max_gain=3  P=2
temp_max=3
temp_max=0
temp_max=4  max_gain=4  P=5
temp_max=0

```

圖4.13:演算法處理後，相關變數資訊

## 第五章、實驗結果

在本章中，說明本文實驗的環境實驗參數與實驗之結果。

### 第一節 實驗環境說明

本文使用之演算法二階分割演算法，利用 JAVA 程式語言進行模擬。實驗電腦硬體設備為 P4 2.2，記憶體為 512 RAM。在廣播伺服器使用的廣播通道設定為 2 個廣播通道來進行模擬。對於資料項的大小上，假設每一個資料項為相同大小。在資料讀取的順序上為無順序之方式。用戶端請求的資料分佈，對常態分佈(Normal Distribution)與均勻分佈(Uniform Distribution)二種來進行實驗。

### 第二節 模擬參數

在該節當中，將列出相關的系統模擬參數來進行實驗。主要參數顯示在表 5.1 當中。

表 5.1:系統模擬參數

資料項數量	100~500
用戶端數量	100~600
選擇率	0.02%~0.06%
平衡率 ( $B$ )	5%

在表5.1當中，資料項數量為廣播伺服器所廣播的所有資料項總數。用戶端數量為向廣播伺服器提出請求資料的用戶端總數。選擇率為一個用戶端請求資料項最大的數量。舉例來說，當伺服器廣播之資

料項為100個單位，在選擇率為2%的情況之下，每一個用戶端最多可以請求的資料項為2個資料單位。平衡率是各個廣播通道之間廣播資料的差距。舉例來說，單一廣播通道所廣播的資料項數量為100個單位，在平衡率為5%時，各個廣播資料量的數量將控制在95~105範圍內。當平衡率增加時，對於廣播排程的結果將造成愈不平衡的結果。

### 第三節 均勻分佈

在實驗當中，利用上述所列之參數來進行模擬。而均勻分佈表示成每一筆請求被存取的頻率均相同。在均勻分佈中，該值設定為1。

在圖5.1與5.2中，初始化表示將所有的廣播資料項平均分割成二個集合所產生的競爭次數。二階分割即為本文所使用之演算法處理後對於競爭次數產生的結果。

#### 一、廣播資料項數量對競爭次數之影響

在廣播資料項數量對競爭次數的影響上，用戶端個數為200個用戶端。平衡率為5%。選擇率為2%。廣播資料項數量介於100~500之間。實驗結果如圖5.1所示，當在相同的用戶端數量之下，當廣播資料數量增加時，競爭次數在初始化化呈現遞增的趨勢。而二階分割法可以有效的降低全體用戶端請求資料的競爭次數。在表5.2中，列出各種廣播資料項數量下，初始化與二階分割對於競爭次數在利用演算法處理後，對於競爭次數減少的百分比，從表中可以看出，在各種廣播數量的情況之下，對於競爭次數的影響，平均減少49%。

表5.2:均勻分佈下:資料項數量對競爭次數之影響

處理方式 \ 資料數量	初始化	二階分割	效能分析
100	47	12	74%
200	179	72	60%
300	367	232	37%
400	542	323	40%
500	780	500	36%
平均減少%			49%

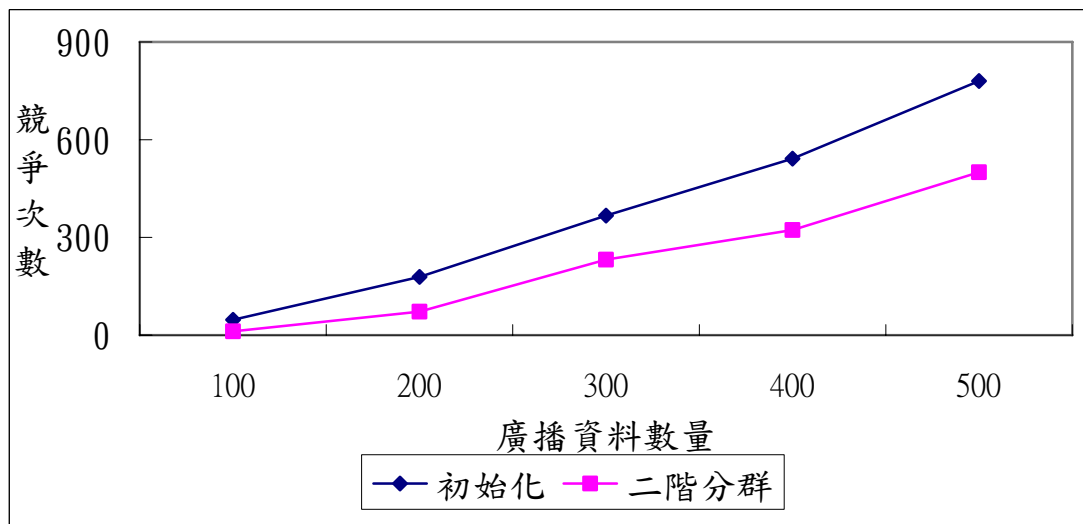


圖 5.1:均勻分佈:廣播資料項對競爭次數之影響

## 二、選擇率對競爭次數之影響

在本次的實驗將於:平衡率為5%。廣播資料數量為200個。用戶端數量為300個。選擇率為0.02~0.06的情況之下進行模擬。將對於用戶端提出請求之資料項數量進行評估。實驗結果如圖5.2所示:在相同的廣播數量與用戶端數量之下,當用戶端所請求的資料量增加時。在初始化中,資料競爭次數快速的增加。這表示當用戶端請求的資料量增

加時，在演算法的計算當中，二個資料項之間被存取的次數相對會增加。所以競爭次數相對增加。而本文所提的演算法在競爭次數增加時，方能有效的減少競爭次數的產生。在該次的實驗當中，對於競爭次數在不同的選擇率下，減少達到44%，結果如表5.3所示。在二種資料分佈的情況下對於競爭次數的減少平均達到46%。由二種資料分佈的情況下證明，本文所提出的二階分割演算法對於競爭次數的減少是有效的，這對用戶端在存取資料的效率上，將大大的提昇。

表5.3:均勻分佈下:選擇率對競爭次數之影響

處理方式 選擇率	初始化	二階分割	效能分析
0.02	275	147	47%
0.03	649	279	57%
0.04	980	426	57%
0.05	1398	893	36%
0.06	1872	1380	26%
平均減少%			44%



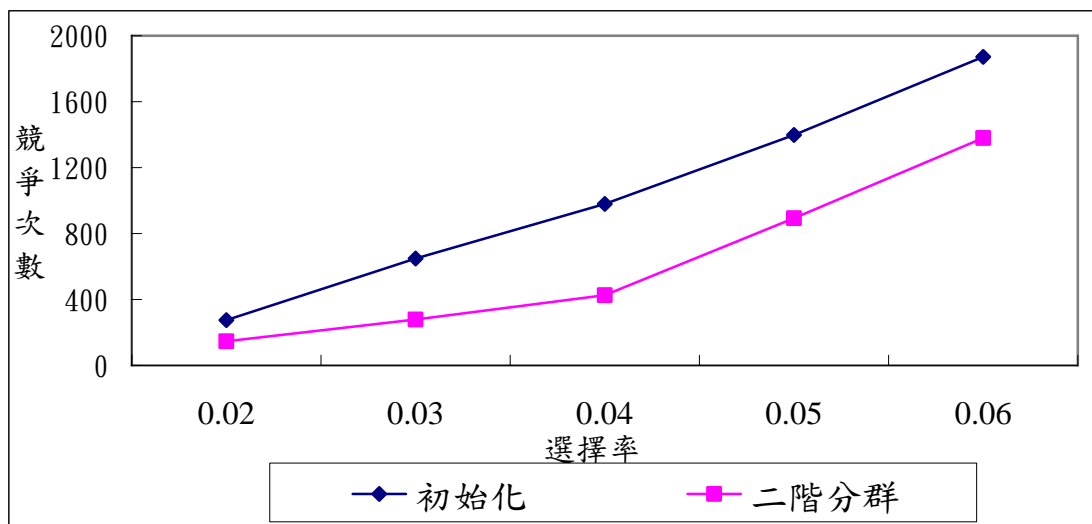


圖 5.2:均勻分佈:選擇率對競爭次數之影響

#### 第四節 常態分佈

在本次的實驗當中，利用常態分佈的資料來進行實驗。而常態分佈表示成每一筆請求被存取的頻率均不相同。如此會產生熱門資料與冷門資料之差異。在實驗中，資料項被存取的頻率將由系統隨機產生。

##### 一、資料項數量對競爭次數之影響

在常態分配的模擬參數上，以用戶端數量為200個。平衡率為5%，選擇率為2%，廣播資料數量介於100~500之間。來進行實驗。

在圖5.3中，對於初始化的表現上，因為資料項被存取的次數，考量到各個資料被存取的頻率均不相同，所以產生了熱門資料項與冷門資料項的差距。這對資料項被存取的次數上，將會有較大的影響。由圖中可以看出，在廣播資料量增加時，競爭次數隨著增加。而初始化曲線所呈現高低的差距，主要因為當熱門資料於初始化時被分配於相同的通道時，對於競爭次數會有減少的現象，反之呈現增加的現象。實驗顯示，本文所提之演算法對於廣播資料量增加時，對於競爭次數的減少將更明顯。

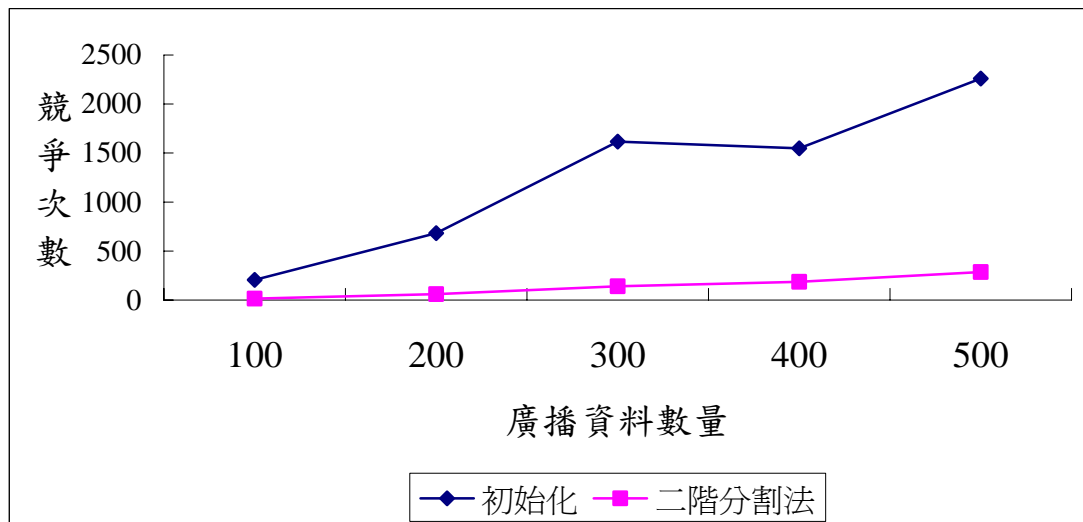


圖5.3:常態分佈:廣播資料項對競爭次數之影響

## 二、選擇率對競爭次數之影響

在本次的實驗將在資料分佈為常態分佈，平衡率為5%，廣播資料數量為200個，用戶端數量為300個，選擇率為0.02%~0.06%的情況之下進行模擬。並對於用戶端提出請求之資料項數量進行評估。實驗結果顯示，在相同的廣播數量與用戶端數量之下，當用戶端所請求的資料量增加時，在初始化當中，資料競爭次數快速的增加。這表示當用戶端請求的資料量增加時，在演算法的計算當中，二個資料被存取次數相對會增加。所以競爭次數相對的增加。而本文所提的演算法在競爭次數增加時，也能有效的減少競爭次數。

從二種資料分佈的實驗結果顯示，本文所提之二階分割演算法於二種資料分佈的情況之下，都可以有效的減少競爭次數，且當廣播資料數量與選擇率增加時，對於競爭次數的減少會有更佳表現。這樣的結果對於不同類型的資料分佈，當用戶端提出資料請求時，對於廣播週期次數的減少會是有助益的。

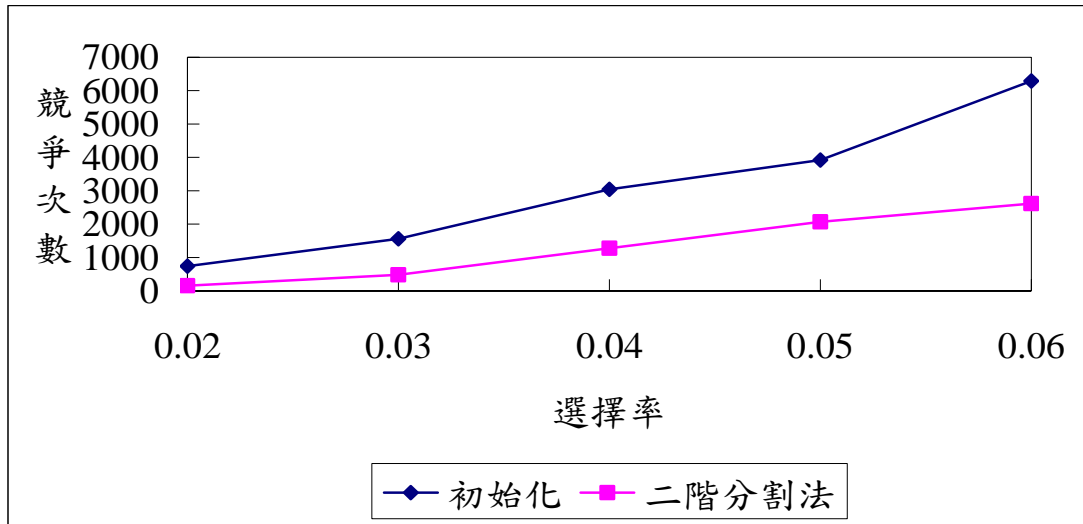


圖 5.4:常態分佈:選擇率對競爭次數之影響

## 第六章、結論與未來方向

廣播在無線環境已被廣泛的應用，在本文當中，我們整理無線廣播上相關的環境與學者所提出的方法。在多廣播通道上，當用戶端所請求的資料項為"多個資料"的請求時，將會產生本文所提出的資料競爭的問題。對所有的用戶端而言，當該情況增加時，在廣播排程上，將會增加用戶端存取資料時所花費的廣播週期次數。這對於用戶端請求資料的存取時間勢必會有增加的現象。而本文所使用的演算法，即是針對這樣的問題進行解決。在文中我們應用"二階分割演算法"來處理這樣的問題，並應用於二個廣播通道上。對於效能評估，實驗結果顯示我們的方法可以有效的減少資料競爭的產生。這樣的結果對於伺服器在對資料進行廣播排程時，將會有效的減少資料碰撞的發生，這對於全體用戶端在存取資料的效率上將會提昇。在未來的研究方向將利用  $K$  階分割演算法，應用在多個廣播通道上。並評估增加通道時對全體用戶端在碰撞次數的影響。

## 參考文獻

- [1] S. Acharya, M. Franklin, and S. Zdonik, "Dissemination Based Data Delivery Using Broadcast Disks," *IEEE Personal Communications*, Vol. 2, pp.50-60, 1995.
- [2] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communication Environments," *Proc. ACM SIGMOD*, pp. 199-210, 1995.
- [3] S. Acharya, M. Franklin, and S. Zdonik, "Prefetching from a Broadcast Disk," *Proc. IEEE International Conference on Data Engineering*, pp. 276-285, 1996.
- [4] S. Acharya, M. Franklin, and S. Zdonik, "Balancing Push and Pull for Data Broadcast," *Proc. of ACM SIGMOD*, pp. 183–194, 1997.
- [5] S. Acharya and S. Muthukrishnan, "Scheduling On-Demand Broadcasts: New Metrics and Algorithms," *MOBICOM*, pp. 43-54, 1998.
- [6] D. Aksoy, Leung and S.-F Mason, "Pull vs. Push: A Quantitative Comparison for Data Broadcast," *IEEE Global Telecommunications Conference GLOBECOM '04*, Vol. 3, pp. 1464 - 1468, 2004.
- [7] D. Aksoy and M. Franklin, "R×W: A Scheduling Approach for Large-Scale On-Demand Data Broadcast," *Transactions on Networking IEEE/ACM*, Vol. 7, pp. 846 - 860, 1999.
- [8] A. A Bertossi, M. C Pinotti, S. Ramaprasad, R. Rizzi, and M.V.S Shashank "Optimal Multi-Channel Data Allocation with Flat Broadcast Per Channel," *Proceedings on 18<sup>th</sup> International Parallel and Distributed Processing Symposium*, pp. 18, 2004.
- [9] G. C. Buttazzo and F. Sensini, "Optimal Deadline Assignment for Scheduling Soft Aperiodic Tasks in Hard Real-Time Environments,"

- IEEE Transactions on Computers*, Vol. 48, pp. 1035 - 1052, 1999.
- [10] Y. I Chang and W. H Hsieh, "An Efficient Scheduling Method for Query-Set-Based Broadcasting in Mobile Environments," *Proceedings on 24th International Distributed Computing Systems Workshops*, pp. 478 - 483, 2004.
- [11] J. Cho, S. Oh, Jaemyoung, J. Kim, H. Ho Lee, and J. Lee, "Neighbor Caching in Multi-Hop Wireless Ad Hoc Networks," *IEEE Communications Letters*, Vol. 7, pp. 525 – 527, 2003.
- [12] C. Y Chow, H. V Leong, and A. Chan, "Cache Signatures for Peer-to-Peer Cooperative Caching in Mobile Environments," *International Conference on Advanced Information Networking and Applications AINA 2004 18<sup>th</sup>*, pp. 196 - 101, 2004.
- [13] Y. D. Chung and M. H. Kim, "Effective Data Placement for Wireless Broadcast," *Distributed and Parallel Databases*, vol. 9, pp. 133-150, 2001.
- [14] O. E Demir, and D. Aksoy, "Energy-efficient broadcast-based event Update Dissemination," *IEEE International Conference on Performance Computing and Communications*, pp. 477-482, 2004.
- [15] Q. Fang, S. V. Vrbsky, Y. Dang, and W. Ni, "A Pull-Based Broadcast Algorithm that Considers Timing Constraints," *Proceedings on International Conference Parallel Processing ICPP 2004 Workshops*, pp. 46 – 53, 2004.
- [16] J. L Huang and M. S Chen, "Dependent Data Broadcasting for Unordered Queries in a Multiple Channel Mobile Environment," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, pp. 1143 - 1156, 2004.

- [17] C. M. Fiduccia and R. M. Mattheyses, "A Linear-Time Heuristic for Improving Network Partition," *Proc. Design Automation Con*, pp. 175-181, 1982.
- [18] G. Herman, G. Gopal, K.C. Lee, and A. Weinrib, "The Databycle Architecture for Very High Throughput Database Systems," *In Proceedings of the 1987 ACM SIGMOD International Conference on Management of Data*, pp. 97-103, 1987.
- [19] C. H. Hsu, G. Lee, and A.L.P. Chen, "Index and Data Allocation on Multiple Broadcast Channels Considering Data Access Frequencies," *In Proceedings of the 3rd International Conference on Mobile Data Management (MDM2002)*, pp. 87-93, 2002.
- [20] C.H. Hsu, G. Lee, and A. L. P. Chen, "A Near Optimal Algorithm for Generating Broadcast Programs on Multiple Channels," *Proc of ACM 10<sup>th</sup> Int'l Conf. Information and Knowledge Management*, pp. 303-309, 2001.
- [21] Q. Hu, W.-C. Lee, and D. L. Lee, "Indexing Techniques for Wireless Data Broadcast Under Data Clustering and Scheduling," *In Proceedings of The Eighth International Conference on Information and Knowledge Management*, pp. 351-358, 1999.
- [22] Q. L. Hu, D. L. Lee, and W. C. Lee, "Dynamic Data Delivery in Wireless Communication Environments," *Proc in: Workshop on Mobile Data Access*, pp. 213-224, 1998.
- [23] C. L. Hu and M. S. Chen, "Adaptive Balanced Hybrid Data Delivery for Multi-Channel Data Broadcast," *Proceedings of IEEE International Conference on Communications*, pp. 960-964, 2002.
- [24] Y. Huang and Y. H. Lee, "An Efficient Index Method for Wireless Data Broadcast with Dynamic Updates," *IEEE International Conference on Communications*, Vol. 1, pp. 358-362, 2002.
- [25] J. L Huang, M. S Chen, and W. C Peng, "Broadcasting Dependent

- Data for Ordered Queries without Replication in a Multi-Channel Mobile Environment," *19<sup>th</sup> International Conference on Data Engineering*, pp. 692 - 694, 2004.
- [26] J. J Hung and Y. Leu, "Efficient Index Caching Schemes for Data Broadcasting in Mobile Computing Environments," *Proceedings on 14<sup>th</sup> International Workshop on Database and Expert Systems Applications*, pp. 139 - 143, 2004.
- [27] J. J Hung and Y. Leu, "An Energy Efficient Data Reaccess Scheme for Data Broadcast in Mobile Computing Environments," *International Conference on Parallel Processing Workshops*, pp. 5-12, 2004.
- [28] K. F Jea and M. H Chen, "A Data Broadcast Scheme Based on Prediction for The Wireless Environment," *Proceedings on 9<sup>th</sup> International Conference on Parallel and Distributed Systems*, pp. 369 - 374, 2002.
- [29] F. C Jesus and K. R. Krithi, "Adaptive Dissemination of Data in Time-Critical Asymmetric Communication Environments," *Proceedings of the 11<sup>th</sup> Euromicro Conference on Real-Time Systems*, pp. 195 - 203, 1999.
- [30] S. J Kim, W. J Kim, and Y. J Suh, "Efficient Broadcast Schemes With Transmission Power Control in Mobile Ad Hoc Networks," *2004 IEEE International Conference on Communications*, Vol. 7, pp. 3859 - 3863, 2004.
- [31] S. K Lee and S. Kim, "Performance Evaluation of a Predeclaration-based Transaction Processing in a Hybrid Data Delivery," *IEEE International Conference on Mobile Data Management*, pp. 266 - 273. 2004.
- [32] G. Lee and S. C Lo, "Broadcast Data Allocation for Efficient Access of Multiple Data Items in Mobile Environments," *Mobile Networks*



*and Applications*, Vol. 8 pp. 365 - 375, 2003.

- [33] W. C. Lee and D. L. Lee, "Signature Caching Techniques for Information Filtering in Mobile Environments," *ACM Journal of Wireless Networks*, pp. 57-67, 1999.
- [34] S. C. Lo and A. L. P. Chen, "Optimal Index and Data Allocation in Multiple Broadcast Channels," *In Proceeding of the 16<sup>th</sup> International Conference on Data Engineering*, pp. 293-302, 2000.
- [35] W. C. Peng and M. S. Chen, "Dynamic Generation of Data Broadcast Programs for a Broadcast Disk Array in a Mobile Computing Environment," *Proc. of the ACM 9<sup>th</sup> Intern'l Conf. on Information and Knowledge Management*, pp. 6-11, 2000.
- [36] W. C Peng, J. L Huang, and M. S Chen, "Dynamic Leveling: Adaptive Data Broadcasting in a Mobile Computing Environment," *"Mobile Networks and Applications*, Vol.8, pp.355 - 364, 2003.
- [37] K. Prabhakara, K. A. Hua, and J. H. Oh, "Multi-Level Multi-Channel Air Cache Designs for Broadcasting in a Mobile Environment," *Proceedings of the 16<sup>th</sup> International Conference on Data Engineering*, pp. 167-176, 2000.
- [38] O. B.V Ramanaiah and H. Mohanty, "NICD: A Novel Indexless Wireless on-Demand Data Broadcast Algorithm," *Proceedings on Information Technology: Coding and Computing ITCC 2004*, Vol. 12 pp. 730 - 734, 2004.
- [39] B. Sanjoy and B. Azer, "Real-Time Mutable Broadcast Disks," *ACM Technical Report*, pp. 3-22, 1997.
- [40] N. Saxena, K. Basu, and S. K Das, "Design and Performance Analysis of a Dynamic Hybrid Scheduling Algorithm for Heterogeneous Asymmetric Environments," *Proceedings on 18<sup>th</sup> International on Parallel and Distributed Processing Symposium*, pp.

223, 2004.

- [41] P. Sumari, R. D Rozaida, and H. Kamarulhaili, "Data Organization for Broadcasting in Mobile Computing," *International Conference on Geometric Modeling and Graphics*, pp. 49 - 54, 2004.
- [42] K. Stathatos, N. Roussopoulos, and J.S. Baras, "Adaptive Data Broadcasting Using Air-Cache," *Proc. First Int'l Workshop Satellite-Based Information Services*, 1996.
- [43] K. Stathatos, N. Roussopoulos, and J. S. Baras, "Adaptive Data Broadcast in Hybrid Networks," *In Proceeding of the 23rd International Conference on Vary Large Data Bases*, pp. 326 - 335, 1997.
- [44] C.-J. Su and L. Tassiulas, "Joint Broadcast Scheduling and User's Cache Management for Efficient Information Delivery," *In Proceeding of the 4<sup>th</sup> ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 33- 42, 1998.
- [45] L. Tassiulas and C. J. Su, "Optimal Memory Management Strategies for a Mobile User in a Broadcast Data Delivery System," *IEEE Journal on Selected Areas in Communications*, pp. 1226-1238, 1997.
- [46] C. Tang, C.S. Raghavendra, and V. Prasanna, "Energy Efficient Adaptation of Multicast Protocols in Power Controlled Wireless Ad Hoc Networks," *Proceedings On Parallel Architectures, Algorithms and Networks I-SPAN '02, International Symposium*, pp. 80-85, 2002.
- [47] N. Vaidya, H. and S. Hameed, "Scheduling Data Broadcast in Asymmetric Communication Environments," *Wireless Networks*, pp. 171-182, 1999.
- [48] W. Vuchida, T. Hara, and S. Nishio, "Scheduling Correlated Broadcast Data Considering Access Frequencies with Temporal Variations" *IEEE International Network Computing and*

- Applications*, pp. 89 - 96, 2003.
- [49] A. B Waluyo, B. Srinivasan, and D. Taniar, "A Taxonomy of Broadcast Indexing Schemes for Multi Channel Data Dissemination in Mobile Databases," *International Conference on Advanced Information Networking and Applications AINA 2004 18<sup>th</sup>*, Vol. 1 pp. 213 - 218, 2004.
- [50] J. X. Yanmin Zhu, Bo Li, Ni, and L. M Lionel, "A Cooperative Caching Algorithm for Multi-Cell Data Broadcasting," *IEEE International Conference on Communications*, Vol. 7, pp. 4072 -4076, 2004.
- [51] J. Xu , B. Zheng, W. C Lee, and D. L Lee, "Energy Efficient Index for Querying Location-Dependent Data in Mobile Broadcast Environments, " *International Conference on Data Engineering 19th*, pp. 239-250, 2003.
- [52] X. Wu and V. C. S Lee, "Preemptive Maximum Stretch Optimization Scheduling for Wireless On-Demand Data Broadcast," *Proceedings on Database Engineering and Applications Symposium*, pp. 413-418, 2004.