

南 華 大 學

資訊管理學系

碩士論文

無線廣播環境下包含多個資料項的請求改良

以編碼技術為基礎的資料排程

Multi-data Request Based on Modified  
Network Coding in On-demand Broadcast

研 究 生：黃慶豐

指 導 教 授：吳光閔 博士

中華民國 98 年 10 月 7 日



南華大學資訊管理學系碩士論文著作財產權同意書

立書人：黃慶堃 之碩士畢業論文

中文題目：無線廣播環境下包含多個資料項的請求改良以編碼技術為  
基礎的資料排程

英文題目：Multi-data Request Based on Modified Network Coding in  
On-demand

指導教授：吳光閔 博士

學生與指導老師就本篇論文內容及資料其著作財產權歸屬如下：

- 共同享有著作權  
 共同享有著作權，學生願「拋棄」著作財產權  
 學生獨自享有著作財產權

學生：黃慶堃 (請親自簽名)

指導老師：吳光閔 (請親自簽名)

中華民國 98 年 9 月 30 日

南華大學碩士班研究生  
論文指導教授推薦函

資訊管理系碩士班 黃慶堃 君所提之論文  
無線廣播環境下包含多個資料項的請求改良以編碼技術為基礎的資料排程  
係由本人指導撰述，同意提付審查。

指導教授 吳光閔

98年9月30日

## 誌 謝

就讀研究所時期，非常感謝昭勳、小賴、岳勳、雅君、珊珊等學長姐的教導與照顧，讓我在踏入陌生的領域不會徬徨無助，可以很快適應。非常感謝同學淳淳、詠生、宣均、阿達以及全班同學的指教與照顧，當我在課業與生活上遇到問題時提供給我幫助。還有要感謝佳濬、雲賢、小花、東迪等學弟妹給我的鼓勵。謝謝大家的幫助與鼓勵，讓我可以順利完成研究所的課業。

最特別要感謝指導教授吳光閔老師與蔡德謙老師的細心指導，透過老師您的指導讓學生渡過論文撰寫的難題。與老師相處的過程中，我學習到老師您嚴謹務實的研究精神與耐心關愛的生活態度，學生獲益良多。希望將來能有所成就不辜負老師您諄諄教誨的苦心。這些年來有太多的事情要感謝老師，學生要誠心的跟老師您說聲：老師謝謝您。

無線廣播環境下包含多個資料項的請求改良以編碼技術為基礎的資料排程

學生：黃慶豐                      指導教授：吳光閔  
南華大學                      資訊管理系

摘 要

在無線網路下的環境架構，資料廣播是一個有效率的資料傳播方式。現今，有很多相關的研究開始探討多個資料項請求(Multi-item Queries)的廣播排程；並且，多個資料項的請求也比較符合現實的情況。在傳統的拉式或請求式(Pull or On-demand)資料廣播排程都假定一個廣播的時間單位(One Time Slot)只包含了一個資料項(Data Item)，因此排程中的一個廣播時間單位只能提供一個資料項的服務。在這裡本研究採用了一個資料編碼的技術(Network Coding)；就是將多個資料項以編碼(Encoding)的方式結合，並安排到一個廣播時間單位上；如此，一個廣播的時間單位就有多個資料項來服務使用者，使用者只要依據已經 Cache 的資料項來解碼(Decoding)，就可以得到感興趣的資料。根據先前的研究，在此本研究加入了考慮請求長度的因素來改善資料排程；我們從模擬實驗結果得知與先前[3]的研究方法作對照，我們的方法可以減少 20%左右的平均存取時間(Average Access Time)。

關鍵字：請求式無線廣播，資料排程，多資料項的請求，編碼技術

# **Multi-data Request Based on Modified Network Coding in On-demand Broadcast**

**Student: Ching-Feng Huang      Advisor: Guang-Ming Wu**

**Department of Information Management  
Nanhua University**

## **Abstract**

Data broadcasting is an efficient method of disseminating information in the on-demand wireless network environment. Recently, there are many related papers began to explore multi-item queries of data scheduling. And, users' queries needed multiple data items in real world. However, traditional on-demand data broadcast scheduling assume that each time slot includes only one data item. Therefore, each broadcast time slot can only include one data item. Using network coding technique, server can broadcast multiple data items in a time slot. In this paper, we propose a new on-demand data broadcast model with modified network coding. We also consider the factor of requests length. Our approach is different from the traditional network coding, each time slot encodes only a subset of data items, which are decided according to identities of the requested and stored data items in client's cache. Experiment results show that our algorithm can reduce the average access time by 20% compared with [3].

**Keywords : On-demand Broadcast, Scheduling Broadcast, Multi-data  
Request, Network Coding**

# 目 錄

書 名 頁	i
口 試 合 格 證 明	ii
著 作 財 產 權 同 意 書	iii
論 文 指 導 教 授 推 薦 函	iv
誌 謝	v
中 文 摘 要	vi
英 文 摘 要	vii
目 錄	viii
表 目 錄	x
圖 目 錄	xi
第 一 章 序 論	1
第 一 節 研 究 背 景 與 動 機	1
第 二 節 研 究 主 題	3
第 三 節 編 碼 技 術 的 運 作	4
第 四 節 研 究 限 制	6
第 五 節 研 究 架 構	7
第 二 章 文 獻 探 討	11
第 一 節 背 景 環 境	12

第二節	相關文獻	14
第三章	問題描述	28
第一節	評量排程效率的尺度	28
第二節	符號定義與說明	29
第三節	系統架構	32
第四節	編碼排序的問題	34
第四章	改良以編碼技術為基礎的資料排程	36
第一節	主要概念	36
第二節	運作程序	36
第三節	改良以編碼技術為基礎的資料排程	38
第四節	以實例說明 Modified OE	40
第五章	模擬實驗	43
第一節	模擬環境	43
第二節	模擬參數	44
第三節	實驗結果	47
第六章	結論	52
	參考文獻	53

## 表 目 錄

表 1	Queries and related data set and frequencies.....	18
表 2	標記的定義.....	29
表 3	模擬實驗參數.....	44
表 4	比較 Zipf 分配的參數表.....	44
表 5	比較 Mean Arrival Rate(request/time slot)分配的參數表.....	45
表 6	比較 Broadcast Cycle 的參數表.....	45
表 7	比較 Requested Length 的參數表.....	46
表 8	比較 Stored Length 的參數表.....	46

## 圖 目 錄

圖 1	Push-based System	13
圖 2	On-demand System	14
圖 3	QEM 實例說明	19
圖 4	OE 與先前研究方法在排程上的區	20
圖 5	傳統編碼與 OE 方法的例子	22
圖 6	實例說明 OE	24
圖 7	資料項在排程中的配置方式	26
圖 8	存取時間的例子	28
圖 9	舉例說明 $Req(q_i)$ 與 $Store(s_j)$	31
圖 10	採用編碼技術的 On-demand System	32
圖 11	XOR 運算說明	34
圖 12	加入請求長度考量的說明	35
圖 13	Modified OE 方法的虛擬碼	39
圖 14	實例說明 Modified OE	42
圖 15	實驗結果_對照不同 Zipf 分配參數	48
圖 16	實驗結果_對照不同的請求到達比率	48
圖 17	實驗結果_對照不同的廣播週期長度	49
圖 18	實驗結果_對照不同的請求長度	50

圖 19 實驗結果\_對照不同的存取長度.....51

## 第一章 序論

「科技始終來自於人性」，我們觀察科技的本質，科技的存在取決於人們的需求，並且滿足其需求。早期人們創造科技是為了解決其基本需求，現在是為了滿足人們更廣泛的需求和慾望。

20 世紀 50 年代，美軍使用計算機用於協調來自不同地方的訊息。70 年代後，美國學院工程師使用電信技術將計算機連結起來，形成計算網路。而後，網路突破了學院與軍隊的使用範圍擴散到了企業與社會上，終於形成了今天的國際網際網路(Internet)。90 年代，隨著電子郵件(E-Mail)與全球資訊網(World Wide Web)技術的擴散，以及乙太網路與 ADSL 網路連接技術的廉價化，網際網路變的普及化。

今日無線網路技術的普及，使得網際網路運用在行動計算上亦如影隨形，有越來越多的科技產品開始運用無線網路技術，人們使用網際網路時將不再只侷限於有線網路，還可透過無線網路技術。例如在筆記本電腦上使用 Wi-Fi 技術就是無線上網的應用。

### 第一節 研究背景與動機

現代的人們透過網路執行工作、娛樂還有訊息的交流，網路為人們的生活帶來便利、使工作變的更有效率，因此現代的人們幾乎已離不開網路的世界。

無線網路擺脫了有線網路設備不能隨身攜帶使用的限制，無線網路的應用讓訊息的傳輸能夠更即時、有效且確實。隨著無線網路的持續發展，無線網路的地位也越來越重要。例如，行動電話就是無線網路系統的一部份，行動電話透過人造衛星使人們可以隨時隨地甚至越洋進行通訊，達到無遠弗界的意義。還有，在災難的應對上，在通訊中斷或設備毀損的災難區域，架設小型的發射器建立無線網路為該地區提供一個快速的管道進行訊息的交流。

雖然無線網路比有線網路有著更多的機動與便利性，但相對的也有一些限制。在無線網路的環境架構下，使用資料廣播來播送資料是一個有效率的方式，但受限於頻寬、能源與無線電也就是傳播媒介等因素的限制。要如何將資料在有限的頻寬與能源下快速並且完整的傳送是值得要去探討研究的。有很多專家學者開始探討多種不同的技術在無線網路上的運用，研究在有限的頻寬與能源的限制下快速並完整的傳送資料。

在此我們要探討的是在一個請求式廣播環境下(On-demand System)，在多個資料項請求(Multi-item Queries)與資料項有冷熱門之分的條件下，使 Server 端能根據 Client 端發出的請求產生出一個有效率的廣播排程，其目的是為了降低 Client 端的平均存取

時間(Average Access Time)。

## 第二節 研究主題

在無線網路的環境下分為幾種不同種類的系統架構，分別被應用在特定的無線網路環境裡，不同的無線網路架構其特性也不相同。本研究將專注於一個有效率廣播排程的產生，提出一個降低 Client 端平均等待時間的排程方法。在過去的文獻中 [3][4][5][8][11][12][13] 有很多學者在不同的限制下提出不同的排程演算法來產生有效率的廣播排程以求降低 Client 端的平均存取時間。還有另一方面的研究 [6][7][14] 則是專注於提升請求成功比率的研究(Success Ratio)。在這兩方面的研究，本文是專注於降低 Client 平均存取時間的探討

[4][5][8][11][12][13] 的研究提出的是資料項排程的方法；而在文獻 [3] 中提出了一種 XOR 編碼技術的廣播排程方法，但 [3] 並沒有考慮到請求長度的因素。

因此，我們決定在 XOR 編碼技術的廣播排程方法上，再考慮請求長度因素的影響。我們加入請求長度因素的考量希望求得較少的平均存取時間。

### 第三節 編碼技術的運作

根據[3][16]研究，我們在此處介紹編碼技術硬體架構的運作。

本研究假定 Client 原本就有一些 Cache 的資料項。Client 端向 Server 提出 Query 之前，會先去 Cache Buffer 中搜尋有無需求的資料項，如果有則存取起來，如果沒有則對 Server 發出 Query。

當 Client 端向 Server 端發出 Query 時，Client 也會將 Cache 的資訊一併傳送給 Server，Server 會分別建立兩張資料表儲存 Query 與 Cache 的資訊。

當 Server 選擇了要服務的 Query 時，Server 會將 Client 所 Query 與 Cache 的資料內容轉成 0 與 1 所組成的資訊(機器語言，Machine Language)，例如，一張圖片最原始的結構是由 0 與 1 的資訊組合而成的。Server 再採用 XOR 運作將要兩兩配對的 Query 與 Cache 資料壓縮在一起。如此便完成資料的編碼。編碼前與編碼後的資料長度是不變的，因為資料內容都被轉成由 0 與 1 所組成的資訊。

當資料完成編碼後，便將資料的編碼放入封包(Packet)中，並且該封包的標頭(Title)會包含編碼在一起的資料的 ID、用來解碼的資訊與編碼的內容是圖片、檔案或文字格式的資訊。接著將封包排入排程中，本研究設定一個廣播時間單位等於一個封包的長度。

當 Client 在傾聽排程裡的資訊時，Client 會先依據封包標頭所含的資訊來判斷該封包有沒有 Client 所需求的資料，如果沒有則忽視，如果有需求的資料項則接收下來，並用封包標頭所含的解碼資訊將壓縮的資料解碼，解碼出來的資料是 0 與 1 所組合的資訊。Client 的 CPU 會根據這些資訊是圖片、檔案或文字的格式，轉換成我們所能看的懂的資料。

當 Client 存取到需要的資料項之後，會將存取下來的資料項放到 Cache Buffer 中，以便下次有相同需求時可以在 Cache Buffer 中找到。當 Cache Buffer 還有空位時則資料項直接排進 Cache Buffer 中；如果 Cache Buffer 已經滿了，本研究是採取 Least-Frequently-Used(LFU)策略[15]。LFU 的 Cache 策略是將 Cache Buffer 中請求頻率最少的資料項從 Buffer 中剔除，再將要 Cache 的資料項排入 Cache Buffer 中。

上述為採用編碼技術 Server 與 Client 的硬體系統運作過程，下面介紹傳統編碼技術與 OE[3]採用 XOR 編碼技術的差別。

XOR 將編碼結合的方式是判斷轉換成 0 與 1 的資料內容在相同位元的值是否相同，如果相同為 0，不同為 1。舉例， $d_1$  資料項轉換的內容為 0001， $d_3$  資料項轉換的內容編碼為 0011， $d_1$  結合  $d_3$  可表達成  $(d_1 \oplus d_3)$  這樣表達可以清楚得知是哪些資料項結合

成編碼， $d_1 \oplus d_3$ 所形成的編碼為 0010。接下來介紹解碼， $d_1 \oplus d_3$ 所形成的編碼為 0010，若用  $d_1$  內容為 0001 去對 0010 編碼做 XOR 運算會得到 0011，然後 CPU 再去解讀就可以得知 0011 資訊代表  $d_3$  資料項；同樣的用  $d_3$  去解碼  $d_1 \oplus d_3$  也可得到  $d_1$ 。在編碼進行編碼的組合或解碼的結果，其編碼的位元數都不會改變。

傳統的編碼技術有使用三個資料項結合的編碼方式，但這樣的傳統編碼技術的編碼運算比較複雜。例如， $(d_1 + d_2 + d_3)$ 、 $(d_1 + 2d_2 + d_3)$ 與 $(d_1 + d_2 + 2d_3)$ 為傳統的編碼方式，三個資料在作編碼時會加上係數，而解碼則需要從排程中存取到兩個編碼的資訊作聯立方程式來解碼。在此我們採用兩個資料項來作編碼的方式，在編碼與解碼的運作上 XOR 比較簡單。

#### 第四節 研究限制

本研究並非適用於所有無線網路，因此有一些關於環境架構上的限制，其限制說明如下：

- 適用於拉式或請求式的廣播系統(Pull or On-demand System)。
- 著重於 Client 端的無線廣播排程演算法，目的在於降低 Client 的平均存取時間。

- 適用於單一頻道的無線廣播環境，並未考慮多頻道資料配置的問題。
- Client 端可發出包含多個資料項的請求 (Multi-item Queries)，也就是說一個請求中包含了多的資料項。
- 資料項的長度是固定的，在此我們將資料項長度設定為 1。
- Client 端除了發送請求之外，也假定事先 Cache 了一些資料項，此處 Cache 的資料項是作為編碼與解碼之用。
- 有限的無線網路頻寬。

## 第五節 研究架構

本研究主要是探討無線廣播環境下請求式廣播系統的資料排程方法。請求式廣播系統的資料請求(Query)方式分為兩種：第一種是 Client 發出的請求為只包含了單一資料項的請求(Single-item Queries)。第二種是包含了多個資料項的請求 (Multi-item Queries)。

本研究探討的是包含了多個資料項的請求 (Multi-item Queries) 方面的議題。[3][4][5][11][12][13]都是探討一個請求包含多個資料項的議題，是普遍比較符合我們現實環境中使用網路的

情況，例如：Client 端請求一個網頁，網頁是由數個元件所組成，Client 端就必須要收到所有請求的元件才能開啟一個完整的網頁。所以多資料項的請求的議題，現今已經被很多學者注重並研究。

在此介紹兩種排程的組織結構：1. 在 [5] [8][9][10][11] [12][13]，排程中的一個廣播時間單位只包含了一個資料，並且相同資料不在同一廣播週期重複廣播。2. 在 [3] 中使用 XOR 編碼技術(Network Coding)的排程方法，一個廣播時間單位是由多個資料項所組成的編碼訊息。

在 [5] [8][9][10][11] [12][13] 中，每一個廣播時間單位(one time slot)只包含了一個資料，並且相同資料不在同一廣播週期重複廣播。因此，Client 端如果在這個廣播週期如果錯過了自己需求的資料，就必須要等到下一個廣播週期才能存取需求的資料；另外，每一個廣播時間單位也只能提供一個資料項來服務對該資料項有需求的 Client 端。

本研究採用了 [3] 編碼技術的排程方法，編碼技術就是將多個資料結合成編碼並且配置在一個廣播單位時間上播送，其將不同的資料項結合成編碼是使用 XOR 的方法。如此，一個廣播時間單位就可以為 Client 端提供多個資料項的服務。但其限制是，假

定 Client 端必須事先 Cache 一些資料項。Cache 資料項的目的在於與 Client 端請求的資料項做編碼(Encode)與解碼(Decode)的動作。另外 Client 端在發出請求之前會先查詢 Cache Buffer 中有沒有請求的資料項，如果有則存取，如果無則向 Server 提出請求。

本研究在採用了編碼技術的排程方法之外還加入了考量請求長度因素來改善資料排程的效率，在研究[4]中指出以請求長度(Request Length)為因素加入演算法來調節資料項配置在排程中的位置，能讓排程有較好的效率。其目的在於使請求長度較小的請求(Request)可以在排程中的位置配置上往前調整，使較短的請求能有機會比較長的請求優先服務。如此，便能使排程變的更有效率，希望在整體效能上能達到較小的資料平均存取時間(Average Access Time)。

本研究有系統的介紹下列的章節：在第二章介紹無線網路的背景環境與文獻探討，在無線網路背景的部份將列出幾種無線網路的架構與運作方式；文獻探討部份將闡述一些先前學者所提出來的研究方法。第三章，將介紹問題描述與符號的定義與說明，問題的描述在探討無線廣播架構下我們發展的演算法所要解決的問題；符號的定義在介紹本研究將使用到的符號並對該符號的意義作說明。第四章，介紹本研究提出來的算法，本研究提出來

的演算法是採取編碼技術為基礎並考慮請求長度的廣播排程演算法。第五章，是本研究的模擬實驗階段，在此階段將驗證本研究所提出來的的方法與先前的研究方法做比較，是否能達成較佳的效果。透過模擬實驗的數據來進行分析之後，在第六章將闡述本研究的結論。

透過這些步驟，可以完整的呈現研究架構與過程。本研究是基於無線網路環境下的請求式廣播系統(On-demand System)。在這個環境架構下本研究採用了先前研究的編碼排程的技術，並加入了考慮請求長度的因素。本研究提出 Modified OE(Modified On-demand Encoding)方法並與先前研究[3]的 OE(On-demand Encoding)方法做對照，從實驗結果得知我們的方法是一個更有效率的廣播排程方法，使 Client 端的平均存取時間更少。

## 第二章 文獻探討

隨著科技的進步，無線網路的使用者與需求大量增加。無線網路因應不同的應用可分為幾種類型：1.個人無線網路，2.無線區域網路，3.無線城域網路，4.行動裝置網路。

1.個人無線網路(Wireless Personal Area Network，WPAN)，是在小範圍內的數個裝置相互連結所形成的無線網路，通常是個人可及的範圍內。例如藍芽連結耳機及筆記型電腦。

2.無線區域網路(Wireless Local Area Network，WLAN)，其標準規範是 IEEE802.11 系列，無線區域網路是透過電波的方式在空氣中傳送，Client 端可以使用無線網卡(Wireless Card)或設備在無線基地台(Base Station，BS)與存取點(Access Point)的服務範圍內進行無線上網的動作，與裝有無線設備的 Server 端進行資料的交換。

3.無線城域網路，是連結數個無線區域網路的無線網路型式，例如 Wi-Fi。

4.行動裝置網路，例如行動電話所使用的全球行動通訊系統(Global System for Mobile Communications，GSM)，GSM 是目前使用最廣泛的行動電話通訊標準。

無線網路已經被大量應用在我們的日常生活當中並且提供許多不同的服務，像是行動電話的普及以及人們對行動電話的依賴，可以

得知我們的生活會與無線網路越來越密不可分。介紹完無線網路的應用與服務後，接下來本研究將討論無線網路的環境架構，進而討論相關的文獻探討。

## 第一節 背景環境

無線網路相較於有線網路的優勢在於可攜帶性與移動性。隨著科技的進步、行動計算機(Mobile Computing)的發達，無線網路的使用者以及需求大量的增加，網路早已融入我們的生活成為人們的必需品。而且有越來越多的科技產品附加了無線網路的功能，使得無線網路越來越普遍。

無線網路的環境下使用著資料廣播的方法進行資料傳播，資料廣播被認為是一個有效率去服務大量使用者的資料傳送方式。

在此我們介紹二種典型的資料廣播系統：

- 推式廣播系統(Push-based System)[1]：Server 端週期性的廣播資料給 Client 端，Client 端則一直監聽從 Server 端播送的資料，如果聽到自己需要的資料，就將資料存取下來。推式廣播系統架構如(圖 1)所示。

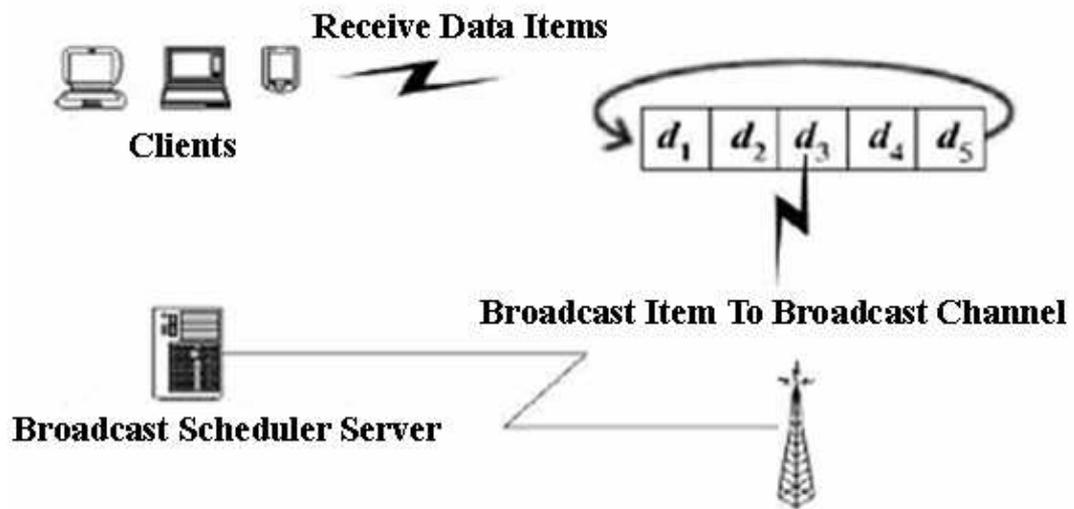


圖 1、Push-based System

- 拉式廣播系統或請求式廣播系統 (Pull-based System or On-demand System) [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14] :

Client 端從上傳頻道(Uplink Channel)送出所需資料項的請求到 Server 端，Server 端將 Client 端的請求儲存在佇列(Queue)中，並且 Server 端嘗試以較佳的排程演算法將 Client 端所請求的資料排入廣播排程，產生的排程再經由下載頻道(Downlink Channel)播送給 Client 端，Client 端可經由下載頻道接收其所需的資料使 Client 端的項請求可以得到滿足。拉式廣播系統或請求式廣播系統架構如(圖 2)所示。

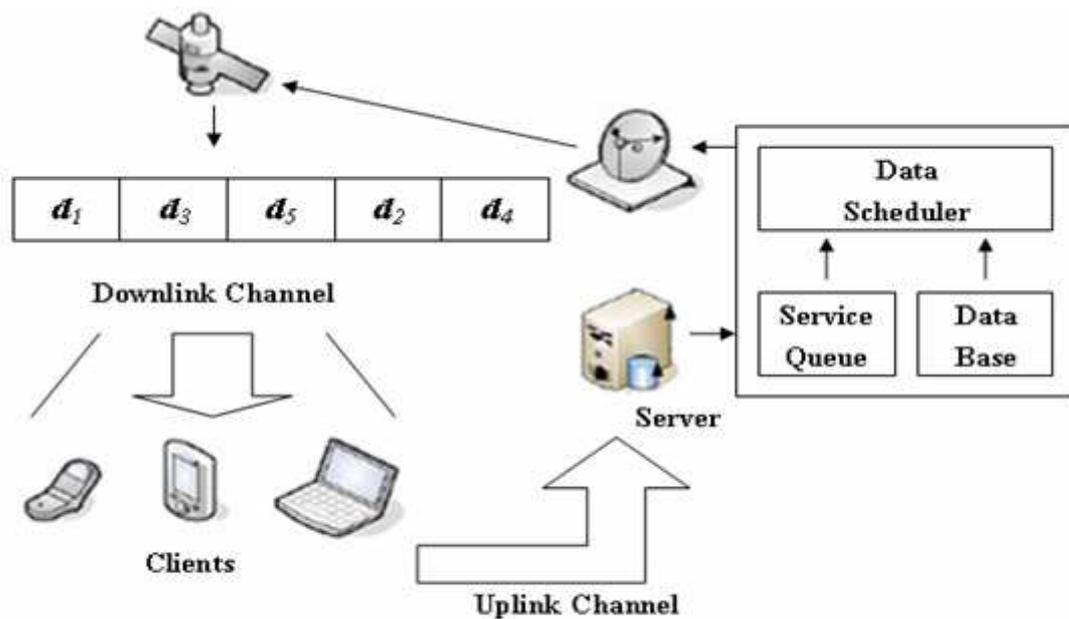


圖 2、On-demand System

## 第二節 相關文獻

在請求式廣播系統的架構下在固定的廣播週期與資料項長度的條件下要產生一個有效率的廣播排程，我們需要考量到兩種問題。一種為選擇請求的問題(Query Selection Problem)，另一種為排程中資料項配置方式的問題(Broadcast scheduling Problem)。

首先介紹選擇請求的問題(Query Selection Problem)。選擇請求的問題指的是，當 Client 端發出請求，那些被存取在 Server 端佇列(Queue)裡還未被服務的請求，我們要依據什麼來挑選那些還未被服務的請求安排入下一個廣播排程中。在此依序介紹在三種不同的條件下，先前研究所提出來的方法。第一種為單一資料項請求的廣播排程；第二種為多資料項請求的廣播排程；第三種為

採用編碼技術的多資料項請求的廣播排程。

- 在請求式廣播系統下的單一資料項請求的廣播排程：

單一資料項請求的廣播排程指的是 Client 端發出的請求只包含了一個資料項，因此 Client 端只要接收到該請求的資料項就代表它的請求已經得到滿足。以下將介紹先前的研究[8, 9, 10]提到的單一資料項請求的排程演算法，像是 FCFS(First Come First Service)，MRF(Most Requests First)，EDF(Early Deadline First)，R×W, LWF(Longest Wait First)。

FCFS(First Come First Service)[9,10]，依照請求的先後次序來決定廣播資料項的優先順序，先到達的請求資料就先廣播。因此 FCFS 可以避免多餘的廣播，而且 FCFS 也不受請求頻率和時間限制的影響。

MRF(Most Requests First)[10]，從還沒有被服務的請求(Pending Requests)中選出請求頻率最高的資料項優先廣播。請求頻率指的是資料項的被請求的次數，被請求次數越多的代表該資料項越熱門(Hot)，反之則越冷門(Cold)。MRF 演算法只考慮資料項的冷熱門程度來做排程的考量，因此在資料項有冷熱門之分並且是單一請求資料項(Single-item Queries)的環境下，MRF 被證明

是一個能有效減少請求回應時間的方法。但是如果將 MRF 運用在多資料項請求(Multi-item Queries)的環境中，那些還未被服務的請求都包含了一個到數個以上的資料項，因此同一個請求所包含的資料項在廣播排程中的位置排放上可能是不連續的，如此便會拉長請求的回應時間。

EDF(Early Deadline First) [10]，EDF 的方法被使用在確保請求能夠即時滿足的環境中，也就是 Real-time System。EDF 的排程方法是以 Deadline 為考量，接近 Deadline 時限的請求越優先排入廣播排程。Deadline 指的是請求資料項的時限，距離時限的長短代表資料項的急迫性，還未被服務的請求越接近 Deadline 的就代表越急迫，其優先權也最高；若請求在超過 Deadline 的時間還未被服務或者已經在排程中但 Client 端還未收到，就代表該請求接收失敗。

$R \times W$ [8]排程演算法則考慮兩種因素，分別是  $R$ (Request)和  $W$ (Wait Time)， $R$  指的是還未被服務的請求的頻率、 $W$  指的是還未被服務的請求的等待時間，其資料排程的優先順序是以請求頻率和等待時間的乘積為權重，權重越大的就越先排入廣播排程中。 $R \times W$  結合了 MRF 和 FCFS 的優點，在請求頻率與等待時間之間取得一個平衡點，一方面考量熱門資料是為了提供較多並且

有效率的服務，因為熱門資料代表需求的人多；另一方面，考慮請求的等待時間則可以排除請求等待時間過長，避免挨餓(Starve)的情況發生。R×W 在請求有考慮冷熱門之分與請求有等待時間的環境下提供一個較好的方法。

LWF(Longest Wait First)[10]選擇資料項廣播的優先次序是以還未被服務的請求並且若有相同的請求則將它們的等待時間相加，其等待時間的總和就是它的權重值，其意義為等待時間總和最長的請求優先廣播。LWF 在資料項有冷熱門之分以及是單一請求資料項的環境下，可以有效的減少請求回應的時間，因為請求的頻率會影響到等待時間的總和，所以 LWF 方法有間接的去考慮到請求頻率的元素。

另外，SSTF(Shortest Service Time First)與 SRST(Shortest Remaining Service Time)[9]方法則是允許資料項的長度不一樣，它是考量有關於請求長度的因素，其意義為花最短時間就能滿足得請求應該優先排入排程。

- 在請求式廣播系統下多資料項請求的廣播排程：

先前所提到的研究，Client 端在請求資料時只請求一個資料項；而在現實環境中 Client 也有發出包含多個資料項請求的情

形。在多資料項請求的環境下 [3, 4, 5, 11, 12, 13]提到了一個 QEM(Query Expansion Method)的方法。介紹 QEM 之前我們先說明 Query Data Set(QDS)與 Query Distance(QD)地意義。QDS 指的是一個請求所包含資料項的集合，列如  $QDS(q_i)$  可以代表  $q_i = \{d_1, d_2, d_4, d_5\}$ 。QD 指的是同一個請求所包含的資料項之間相距最遠的距離，例如  $q_i = \{d_1, d_2, d_4, d_5\}$  的 QD 為 4。

QEM 建構資料排程的方法的如下：第一點，QEM 是以 Query 的請求頻率來決定排程先後次序，依據請求的熱門到冷門程度依次序將請求的 QDS 展開。第二點，在 Query 一個一個接著展開的過程中當中，先前已經展開的 Query 必須保持著相同的 QD 不變。第三點，同樣在 Query 展開的過程當中，可以再先前展開的 Query 保持著相同 QD 的情形下調整資料項的位置，使其有相依性的資料可以緊密的排在一起。以下舉例說明 QEM 演算法的流程。(表 1)為個各請求的資料項集合與頻率，(圖 3)為 QEM 方法的實例說明。

表 1：Queries and related data set and frequencies

query	query data set	Frequency
$q_1$	$d_2, d_3, d_4, d_6$	10
$q_2$	$d_1, d_3, d_4, d_5$	9
$q_3$	$d_3, d_7$	8

Step1 展開 query  $q_1$  如下：

$[d_2, d_3, d_4, d_6]$

Step2 展開 query  $q_2$  如下：

$[d_2, d_6] [d_3, d_4] [d_1, d_5]$

展開  $q_2$  時發現  $d_3, d_4$  分別與  $q_1, q_2$  有相關性，因此可調整  $d_3, d_4$  的位置於兩個請求之間，並且保持著先前已經展開的請求 QD 不變。

Step3 最後展開 query  $q_3$  的結果如下：

$[d_2, d_6] [d_4] [d_3] [d_1, d_5] [d_7]$

展開  $q_3$  時發現與先前已經展開的請求有相同的資料項  $d_3$ ，因此在不改變先前已經展開請求的 QD 情況下， $d_3$  可以往  $q_3$  的方向調整位置。

圖 3、QEM 實例說明

- 在請求式廣播系統下採用編碼技術的多資料項請求的廣播排程：

關於編碼技術資料排程的多資料項請求在拉式廣播環境下，在[3]提出了 OE(On-demand Encoding)的演算法。OE 的排程方法與先前的研究在單一資料項或多資料項請求環境中所提出來的的方法最大的不同在於，先前研究的排程方法在排程中的一個廣播

時間單位(Time Slot)都只排放了一個資料項;但 OE 方法卻是將不同的資料項以 XOR 方式結合成一個編碼再排放到排程中，因此 OE 方法在排程中的一個廣播時間單位就可以提供多個資料項的服務。OE 與先前研究在排程中的差別如(圖 4)。

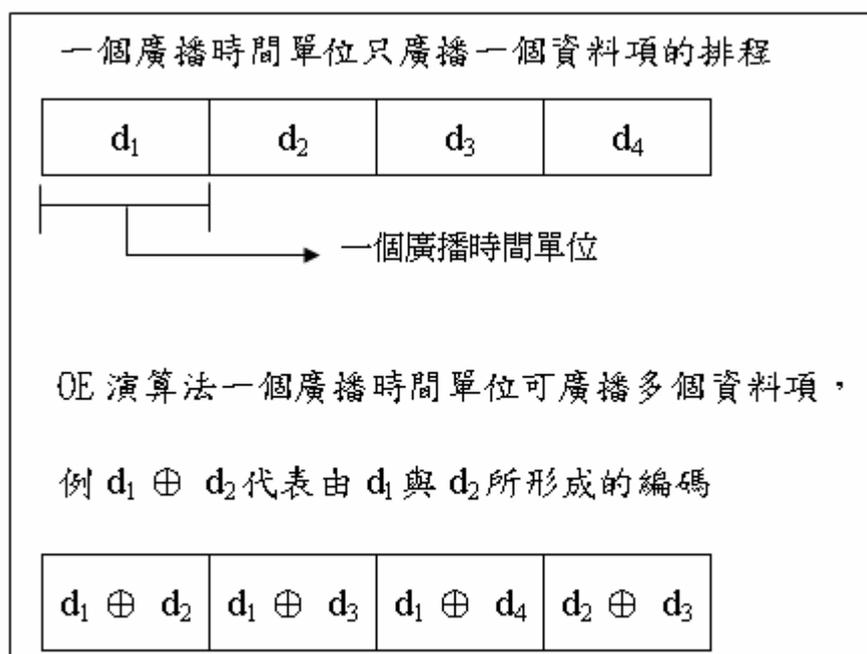


圖 4、OE 與先前研究方法在排程上的區別

編碼技術的資料排程，Client 端除了發出請求的資料項之外 (Requested Data Items)，Client 端本身還包含了一些已經事先存取的資料項(Stored Data Items)或稱為 Cache。同時，在向 Server 端提出請求時也會將存取的資料項的資訊發送給 Server 端；Server 端則採用 XOR 的運算使還未處理過的請求資料項與存取資料項做結合並形成編碼安排到廣播排程中；因此，Client 端在接收資

料時再使用封包的標頭裡所含的解碼資訊來解碼，以得到需求的資料項。

在此介紹 OE 演算法的運作過程，OE 運作過程分為要服務的請求的選取、編碼與排程、廣播。

首先選取要作排程的請求。Server 端從佇列中那些還未被滿足的請求裡選取要先服務的請求，其請求要被服務的先後次序是以該請求的資料項在所有請求裡出現的平均次數乘上所 Cache 的資料項在所有 Cache 中平均出現的次數再乘以等待時間就可以得到權重值，權重越高的就越優先選取。

第二步為編碼與排程。在選取要先排程的請求之後先將該請求的 Query 資料項與 Cache 資料項兩兩相互結合形成編碼，再以形成的編碼可以服務多少個請求的次數乘上可以服務的請求的等待時間總和得到該編碼的權重值，權重越高的編碼越優先排入廣播排程。編碼可以服務的請求指的是編碼可以被 Client 端解碼並存取。

最後 Server 將排程廣播出去，並且刪除佇列中已經服務的請求以及更新這段時間到達 Server 端的請求。

另外在此舉一傳統編碼技術與 OE 編碼技術的例子，如下(圖 5)。圖 5(a)是未使用編碼技術的排程，其總和的存取時間為 11；

圖 5(b)為傳統的編碼技術排程，使用方法是以三個資料項加上係數做編碼，但 Client 端在解碼的時受必須存取兩個編碼並使用聯立方程式才能解碼，其總和存取時間為 10；圖 5(c)只使用資料項子集的編碼方式，使用兩個資料項來做編碼的方式，代表 Client 端只需要一個相對應的 Cache 資料就可以進行解碼得到需求的資料項，其總和存取時間為 9；圖 5(d)則是排除圖 5(c)的一個沒有作用的編碼，來達到改善排程效率的目的，其總和的存取時間為 7。然而，在此我們發現[3] 所提出的 OE 方法並未考慮請求長度的問題。

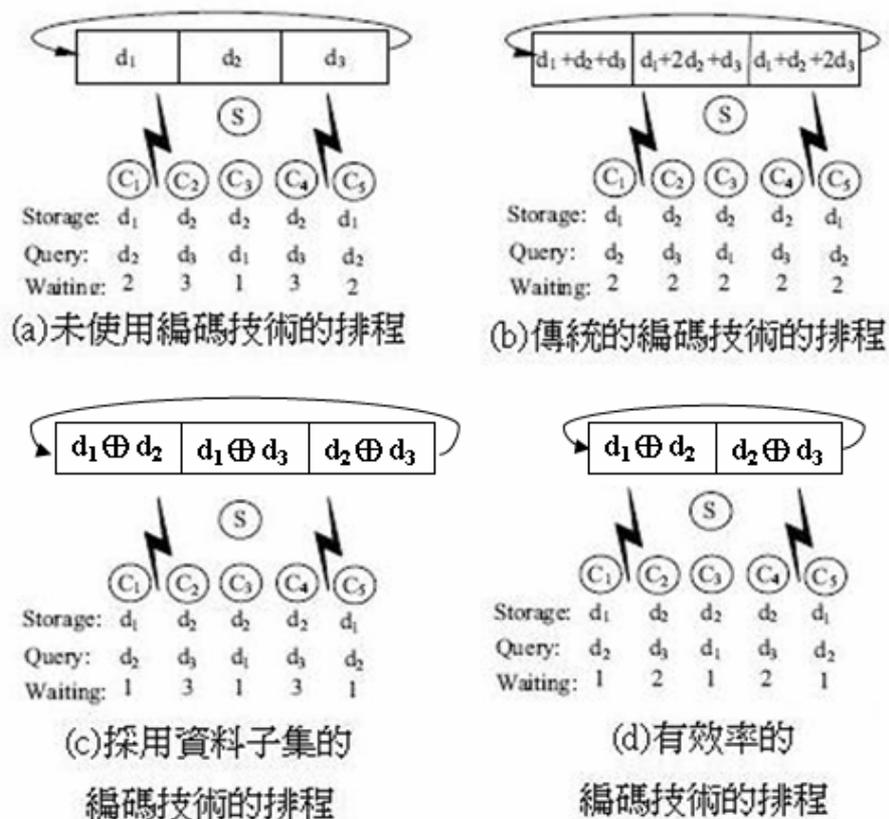


圖 5、傳統編碼與 OE 方法的例子

上述介紹了傳統編碼技術與 OE 的不同，在此我們以一個例子來說明 OE 的方法，如(圖 6)。

如圖 6(a)的例子，包含了 4 個請求、request-stored 的資料、wait time、request length 的資訊。

要做資料排程時，首先要先選取服務哪個請求，選擇要服務的請求的優先次序是以  $p_i$  值來決定， $p_i$  值代表請求的權重，其計算方式是  $p_i = Req(q_i) * Store(s_i) * w_i$ ，如圖 6(b)。

我們發現  $q_1$  的權重是最高的，所以  $q_1$  要優先被服務。在選擇要服務的請求之後，便以該請求的請求資料項與 Cache 資料項相互配對作編碼，並且計算編碼的強度  $DI(e_j)$ ，編碼權重的計算方

式為  $DI(e_j) = |Dem(e_j)| * \sum_{ui \in Dem(e_j)} w_i$ ，如圖 6(c)。在計算出編碼

的權重之後，再將編碼依權重由高到低排入排程中，如圖 6(d)

接著判定排程是否還有空位，如果沒有則確立排程完成並且進行資料廣播；如果排程還有空位選擇次高的  $p_i$  值請求來作編碼並排程，我們發現  $q_2$ 、 $q_3$  與  $q_4$  的  $p_i$  權重都是次高並且相同的，如果發現此情形我們是採用隨機選取一個請求來服務，在這個例子上我們選取的是  $q_2$  來做服務，其編碼權重，如圖 6(e)。將編碼排入排後，程如圖 6(f)。

$q_2$  所產生的編碼排完後，排程還有空位，接著選取  $p_i$  值相同

的  $q_3$  或  $q_4$  中的一個請求來服務。在此我們選擇  $q_3$ ， $q_3$  所產生的編碼權重如圖 6(g)，編碼排入排程如圖 6(h)。當排放完  $q_3$  所產生的編碼，我們發現排程已經被排滿了，因此中止演算法的執行，圖 6(h)就是此例子採用 OE 演算法所產生的排程。

Storage	Request	Waiting time (time slot)	Request length ( $l_i$ )
$s_1 = \{d_1\}$	$q_1 = \{d_2, d_3, d_4\}$	1	--
$s_2 = \{d_1\}$	$q_2 = \{d_5\}$	2	--
$s_3 = \{d_2\}$	$q_3 = \{d_1, d_3\}$	4	--
$s_4 = \{d_1\}$	$q_4 = \{d_2, d_5\}$	1	--

(a)使用者 request-stored 的資料與 wait time 但不考慮 request length

Request	Store( $s_i$ )	Req( $q_i$ )	Product $p_i$
$q_1$	3/4	5/4	15/16
$q_2$	3/4	1/2	3/4
$q_3$	1/4	3/4	3/4
$q_4$	3/4	1	3/4

(b)計算每一個 request 的  $p_i$  值

Encoding data $e_j$	Dem( $e_j$ )	Total waiting time	DI( $e_j$ )
$d_1 \oplus d_2$	$\{u_1, u_3, u_4\}$	6	18
$d_1 \oplus d_3$	$\{u_1\}$	1	1
$d_1 \oplus d_4$	$\{u_1\}$	1	1

(c)取  $u_1$  的 request-stored 資料作編碼，並計算編碼的權重 DI( $e_j$ )

$d_1 \oplus d_2$	$d_1 \oplus d_3$	$d_1 \oplus d_4$		
------------------	------------------	------------------	--	--

(d) 使用者  $u_1$  的 request-stored 的產生的編碼依權重排入排程

Encoding data $e_j$	Dem( $e_j$ )	Total waiting time	DI( $e_j$ )
$d_1 \oplus d_5$	$\{u_2, u_4\}$	3	6

(e)取次高  $p_i$ ，若  $p_i$  相同隨機取一，在此取  $u_2$ ，計算每一編碼的權重 DI( $e_j$ )

$d_1 \oplus d_2$	$d_1 \oplus d_3$	$d_1 \oplus d_4$	$d_1 \oplus d_5$	
------------------	------------------	------------------	------------------	--

(f)依 DI( $e_j$ )權重大小排入

Encoding data $e_j$	Dem( $e_j$ )	Total waiting time	DI( $e_j$ )
$d_1 \oplus d_2$	$\{u_1, u_3, u_4\}$	6	18
$d_2 \oplus d_3$	$\{u_3\}$	4	4

(g)接著選取  $u_3$  的 request-stored 編碼

$d_1 \oplus d_2$	$d_1 \oplus d_3$	$d_1 \oplus d_4$	$d_1 \oplus d_5$	$d_2 \oplus d_3$
------------------	------------------	------------------	------------------	------------------

(h)排程已滿，演算法執行到此停止，OE 例子的排程結果

圖 6. 實例說明 OE

從[3, 4, 5, 6, 8, 9, 10, 11, 12, 13]，我們觀察到為了達到減少 Client 端平均存取時間的目的，可以根據下列三種因素。第一點，請求的頻率越高應該要有較高的權重。第二點，請求的等待時間越長，就應該要更優先考慮。第三點，應選取請求長度較小的先做排程；請求的長度指的是一請求裡面資料項集合的長度，在此，我們假定資料項的長度皆為 1。所以我們將採用[3]的編碼技術在 On-demand 的多資料項請求的環境下，並考慮請求長度的因素來改良[3]的 On-demand Encoding 方法。

介紹了選擇請求的問題(Query Selection Problem)之後，接下來將介紹排程中資料項配置方式的問題(Broadcast scheduling Problem)。排程中資料項配置的方式有兩種：第一種在同一個廣播週期中同一個資料項只配置一次；另一種為在同一個廣播週期中同一個資料項可以重複的配置。如下(圖 7)。

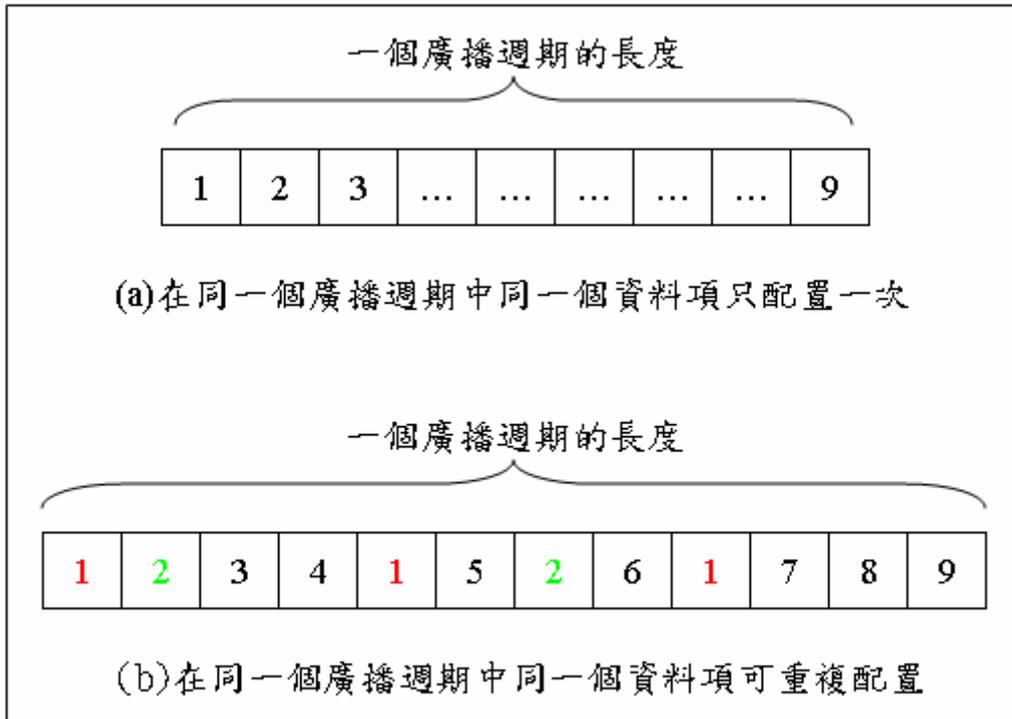


圖 7、資料項在排程中的配置方式(a)單一次配置(b)重複配置

- 在同一個廣播週期中同一個資料項只配置一次

先前的研究[3, 5, 8, 9, 10, 11, 13]採用的方式都是同一資料在一廣播週期都只配置一次，如(圖 7(a))。此種單一次資料配置的方法相較於重複配置的方法，優點在於所產生的排程廣播週期長度會比較短，因此下一個排程可以較快的被廣播。缺點則是當 Client 端如果在現行的廣播排程當中錯過了需求的資料項，就必須要等到下一個有需求資料的廣播排程才能收到所需的資料項。

- 在同一個廣播週期中同一個資料項可以重複配置

研究[4, 6, 7, 11, 12]採用的資料項配置方法是在同一個廣播週

期中重複的配置資料項，(如圖 7(b))。資料項重複配置的方法其優點在於當 Client 端如果錯過了請求的資料項，還有機會在該資料項重複出現的時候存取資料項。缺點則是重複配置資料項的方法所產生的廣播週期長度比較長，因此下個廣播排程就要等待比較久的時間才能被廣播。

[6, 7]所提出的方法，用來決定資料項在廣播週期中出現的次數與請求頻率有關，如(圖 7(b))。例， $d_1$  與  $d_2$  都是熱門的資料項，最熱門的  $d_1$  要在廣播週期中出現的頻率是 3 次，次熱門的  $d_2$  是 2 次。而資料項排放到排程中的位置是以廣播週期的長度除以資料項要出現的頻率為區間排放，如果要排放的位置已經有資料項了，則順移到下一個空的位置。等到熱門的資料項排放完後，如果排程還有空間則用冷門的資料補足。

### 第三章 問題描述

再前一章節的部份，我們已經介紹了一些相關的文獻著作。接著在本章我們將會詳細說明在文獻探討中，所發現的問題。本章節將有系統的介紹評量尺度、符號定義與說明、系統架構、編碼排序的問題。

#### 第一節 評量尺度

我們用來評估廣播排程效率的尺度為平均存取時間(Average Access Time, AAT)，存取時間(Access Time, AT)是指 Client 發出請求後，直到 Client 端接收完所有感興趣的資料的這段時間，如(圖 8)， $q_i$  請求了 A, B, E 三個資料項，因此  $q_i$  必須接收到 E 才能滿足它的請求，這段時間就是存取時間。

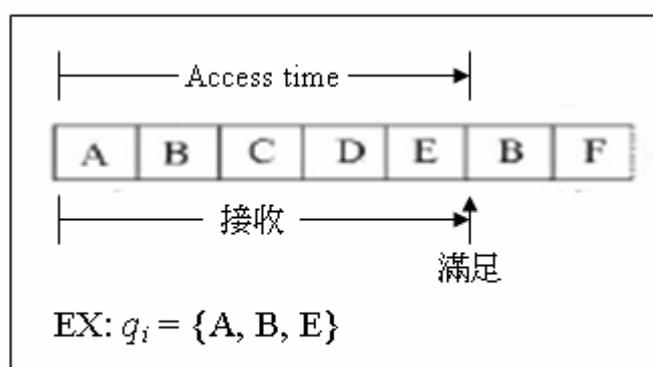


圖 8、存取時間的例子

而我們做廣播排程的目的就是要找出一個理想的方法降低 Client 端的平均存取時間(AAT)。平均存取時間的計算我們可以表示為下列方程式(1)，平均存取時間等於所有被滿足的請求的存取

時間總和除以被滿足請求的個數。

$$AAT = \frac{\text{Total access time of arrival requests}}{\text{Number of arrival requests}}. \quad (1)$$

## 第二節 符號定義與說明

表 2 定義本研究所使用的標記符號，以下將清楚的說明本研究會使用到的標記符號的意義。

表 2：標記的定義

標記	意義
$D$	資料庫裡的資料項集合
$ D $	資料庫裡資料項的個數
$L$	廣播週期的長度
$ Q $	總共有多少個 request, 相對的也就有多少個 store
$u_i$	user $i$
$q_i$	user $i$ 所發出的請求
$w_i$	$q_i$ 的等待時間
$l_i$	$q_i$ 的長度
$s_i$	user $i$ 所 cache 的資料項的集合
$d_j$	資料庫裡第 $j$ 個資料項
$s_{d_j}$	$d_j$ 這資料項在所有 cache 資料項中出現的次數
$r_{d_j}$	$d_j$ 這資料項在所有請求資料項中出現的次數
$Store(s_i)$	$s_i$ 所包含的資料項在所有 cache 裡平均出現的次數
$Req(q_i)$	$q_i$ 所包含的資料項在所有 request 裡平均出現的次數
$e_j$	Request 與 Store 資料項配對形成的編碼組合， ( $e_j = \sum_{i=1}^{n_d} c_i d_i$ ), $1 \leq j \leq L$
$p_i$	$p_i$ 權重決定選取哪一個使用者的 request 與 store 來做編碼
$Dem(e_j)$	有多少使用者可從 $e_j$ 得到資料
$DI(e_j)$	$e_j$ 編碼的權重，權重大小決定編碼排入排程的次序

在表 2 中， $D$  指的是資料庫中所有資料項的集合， $D$  可表達成  $D = \{d_1, d_2, d_3, \dots, d_n\}$ ， $d_j$  表示指的是第  $j$  個資料項， $|D|$  則是資料庫中資料項的個數總和。由 Client 端發出的請求(Request)標示成  $q_i$  (此處  $i$  代表 Request 的編號)， $|Q|$  標示代表有幾個 Request。Client 端所存取(Store)的資料項也就是 Cache 則標記成  $s_i$  (此處  $i$  代表著與 Request 相同的編號，因為 Client 端發出請求時同時也向 Server 端傳遞事先已經 Cache 的資料項)。 $w_i$  標記代表  $q_i$  的等待時間。 $\ell_i$  標記代表的  $q_i$  長度，在此資料項設定了固定的長度為 1，因此  $\ell_i$  也可指  $q_i$  所包含的資料項個數。

$r_{d_j}$  標記代表  $d_j$  這個資料項在所有 Request 的資料項集合中出現的次數，而  $Req(q_i)$  則代表  $q_i$  這個請求的資料項在所有 Request 裡出現的平均次數，其計算方式為(方程式 2)。 $s_{d_j}$  標記代表  $d_j$  這個資料項在所有 Cache 的資料項集合中出現的次數。 $Store(s_i)$  則代表  $s_i$  這個 Cache 的資料項在所有 Cache 裡出現的平均次數，其計算方式為(方程式 3)。

$$Req(q_i) = \frac{\sum_{d_j \in q_i} r_{d_j}}{|Q|} \quad (2)$$

$$Store(s_i) = \frac{\sum_{d_j \in s_i} s_{d_j}}{|Q|} \quad (3)$$

在此舉例說明  $Req(r_{dj})$  與  $Store(s_{dj})$ ，如下(圖 9)。

Storage	Request	Wait time
$s_1 = \{d_1, d_3\}$	$q_1 = \{d_2, d_5\}$	$w_1 = 2$
$s_2 = \{d_3, d_5\}$	$q_2 = \{d_2, d_4\}$	$w_2 = 4$

**$Store(s_1)$  等於 1.5，其過程如下：**

$$Store(s_i) = \frac{\sum_{d_j \in S_i} s_{d_j}}{|Q|} = \frac{s_{d1} + s_{d3}}{|Q|} = \frac{1 + 2}{2} = 1.5$$

**$Req(q_1)$  等於 1.5，其過程如下：**

$$Req(q_i) = \frac{\sum_{d_j \in q_i} r_{d_j}}{|Q|} = \frac{r_{d2} + r_{d5}}{|Q|} = \frac{2 + 1}{2} = 1.5$$

圖 9、舉例說明  $Req(q_1)$  與  $Store(s_1)$

當演算法在運作時，Server 端要優先選取要以哪一個 Request 與一同傳遞過來的 Cache 資料項相互結合成編碼是以  $p_i$  為依據。所以  $p_i$  是決定哪一個請求優先服務的權重值， $p_i$  越大代表權重越高。 $p_i$  的計算方式如(方程式 4)。

$$p_i = Req(q_i) \times Store(s_i) \times w_i \times \frac{1}{\ell_i} \quad (4)$$



在本文前述的部份，已經說明了無線網路的整個大環境。而本研究主要探討的是包含多資料項請求的請求式廣播系統並且採用編碼排程方法的架構，如上(圖 10) 所示。

請求式廣播系統是依據 Client 端所提出的 Request 再給予回應，Client 端經由 Uplink Channel 發出 Request 與 Cache 的資訊，Server 端會將 Request 的資訊儲存在佇列(Queue)中並等待服務。隨著演算法技巧的不同，資料項排入排程的先後次序也不相同，本研究因為採用編碼的方法所以排程中廣播的資訊是以編碼的形式，編碼是由 Request 與 Cache 的資料項先轉成 0 與 1 的資訊，再相互配對組成編碼。當排程產生後 Server 端就將排程經由 Downlink Channel 廣播出去，Client 端此時則傾聽需求的資料。Client 會先依據封包標頭所含的資訊來判斷該封包有沒有 Client 所需求的資料，如果沒有則忽視，如果有需求的資料項則接收下來，並用封包標頭所含的解碼資訊將壓縮的資料解碼，解碼出來的資料是 0 與 1 所組合的資訊，CPU 會將該資訊轉成我們看的懂的資料。

以上為本研究的系統架構的運作，接下來介紹 Request 與 Cache 該如何做編碼與解碼。其編碼與解碼的運算方法是採用 XOR 的運算方法，如(圖 11)所示。

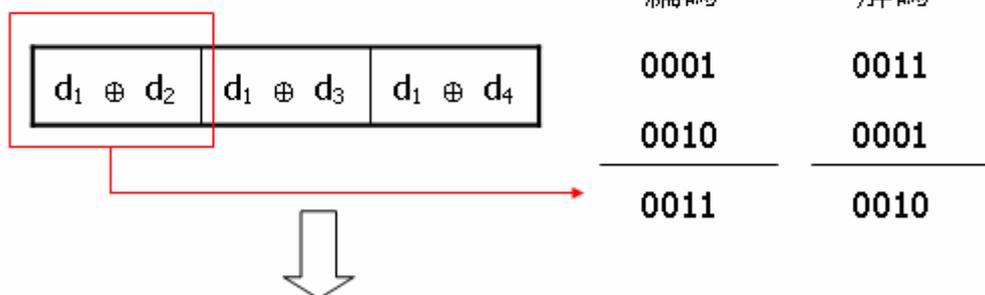
例子：假設這 Request 被選擇進入排程

User	Storage	Request	Waiting time	request length
$u_1$	$s_1 = \{d_1\}$	$q_1 = \{d_2, d_3, d_4\}$	1	3

此為資料項轉成 0 與 1 資訊的對照表

資料項	$d_1$	$d_2$	$d_3$	$d_4$
代碼	0001	0010	0011	0100

這是依據該 Request 所形成的編碼排程



這是排程中實際的資訊

0011	0010	0101
------	------	------

圖 11、XOR 運算說明

#### 第四節 編碼排序問題

在文獻探討的部份，我們了解到各式各樣的演算法都與下列三種因素有關。第一種，請求的頻率越高應該要有較高的權重，其意思為先服務大部分請求熱門資訊的使用者。第二種，請求的等待時間越長，就應該要更優先考慮，避免挨餓情形。第三點，

應選取請求長度較小的請求先做排程。

我們的方法是採用[3]OE 演算法的編碼方式，但我們發現 OE 演算法並沒有考慮到請求長度的因素，因此本研究提出的方法加入了請求長度的考量。在(圖 12)說明為何要考量請求長度的因素。

例子：假設 Queue 中 2 個請求

Query	Storage	Request	Waiting time	request length
q <sub>1</sub>	s <sub>1</sub> = {d <sub>1</sub> , d <sub>5</sub> }	q <sub>1</sub> = {d <sub>2</sub> , d <sub>3</sub> }	2	2
q <sub>2</sub>	s <sub>2</sub> = {d <sub>1</sub> }	q <sub>2</sub> = {d <sub>4</sub> }	1	1

請求較長的先排，先選 q<sub>1</sub> 在選 q<sub>2</sub> 進入排程的結果如下：

d <sub>1</sub> ⊕ d <sub>2</sub>	d <sub>1</sub> ⊕ d <sub>3</sub>	d <sub>2</sub> ⊕ d <sub>5</sub>	d <sub>3</sub> ⊕ d <sub>5</sub>	d <sub>1</sub> ⊕ d <sub>4</sub>
---------------------------------	---------------------------------	---------------------------------	---------------------------------	---------------------------------

q<sub>1</sub> 在讀取到 d<sub>1</sub> ⊕ d<sub>2</sub> 與 d<sub>1</sub> ⊕ d<sub>3</sub> 就可以存取到 d<sub>2</sub> 與 d<sub>3</sub>，因此存取時間為 2。  
 q<sub>2</sub> 必須讀取到 d<sub>1</sub> ⊕ d<sub>4</sub> 才能滿足，存取時間為 5。  
 總和存取時間為 7。

請求較短的先排，先選 q<sub>2</sub> 在選 q<sub>1</sub> 進入排程的結果如下：

d <sub>1</sub> ⊕ d <sub>4</sub>	d <sub>1</sub> ⊕ d <sub>2</sub>	d <sub>1</sub> ⊕ d <sub>3</sub>	d <sub>2</sub> ⊕ d <sub>5</sub>	d <sub>3</sub> ⊕ d <sub>5</sub>
---------------------------------	---------------------------------	---------------------------------	---------------------------------	---------------------------------

q<sub>1</sub> 的存取時間為 3。 q<sub>2</sub> 的存取時間為 1。  
 總和存取時間為 4。

圖 12、加入請求長度考量的說明

## 第四章 改良以編碼技術為基礎的資料排程

本研究在此提出一個有效率的排程方法為 Modified On-demand Encoding (Modified OE)，在 OE 演算法的基礎下並考量請求長度的因素可以有效的減少 Client 端存取資料的時間。下列將完整詳細介紹 Modified OE 的運作。

### 第一節 主要概念

如前所述，本研究演算法的目的主要是為了減少 Client 端的平均存取時間。為了解決上述問題而根據 OE 方法並加入了請求長度的考量因素。

雖然加入了請求考量的因素，但我們也不能忽略請求頻率和等待時間對資料排程的影響。在 OE 方法的基礎上，已經有考慮到請求頻率與等待時間的因素，本研究加入請求長度的考量是為讓原本只考慮請求頻率與等待時間所形成的排程，使請求長度較小的請求在排程的次序上可以往前調整，請求的排程的次序並非完全由請求長度來決定，而是在這三個因素中尋求一個平衡點。

### 第二節 運作程序

在此介紹整個系統的運作程序。

當系統開始運作時，我們假定 Client 端已經事先 Cache 了一些資料項。在 Client 端發出請求時 Client 端也會同時將它 Cache 的資訊一併傳送給 Server 端，Server 會將 Client 端傳送過來的請求與 Cache 資訊以表格的方式存取在佇列中等待服務。接下來依據演算法產生排程。最後就將排程廣播出去，被服務過的請求則從等待佇列裡移除。

當 Client 端在傾聽資訊時，Client 會從封包標頭所包含的資訊判斷有無需求的資料，如果無則忽視該封包，如果有則存取下來。Client 端會使用封包的標頭所包含的解碼資訊來解碼存取下來的編碼資訊，解碼出來的資訊會被 CPU 轉換成我們所看的懂的資料。同時存取下來的資料項同時也會 Cache 起來，以便 Client 端下次有同樣需求時可以直接從 Cache 中讀取。

接下來介紹演算法的運作程序，演算法可以分為三個部份在運作，分別是選擇要服務的請求、編碼與排程、廣播與更新。

- 選擇要服務的請求

Server 使用方程式(4) ( $p_i = Req(q_i) * Store(s_i) * w_i * (1/l_i)$ )

計算在等待佇列中還未被服務的請求的權重值。方程式(4)所產生的權重值是根據請求的請求頻率、等待時間、請求長度

來三個因素產生的。

- 編碼與排程

當選出要先服務的請求之後，該請求的資料項會與一起傳送到 Server 端 Cache 的資料項兩兩配對形成編碼。這些編碼排入排程的次序是依據方程式(5) ( $DI(e_j) = |Dem(e_j)| * \sum_{i \in Dem(e_j)} w_i$ )。方程式(5)求得的  $DI(e_j)$  代表編碼的權重， $DI(e_j)$

值越高的編碼越優先排入排程。如果編碼未將廣播排程排滿則繼續挑選請求並形成編碼，直到廣播排程都被排滿或者所有的請求都被服務。

- 廣播與更新

當排程程序完成後就將排程廣播出去。接下來 Server 會從佇列裡刪除已經服務的請求並且更新這段時間到達 Server 端的請求與 Cache 資訊。

### 第三節 改良以編碼技術為基礎的資料排程

我們所提出的 Modified OE 方法可以有效的減少 Client 端整體的平均存取時間。接著下列為 Modified OE 方法的虛擬碼，如(圖 13)。

**Algorithm Modified OE :**

**Input :** The arrival users  $u_1, u_2, \dots, u_{nu}$ , the wait time  $w_1, w_2, \dots, w_{nu}$ ,  $\ell_1, \ell_2, \dots, \ell_{nu}$  for each user after issue a request, the set of stored data items  $s_1, s_2, \dots, s_{nu}$ , and the set of requested data items  $q_1, q_2, \dots, q_{nu}$ , the broadcast cycle length  $L$

**Output :** The broadcast program  $P$

1. Calculate each  $Store(s_i)$  in Eq (3)

$$Store(s_i) = \frac{\sum_{d_j \in s_i} s_{d_j}}{|Q|}$$

2. Calculate each  $Req(q_i)$  in Eq (2)

$$Req(q_i) = \frac{\sum_{d_j \in q_i} r_{d_j}}{|Q|}$$

3. Calculate each  $p_i$  in Eq(4)

select the query  $q_i$  and  $s_i$  with the highest value of  $p_i$

$$p_i = Req(q_i) \times Store(s_i) \times w_i \times \frac{1}{\ell_i}$$

4. for each  $d_m$  in  $q_i$ , where  $1 \leq m \leq |q_i|$
5. for each  $d_n$  in  $s_i$ , where  $1 \leq n \leq |s_i|$
6. Calculate  $e_j = d_m + d_n$ , where  $1 \leq j \leq |q_i| \times |s_i|$
7. Calculate  $DI(e_j)$  in Eq (5)

$$DI(e_j) = |Dem(e_j)| \times \left( \sum_{u_i \in Dem(e_j)} w_i \right)$$

8. Select the encoding data with highest value of  $DI(e_j)$ , and allocate the encoding data into the broadcast  $P'$  such that  $q_i$  is satisfied
9. Respect step 1 to 8 until  $|P'| \geq L$
10. Replace the existing broadcast  $P$  with  $P'$
11. Return  $P$

圖 13、Modified OE 方法的虛擬碼

#### 第四節 以實例說明 Modified OE

在此我們以一個實例說明 Modified OE，如(圖 14)。

圖 14(a)是我們所要說明的例子。此例子有 4 個請求並且包含了 request-stored 的資料、wait time、request length 的資訊。

要做資料排程時，首先要先選取服務哪個請求，選擇要服務的請求的優先次序是以  $p_i$  值來決定， $p_i$  值代表請求的權重，其計算方式是  $p_i = Req(q_i) * Store(s_i) * w_i * (1/\ell_i)$ ，如圖 14(b)。

我們發現  $q_2$  的權重是最高的，所以  $q_2$  要優先被服務。在選擇要服務的請求之後，便以該請求的請求資料項與 Cache 資料項相互配對作編碼，並且計算編碼的強度  $DI(e_j)$ ，編碼權重的計算方

式為  $DI(e_j) = |Dem(e_j)| * \sum_{ui \in Dem(e_j)} w_i$ ，如圖 14(c)。在計算出編

碼得權重之後，再將編碼依權重由高到低排入排程中，如圖 14(d)。

接著判定排程是否還有空位，如果沒有則確立排程完成並且進行資料廣播；如果排程還有空位選擇次高的  $p_i$  值請求來作編碼並排程，我們發現  $q_3$  與  $q_4$  的  $p_i$  權重都是次高並且相同的，如果發現此情形我們是採用隨機選取一個請求來服務，在這個例子上我們選取的是  $q_3$  來做服務，其編碼權重，如圖 14(e)。將編碼排入排後，程如圖 14(f)。編碼排入排程後我們發現排程還有空位，於是我們接著選取  $q_4$  來做服務，但  $q_4$  所產生的編碼為  $d_1 \oplus d_2$ 、 $d_1$

⊕  $d_5$ ，此兩種編碼在在之前就已就被排進排程了，因此在編碼要排入排程時會去檢查有無重複的編碼，如果重複了就放棄要排入排程。

最後選擇服務  $q_1$ ， $q_1$  所產生的編碼配對與編碼強度，如圖 14(f)。以及依據編碼強度將編碼排入排程，如圖 14(h)。

最後我們發現排程已經沒有空位了，如圖 14(h)就是將要被廣播的排程結果。

Storage	Request	Waiting time (time slot)	Request length ( $l_i$ )
$s_1 = \{d_1\}$	$q_1 = \{d_2, d_3, d_4\}$	1	3
$s_2 = \{d_1\}$	$q_2 = \{d_5\}$	2	1
$s_3 = \{d_2\}$	$q_3 = \{d_1, d_3\}$	4	2
$s_4 = \{d_1\}$	$q_4 = \{d_2, d_5\}$	1	2

(a) 使用者 request-stored 的資料與 wait time 和 request length

Request	Store( $s_i$ )	Req( $q_j$ )	Product $p_i$
$q_1$	$3/4$	$5/4$	$5/16$
$q_2$	$3/4$	$1/2$	$3/4$
$q_3$	$1/4$	$3/4$	$3/8$
$q_4$	$3/4$	1	$3/8$

(b) 計算每一個 request 的  $p_i$  值

Encoding data $e_j$	Dem( $e_j$ )	Total waiting time	$DI(e_j)$
$d_1 \oplus d_5$	$\{u_2, u_4\}$	3	6

(c) 取  $u_2$  的 request-stored 資料作編碼，並計算編碼的權重  $DI(e_j)$

$d_1 \oplus d_5$				
------------------	--	--	--	--

(d) 使用者  $u_2$  request-stored 的資料只有一個配對

Encoding data $e_j$	Dem( $e_j$ )	Total waiting time	$DI(e_j)$
$d_1 \oplus d_2$	$\{u_1, u_3, u_4\}$	6	18
$d_2 \oplus d_3$	$\{u_3\}$	4	4

(e) 取次高  $p_i$ ，若  $p_i$  相同隨機取一，在此取  $u_3$ ，計算每一編碼的權重  $DI(e_j)$

$d_1 \oplus d_5$	$d_1 \oplus d_2$	$d_2 \oplus d_3$		
------------------	------------------	------------------	--	--

(f) 依  $DI(e_j)$  權重大小排入

Encoding data $e_j$	Dem( $e_j$ )	Total waiting time	$DI(e_j)$
$d_1 \oplus d_2$	$\{u_1, u_3, u_4\}$	6	18
$d_1 \oplus d_3$	$\{u_1\}$	1	1
$d_1 \oplus d_4$	$\{u_1\}$	1	1

(g)  $u_4$  的編碼皆已出現過，在此取  $u_1$  的 request-stored 編碼

$d_1 \oplus d_5$	$d_1 \oplus d_2$	$d_2 \oplus d_3$	$d_1 \oplus d_3$	$d_1 \oplus d_4$
------------------	------------------	------------------	------------------	------------------

(h) 這例子的廣播排程

圖 14·實例說明 Modified OE

圖(14)用來說明 Modified OE 演算法與圖(6)用來說明 OE 演算法的例子皆採用同一個例子，因此我們可以去比較這兩個例子所產生的排程結果。

根據圖(14)與圖(6)兩種方法採用相同例子所產生的排程結果，我們比較 Client 對兩種排程結果的存取時間，我們得到 Modified OE 方法的存取時間為  $5 + 1 + 3 + 2 = 11$ ；而 OE 演算法的存取時間則為  $3 + 4 + 5 + 4 = 16$ 。由此我們可以得知，Modified OE 的方法能有效的減少 Client 的存取時間。

## 第五章 模擬實驗

為了驗證本研究所提出來的 Modified OE 演算法比 OE 演算法有比較好的效率，可以減少 Client 端的平均存取時間，我們以模擬實驗的方式呈現。以下將分別說明模擬環境、參數和最後的實驗結果。

### 第一節 模擬環境

本研究模擬環境是使用 Visual Basic 6.0 (VB6.0)作為模擬程式的撰寫語言，VB6.0 的優點在於入門語法簡單，並且可快速的建立物件，可以減少開發的時間。因此 VB6.0 可以快速的幫助我解決問題。

## 第二節 模擬參數

模擬環境的參數，使用下列表格的參數。

表 3：模擬實驗參數

參數	預設值	變動範圍
Number of Data Item	100	--
Broadcast Cycle	100	50 ~ 200
Arrival Rate	5	5 ~ 20
Requested Length	10	5 ~ 20
Stored Length	10	5 ~ 20
Zipf	0.8	0 ~ 1

本研究將實驗、對照 Modified OE 與 OE 演算法在 Zipf、Mean Arrival Rate、Broadcast Cycle、Requested Length、Stored Length 這 5 種參數產生變化時的平均存取時間。在實驗時的參數設定如表 4、表 5、表 6、表 7、表 8。

表 4、比較 Zipf 分配的參數表

參數	設定值
Number of Data Item	100
Broadcast Cycle	100
Arrival Rate	5

Requested Length	10
Stored Length	10
Zipf	0.2 0.4 0.6 0.8

表 5、比較 Mean Arrival Rate(request/time slot)分配的參數表

參數	設定值
Number of Data Item	100
Broadcast Cycle	100
Arrival Rate	5 10 15 20
Requested Length	10
Stored Length	10
Zipf	0.8

表 6、比較 Broadcast Cycle 的參數表

參數	設定值
Number of Data Item	100
Broadcast Cycle	50

	100 150 200
Arrival Rate	5
Requested Length	10
Stored Length	10
Zipf	0.8

表 7、比較 Requested Length 的參數表

參數	設定值
Number of Data Item	100
Broadcast Cycle	100
Arrival Rate	5
Requested Length	5
	10
	15
	20
Stored Length	10
Zipf	0.8

表 8、比較 Stored Length 的參數表

參數	設定值
Number of Data Item	100

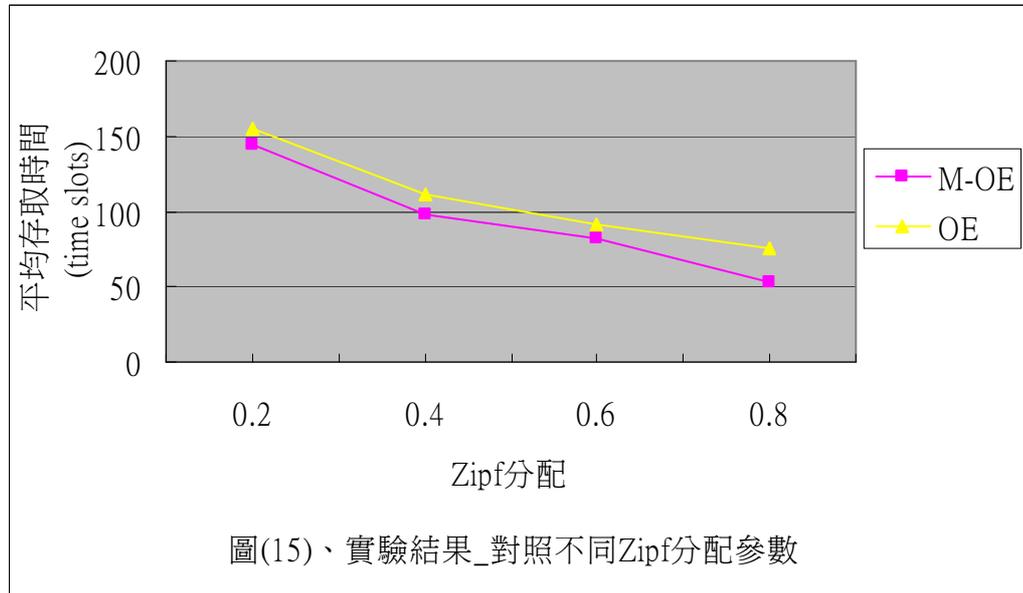
Broadcast Cycle	100
Arrival Rate	5
Requested Length	10
Stored Length	5
	10
	15
	20
Zipf	0.8

### 第三節 實驗結果

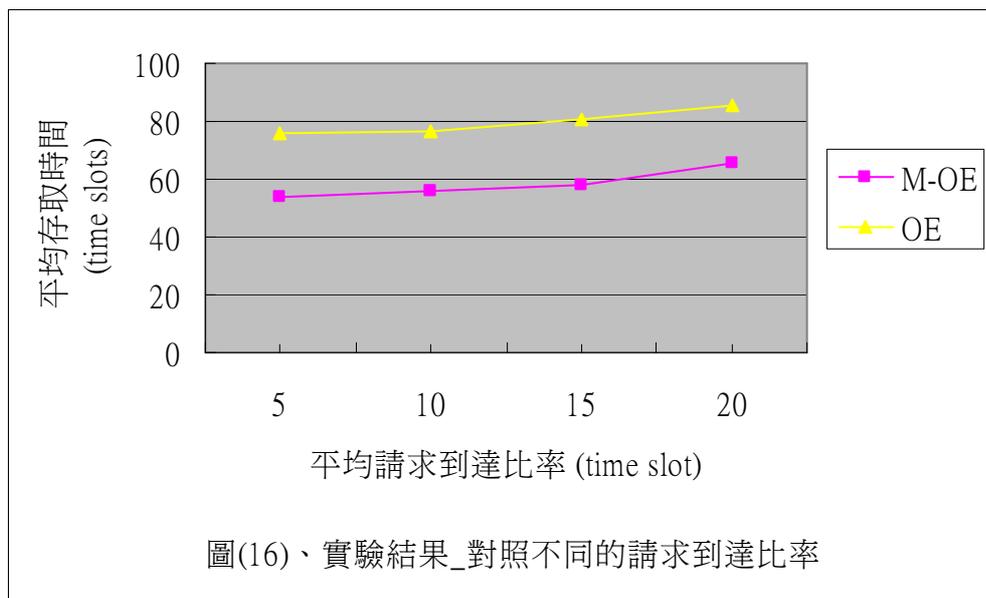
模擬實驗比較 Modified OE 演算法與 OE 演算法的平均存取時間。我們分別測試當 Zipf 分配、Mean Arrival Rate (每個時間單位請求到達的比率)、Broadcast Cycle(廣播週期)、Requested Length(請求長度)、Stored Length(存取長度)這 5 種參數有變化時 Modified OE 演算法與 OE 演算法的實驗結果。模擬實驗結果方形的點為 Modified OE (M-OE)演算法；三角形的點代表 OE 演算法。

從(圖 15)來看改變 Zipf 分配對平均存取時間的影響。Zipf 越高代表資料項冷熱門程度分配的斜率越大；Zipf 越小代表資料項冷熱門的斜率越平緩，也就是說資料項的冷熱門之分不明顯。由圖我們看到 Zipf 等於 0.8 時平均請求時間是最低的；當 Zipf 等於 0.2 時平均請求時間最高；因此我們得知當資料項冷熱門程度越

高平均存取時間越低。在 Zipf 參數的實驗下，我們的方法在實驗結果上比 OE 方法少約 10~25 的平均存取時間。

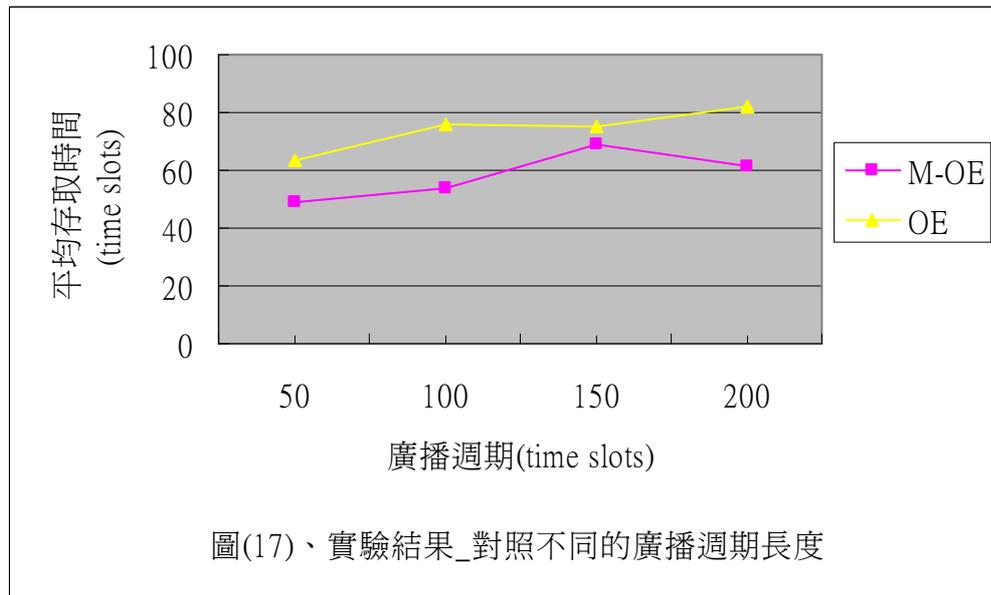


Mean Arrival Rate(request/time slot)指的是每個時間單位請求到達 Server 端的個數。由(圖 16)我們得知當請求到達比率越高，平均存取時間遞增。我們的方法在不同的請求比率上皆比 OE 方法少 25 左右的平均存取時間。

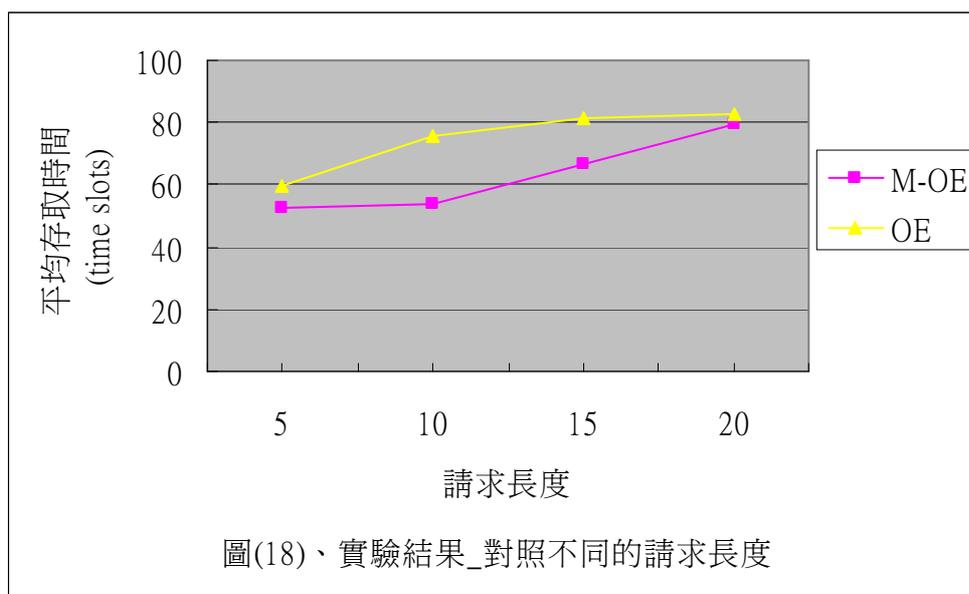


在(圖 17)去對照不同的廣播週期長度平均存取時間的情形。

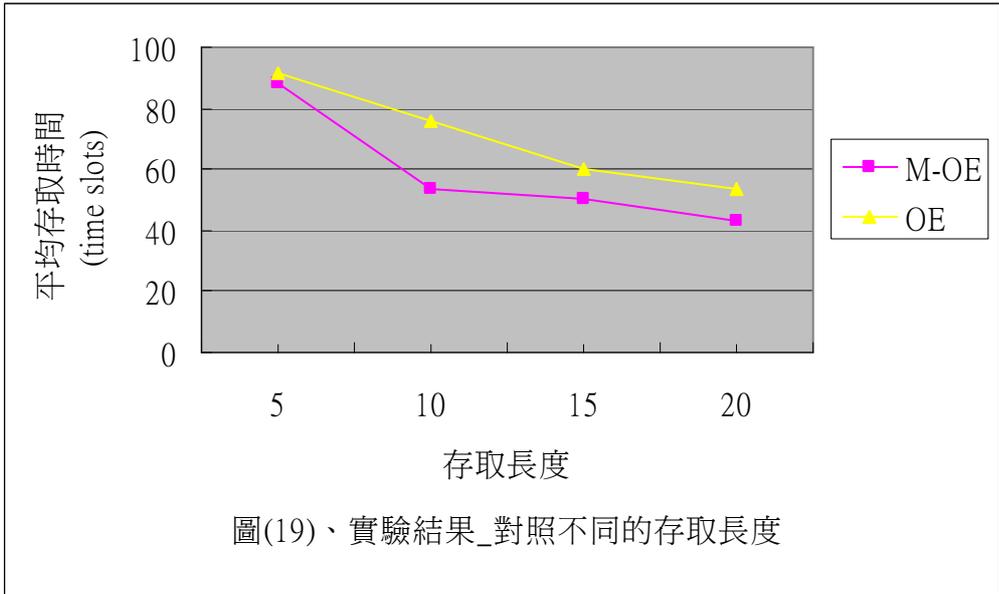
由(圖 17)我們得知當廣播週期越長，平均存取時間會遞增。但我們的方法還是勝過 OE 方法。



(圖 18)是在請求長度變化的比較，請求長度指的是一個請求所能包含的資料項個數，由圖得知當請求長度越長平均存取時間會遞增。當請求長度等於 20 時 M-OE 與 OE 方法的平均存取時間很接近是因為請求的資料項變多，因此 M-OE 滿足請求的效率變低。



(圖 19)是比較存取長度，存取長度指的是 Client 能 Cache 資料項的多寡，Cache 的資料項越多 Client 端就有較高的機率可以從 Cache 中取得所需的資料項，如果 Client 從 Cache 得到滿足，Client 端就不用向 Server 端發出請求。因此存取長度越大平均存取時間越低；反之存取長度小，平均存取時間就高。如(圖 19)，我們的方法還是勝過 OE 方法。



## 第六章 結論

在無線網路的介紹，我們說明了無線網路的架構；在相關文獻部份，我們介紹在不同的條件限制下先前研究所提出來的演算法；問題描述部分，我們說明先前演算法未考慮到的地方；最後本研究提出了一個 Modified OE 方法去改善先前研究所提出的 OE 方法，在多資料項請求的請求式廣播系統環境下。

本研究提出了一個 Modified OE 方法的與 OE 做比較，我們所提出的 Modified OE 方法確實可以使 Client 端得到較少的平均存取時間。我們的方法是採用 OE 編碼技術為基礎，在演算法運作時 Server 從還未被服務的請中挑選要服務的請求時加入了請求長度的考量；選擇出請求後再進行編碼與排程。我們的方法與 OE 在不同的參數環境下做模擬實驗，實驗結果顯示我們方法的平均存取時間約比 OE 方法低 20% 左右。

## 參考文獻

- [1] G. Lee, M. Yeh, S. Lo, and A. Chen. "A strategy for efficient access of multiple data items in mobile environments." In Proceedings of the 3rd International Conference on Mobile Data Management, pages 71–78, 2002.
- [2] Shih-yang Lin. "A Dynamically Schedule Algorithm for Wireless Broadcast." June 2005.
- [3] Chung-Hua Chu, De-Nian Yang and Ming-Syan Chen. "Multi-data Delivery Based on Network Coding in On-demand Broadcast." Mobile Data Management, 2008. MDM 08. 9th International Conference on 27-30 April 2008 Page(s):181 - 188
- [4] Weiwei Sun, Zhuoyao Zhang, Ping Yu, Yongrui Qin. "Skewed Wireless Broadcast Scheduling for Multi-Item Queries." Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on 21-25 Sept. 2007 Page(s):1865 – 1868
- [5] Zhuoyao Zhang, Weiwei Sun, Ping Yu, Yongrui Qin. "Preemptive Scheduling for Multi-item Queries in On-demand Data Broadcast." Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for 18-21 Nov. 2008 Page(s):610 – 615
- [6] Sang Hyuk Kang, Sujeong Choi, Seong Jong Choi, Gwangsoon Lee, Jaegug Lew, Jun Lee. "Scheduling Data Broadcast Based on Multi-Frequency in Mobile Interactive Broadcasting." Broadcasting, IEEE Transactions on Volume 53, Issue 1, Part 2, March 2007 Page(s):405 - 411
- [7] Sujeong Choi, Sang Hyuk Kang, Yoon Goo Nam. "A Hybrid Broadcast Scheduling for Mobile Data Broadcasting with Return Channels." Communications, Computers and Signal Processing, 2007. PacRim 2007. IEEE Pacific Rim Conference on 22-24 Aug. 2007 Page(s):549 – 552
- [8] D. Aksoy and M. Franklin, "R×W: A Scheduling Approach for Large-Scale On-Demand Data Broadcast." IEEE/ACM Transactions on Networking, 7(6) 1999, pp.846-860.
- [9] J. Xu, D. Lee, Q. Hu, and W. C. Lee. "Data Broadcast", In Handbook of Wireless Networks and Mobile Computing." John Wiley & Sons. 2002.
- [10] Jun Chen, Kai Liu, Victor C.S. Lee. "Analysis of data scheduling

- algorithms in supporting real-time multi-item requests in on-demand broadcast environments.” Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on 23-29 May 2009 Page(s):1 – 8
- [11] Ye-In Chang, Wu-Han Hsieh. ” An efficient scheduling method for query-set-based broadcasting in mobile environments” Distributed Computing Systems Workshops, 2004. 24th International Conference on 2004 Page(s):478 – 483
- [12] Guanling Lee, Meng-Shin Yeh, Shou-Chih Lo, Chen A.L.P. ” A strategy for efficient access of multiple data items in mobile environments” Mobile Data Management, 2002. Proceedings. Third International Conference on 8-11 Jan. 2002 Page(s):71 – 78
- [13] Yon Dohn Chung, Myoung Ho Kim. “QEM: a scheduling method for wireless broadcast data” Database Systems for Advanced Applications, 1999. Proceedings., 6th International Conference on 19-21 April 1999 Page(s):135 – 142
- [14] Li Jian Pan, Tung-ying Lee. “Data Dispose Algorithm for Scheduling in Multi Channels” Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on Volume 1, 17-20 Feb. 2008 Page(s):555 – 560
- [15] Shenggang Wan, Qiang Cao, Xubin He, Changsheng Xie, Chentao Wu. “An Adaptive Cache Management Using Dual LRU Stacks to Improve Buffer Cache Performance” Performance, Computing and Communications Conference, 2008. IPCCC 2008. IEEE International 7-9 Dec. 2008 Page(s):43 – 50
- [16] Chung-Hua Chu, De-Nian Yang, Ming-Syan Chen. “Using Network Coding for Dependent Data Broadcasting in a Mobile Environment” Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE 26-30 Nov. 2007 Page(s):5346 - 5350