

Adaptive-aware Wireless Broadcast Systems

林聯發

高苑科技大學資訊傳播系，成大資工所博士班

lienfa@cc.kyit.edu.tw

摘要

在無線資訊系統中，廣播磁碟技術已經變成一個熱門的資料散播的方式。然而，因行動裝置的電力有限、無線環境連線成本較昂貴，不適合長時間的資訊抓取。另外行動使用者具移動性往往不會在基地台的服務範圍待太久，造成資訊的服務無法持續太久。因此無線環境設計一個高效率資料廣播系統有別於傳統的設計方式，必須就節省電源與行動使用者必須在有限的時間存取資料的特性加以考量。

本論文提出了一個完整的機制來解決行動通訊使用者在行動環境中如何有效率的存取廣播的資料。我們的目標是發展一種利益導向(benefit oriented)的資料擷取協定，讓行動通訊使用者能在限定的時間內以消耗更少電力的方式，取得對行動使用者而言更多有價值的資料。另外我們進行了大量的實驗來評估我們所提方法的效能。

關鍵詞：行動計算，資料管理，無線資訊系統，廣播磁碟，效能評估，演算法設計與分析。

1. Introduction

電腦軟硬體以及無線通訊技術的進步，導致行動資訊系統的快速發展。這樣的環境，server到client端上傳頻寬遠大於client到server端的下傳頻寬的特性，我們稱這為非對稱的特性。許多研究資料管理的研究學者發現，在這個非對稱特性環境下，資料的傳輸如果使用現在的方式(i.e.;傳統client-server model)，系統的效率會十分差。一個解決這種窘境的方法就是使用資料廣播的方法來克服這種特殊的環境。所謂的資料廣播的方法就是，當行動客戶端(mobile user-MU)需要資料的時候，不要馬上透過上傳頻道把要求傳送給伺服器，而是行動客戶端要先到廣播頻道中監聽廣播伺服器廣播的資料。若是有聽到想要的資料，那麼行動客戶端就少發出一次要，所以減少上傳頻道的使用量。因此，資料廣播變成在無線通訊網路的環境下一種有效而且重要的手段來服務行動使用者 [4, 6, 13]。

無線資料廣播的服務已經被使用在商用產品有多年了，例如 StarBand [17] 與 Hughes Network [18] 即是此類著名的商品。而 Microsoft 在 2003 年在國際消費電子展中宣佈的 Smart Personal Objects Technology (SPOT) 的 DirectBand Network [6] 更進一步確定企業界對於能便利的利用無線廣播來提供無處不在隨時隨地的資訊服務深感興趣。

無線資訊系統由於環境與行動裝置本身的特性，其資訊擷取 (information retrieval) 有以下的特性，(1) 行動裝置存空間有限，沒有辦法儲存太多資料。(2) MU 有 mobility 的特性，MU 經過一段時間的移動，可能就會切換到另一基地台 (base station) 的服務範圍，新的基地台服務的資料可能有所不同或者新舊基地台廣播節目的排程時序不一致，這時 MU 將無法抓取所需資料，故 MU 必須在限定的時間完成資料的存取。(3) 行動裝置電力有限且連線費用通常是較昂貴，不宜也沒辦法一直連線抓取資料。因此無線環境設計一個高效率資料廣播系統有別於傳統的設計方式，必須就節省電源與 MU 能在有限的時間存取資料的特性加以考量。

另外，行動裝置日益普及以及全球 3G 無線網路基礎建設的廣佈 [17]，可以預期的將會有越來越多的行動使用者。因此對於能同時服務多人節省網路資源以及滿足各式各樣資訊需求的無線資訊系統也會越來越多。舉一個與我們每天生活息息相關的新聞 (news) 服務而言，新聞讓我們了解每天週遭所發生的事情，是我們每天獲得重要資訊的來源。針對新聞這類的資訊，client 在讀取新聞的時候，client 端在尚未抓取資料前，並不清楚 browser 會存取哪些的新聞。等到 browser (news reader) 連上新聞網站下載網頁後，client 才能知道有哪些新聞可以閱讀。同樣的，因 MU 有 mobility 的特性，不同的基地台提供不同的資訊服務，故 MU 往往沒辦法事先得知目前所處位置的基地台提供資訊的內容，若沒有任何輔助資訊，MU 只能持續連線聽取廣播資料。故 client 無法事先知道 broadcast server 廣播資料為何亦必需加以考量。

目前在 broadcast 環境下已經有許多論文討論資料的組織以及索引的論文，這些成果對於單一資料提供了的有效率的存取方式。然而，就我們知識所及，目前並沒有任何一個研究曾經考慮設計一個有效率的資料廣播系統處理在使用者無法事先知道廣播內容以及 MU 必須在有限的時間存取資訊的問題。本論文針對上述問題，設計一套有效率的資料廣播策略讓 MU 可以更省電的方式抓取更重要的資訊。本研究的成果不但拓展了過去研究的角度，也提供一個更真實且完整的無線應用程式發展的模型。本論文的技術貢獻可以歸類成下面四點：

- 提出無線應用程式的新存取模型。
- 設計支援有效率的存取機制。
- 設計接近最佳化的計算及最有效率存取模式的演算法。
- 完整的實驗結果支持提出的機制及演算法。

本論文架構如下：第二節為 Related Works。第三節說明 News Broadcast Server 的 broadcast strategy。第四節則介紹如何找出 optimal access behaviors 的演算法。第五節介紹 Performance Study 的相關設定與實驗結果。第六節為 Conclusions and Future Works。

2. Related Works

在[19]中提出許多行動計算環境的特性，以及許多資料庫領域相關的議題。其中資料傳遞 (data dissemination) 是一個十分新穎且在傳統環境下很少被深入討論。Tomasz Imielinski 等人指出在無線網路中，由於硬體的限制而造成下傳頻寬遠大於上傳頻寬[9]。因此，在這個環境中，客戶端最好不要常常送出要求而造成上傳頻寬的浪費。為了達到這個目的，專家們建議資料傳送用廣播的方式。也就是伺服器不停的利用大量的下傳頻寬把資料在廣播頻道中播出，當客戶端要求資料的時候，客戶端並不馬上送出要求，而是先到廣播頻道中監聽是否有自己想要的資料。等確定自己想要的資料不在廣播中再對伺服器送出要求。利用這種廣播的方式，許多使用者就可以直接從廣播頻道中拿到自己想要的資料而不浪費任何上傳頻寬。

但是，廣播的方式可能會造成使用者的存取時間(access time)大量增加。為了減緩這個問題，Swarup Acharya 等人提出 Broadcast Disk 的方法有效的降低存取時間[1, 2]。這個方法的主要想法就是提高熱門資料的廣播次數，而降低冷門資料的廣播次數。Broadcast Disk 的方法十分簡單而且有效率，之後的研究有許多都是根據這個方法做延伸。在這些延伸的研究中大致可以分成兩類。

■ 在客戶端增加機制再降低存取時間。

這類的方法主要利用 prefetching、caching 或是 push and pull bandwidth 的平衡。Swarup Acharya 等人提出一個稱為 *PIX* 的 cache 方式來降低存取的時間。*PIX* 會根據資料的存取機率以及資料廣播的頻率在 cache 中挑選一個最少會被用或是即使很常用但很快會再被播出的資料當犧牲者從 cache 中移除，而把空出來的位置來放置目前抓取的資料能改善存取效率的資料[2]。Swarup Acharya 等人也提出利用 backchannel 來平衡 push channels 和 pull channel 的使用量[3]。Quinglong Hu 等人[8]則是整合快取以及 pull-based 點對點資料傳輸來達到降低存取時間的目的。

■ 降低客戶端電源的使用

這一類的方法是在廣播節目中使用 index 技巧以提供客戶端可以用比較低的電源讀取資料。主要是在減少客戶端讀取資料時所花費的 tuning time [8, 11]。這類的研究主要在讓行動裝置在廣播不是自己感興趣的資料是進入睡眠模式 (doze mode)，等到自己感興趣的資料廣播時才切入廣播

頻道中監聽資料。這些研究成果雖然也是在節省電力的前提下進行研究，但有這些論文跟本研究的重點有幾點不同：(1) index 技術沒考慮針對個人最佳化，(2)兩者的應用環境不同。上面 index 相關文獻是在處理如何有效率的抓特定一個 data，我們的 work 並不是在處理如何存取特定的 data，而是在固定的工作時間(turnaround time T)的時間內讓 client 可以更省電的方式抓到更重要資料。

3. Broadcast strategy for the News Broadcast Server

資料在廣播頻道播放的順序安排，也就是所謂的廣播節目(broadcast program)將影響 MU 抓取資料的效能。因此 broadcast server 必需提供一個良好的廣播策略讓 MU 可以在有限時間內以更省電的方式抓取更有價值的資料。

3.1 Naive Broadcast

Naive broadcast 是最簡單的傳送方式。Broadcast server 直接把 news 透過 broadcast channel 廣播。很明顯的，naive broadcast 不是最好的 Newscast 方式。因為他並沒有提供任何的訊息讓使用者可以有效率(更省電或是更高品質的資料)存取。在 naive broadcast 中，使用者存取 news 的方法為 fully download 或 random download。然而，這兩者皆不是有效的方法。因為 broadcast program 並沒有提供任何的輔助資訊。MU 只能 fully download 持續抓取資料，直到抓取所需的 data 為止。這樣的方式，很明顯不是很有效率。這將造成 MU 花費太多的電力抓取不需要的資料。random download 任意抓取的方式，代表 MU 可自行任意調整要 active 抓取資料，以及進入 doze mode 省電模式的時間。這樣的行為，MU 只花部分的時間抓取資料，故比較省電。然而，任意抓取資料的結果，所抓取的資料，可能會抓取到部分或全部不是 MU 所需的資料，這樣只是為了省電而漫無目標的資料抓取，很顯然亦是一個效率很差的存取方式。

3.2 Index-involving Broadcast

Index-involving broadcast 是在 broadcast program 加入一些 index data，提供 MU 關於每筆 data item 在 broadcast program 播放的時間，至於如何將 index 資料加入 broadcast program 的方法，包括有 Distributed Index, Flexible Index [9, 10]，而(1:m)indexing[10]的方法，是將 broadcast program 分成 m 等份，在每一等份的前面加入一份 index 資料，用來降低 initial probe waiting time。當 broadcast program 包含 index 資訊時，則 MU 存取資料的方式如以下的步驟：

步驟 1: MU 必須先抓取 index 資料，才能知道所需時間在 broadcast program 的播放時間。因此 MU 必須先進入 doze mode 直到 broadcast index 資料，這一段時間即是 initial probe waiting。

步驟 2: 當抓取完 index 資料 MU 進入 doze mode 直到所需資料廣播時才 active 抓取資料。很明顯利用上述的步驟來抓取資料，要比起 naive broadcast 的抓取方式有效多了。當所需資料 broadcast 時 MU 才 active 抓取資料，其他時間則進入較省電的 doze mode。不過 index 的方法只適用在 MU 明確資料 broadcast server 所播的資料為何，而且所有 users 只存取單一筆資料或一群相關的資料，這樣 server 可根據這單一筆資料或多筆相關資料給一個 key 值來製作 index，再將 data 根據 key 值依序安排在 broadcast program 上播放。

就 news 的資料而言，MU 並不知 broadcast server 將播放哪些 news 資料，對大部分 news reader 而言，閱讀 news 的習慣是先閱讀最重要最熱門 news，而且閱讀資料的行為往往與閱讀的時間多寡有關。一般會先讀取新聞標題，若時間上比較充裕會再看各 news 的詳細內容，當時間更多時則再看一些較不重要的社會新聞或花邊新聞。也就是使用者所要的資料量並不是固定的。Broadcast server 沒有辦法按照各 news 給一個 key 製作 index，因為這樣的 index 對 news reader 並沒有幫助，而是應該根據各 news 的受歡迎程度製作 index，如此才能符合 news reader 的行為。因此我們將傳統根據 data 的 key 來製作 on air index 的方法做一改良，將資料依其受歡迎程度 (access frequency) 製作 index，稱為 BNews-Index。此 index 的架構如圖 1 所示。index 的 data structure 包括 access frequency (AF)，broadcast packet id (BID)，與 broadcast time (BT) 三個欄位。其中 AF 代表該資料的重要性，BID 代表該資料的 broadcast packet 編號，BT 代表該資料從 index packet 後算起在 broadcast program 的廣播時間。假設 MU 進入 broadcast channel 的時間為 t_s-5 ，initial probe waiting time 為 5，MU 的 turnaround time T 是 12，則 MU 抓取資料的行為將如圖 1 所示。將會抓取 packet 1, 2, 3, 6, 7。其他時間則進入 doze mode。因為 MU 會先抓 AF 最高的資料 packet 1，等 packet 1 抓取完畢後再抓 AF 次高的 packet 2，packet 3，再抓取次高的 packet 6, packet 7，在 packet 7 turnaround time 用完，停止抓資料。

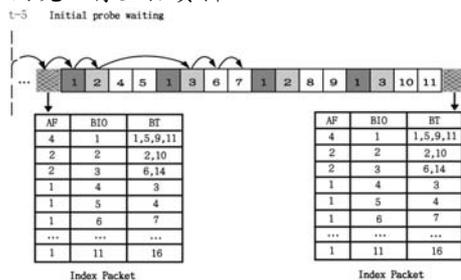


圖 1. Data structure of the Bnews-Index。

3.3 Adaptive-aware Broadcast

在 BNews-Index 的方法除了為節省 tuning time，增加 index 資料帶來 initial probe waiting 的問題外，就考量到 total power consumption 與 total 抓取 news 的總重要性並不是 optimal。就 initial probe waiting time 而言將造成 mobile client news reader (MCNR) 會浪費部份可用的時間資源以及可能錯過一些重要的資料沒辦法抓取。因為 MCNR 會設定 retrieval news data 的時間 T，就 initial probe waiting 而言，這是 MCNR 等待抓取 index data 所需花費的時間。這段時間 MCNR 就只能 waiting 不能抓取任何資料，故將耗用掉 client news reader 部份可用的時間，卻對 news data 的抓取毫無貢獻。另外，因為 MCNR 必須等到抓取 index data 才知道在 broadcast program 中各個 time slot 播放資料的重要性，故就算 MCNR tune into channel 的時間正在廣播極為重要的資訊，MCNR 依然沒辦法抓取。若是 T 的時間夠長，則必須等到下一 broadcast cycle 才能抓取，若 T 的時間不夠長則將完全錯過抓取該筆資料的機會。再就 total power consumption 與 total 抓取資料的重要性而言，若重要性相同的兩筆 news 的 broadcast 時間相距很遠，則 MCNR 將進入很長的 doze mode，doze mode 雖然較省電但是仍會耗電，而且 doze mode 並不能抓取任何資料。就有效電力使用與抓取資料總重要性的角度，這並不是一個有效率的方法。為了有效處理 BNews-Index 的問題，我們提出 Adaptive Broadcast 的 approach。

我們的方法是在每一個 broadcast data page 包括了 data page 的 content 以及一個稱為 Flexible Access Mode Indicator (FAMI) 的 small table，FAMI table 提供了 MCNR 該如何有效存取 broadcast data 的有價值資料。MCNR 一進入 broadcast channel，可根據其 turnaround time T 的設定，馬上找到 T 所對應到 MCNR 該 follow 的 access behavior。因此不會錯過任何一筆重要的 news data，而且可以對有限的時間 T 做最佳的利用，當 MCNR 進入 doze mode 一定是該 time slot 播放的 news data 是較不重要。FAMI table 的結構由 turnaround time (T) 與 access behavior 兩個欄位組成如圖 2 所示。

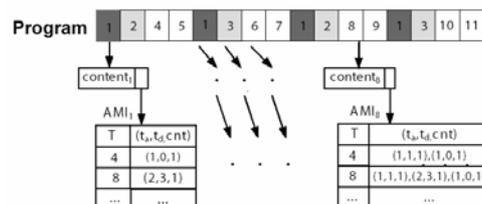


圖 2. The data structure of flexible access mode indicator (FAMI)。

T 代表client從進入廣播頻道開始聽取資料直到client接收完所需資料，離開系統的這一段工作時間。access behavior是由多組 (t_a, t_d, cnt) 組成的集合，其中 t_a 代表MU將active持續抓取資料的時間， t_d 則代表進入doze mode持續的時間， (t_a, t_d) 稱為access mode， (t_a, t_d, cnt) 則代表MU將重複 (t_a, t_d) 的access mode 抓取行為 cnt 次。Flexible Access Mode Indicator 是由server端根據不同turnaround time T ，以及個別 data page 的播放時間，計算出不同的 (T, t_a, t_d, cnt) 組合所能提供MU最佳的access behavior，當server generator broadcast program時，將access behaviors伴隨data content放入同一packet 播出。以圖2為例，若MU進入public channel開始聽取資料的時間 t_s 為8， $T=4$ ，則client從抓取到的data packet 查詢FAMI table的 T 欄位，可以得到MU該follow的access behavior為 $(1, 1, 1)$ ， $(2, 3, 1)$ ， $(1, 0, 1)$ 。因此MU將採取 $(1, 1)$ 的access mode 一次，接下來 $(2, 3)$ 的access mode 一次，最後 $(1, 0)$ 的access mode 一次，就能以最省電的方式抓取較熱門的資料。

4. Find Optimal Access Behaviors of Mobile Users

從MU的觀點，對MU最有利的方式就是能利用最少的power consumption，在有限的時間內能從broadcast data 抓取更多更重要的資料，而不管MH是從哪一個進入點開始聽取資料。power consumption 以及抓取資料的重要性是判斷MU選取的access behavior pair 有多好的指標。基本上，power consumption 愈小，代表所選的access behavior pair 越好。抓取資料的重要性愈高亦代表所選的access behavior pair 愈好。有鑑於此，我們觀察到兩個重要的規則。

規則 1: MU active 聽取資料的次數愈小愈好。因為active的次數愈小，相對也愈減少power consumption。也就是MU可以較少的power consumption 抓取所需資料。

規則 2: MU active 時所抓取的news資料，愈熱門愈好。因為愈熱門的資料，代表該news 愈重要，代表MU更能有效利用有限的電力。我們將利用這兩項基本規則設計FAMI Approach 的演算法。我們將FAMI Approach分成Greedy Phase與Refine Phase。首先，我們將news依重要性的大小排序，依據MUS設定的有限抓取時間turnaround time T ，從AF最高的news開始抓取，一直抓到滿足 T 為止。這樣我們可以得到所抓取的news以及抓取那些news所耗的電力。因為broadcast program 的產生是根據broadcast disks 應該抓取哪些資料能滿足以最省電的方式抓取更多更重要資訊。這些該抓取的資料可以換算成所謂的DiskAccessMode $(num_1, num_2, \dots, num_k)$ 其中 num_i 代表disk i 要抓取的pages

數。DiskAccessMode代表MU必須從不同的broadcast disk各抓取多少pages資料才能滿足的需求。然後根據MU的 t_s 可將DiskAccessMode轉換成access behavior pairs，根據access behavior pairs就可算出所需的power consumption，得到一組近似解，這是我們的Greedy Phase。Greedy Phase從AF最高的資料開始抓取，因此能以最少次的active 抓取滿足查詢的資料。然而Greedy Phase的方式只能保證active的次數最少，但卻不能保證最省電。因為在抓取最重要news資料的過程，上一次的抓取的data page 與下一次欲抓取的data page可能相距很遠，也就是要抓下一個資料必須經過很長的doze mode，反而造成更多電力的浪費，或許在仍然滿足需求更多更重要資料的前提下，可直接抓取多一些次高重要的data pages。雖然多花了一些active的時間，然而卻能在總抓取資料的價值不減少的情形下使得耗電力反而能減少，這就是我們的Refine phase。詳細的演算法如Algorithm 1以及Algorithm 2所示。其中DiskNum(t_s)函數計算 t_s 所在位置的page屬於那一個disk。

```

INPUT:  $E, t_s, T$ ;
OUTPUT: NearOptimalPower;
1: initial DiskAccessMode[ $num_1, num_2, \dots, num_k$ ] LIST;
2: initial AccessModes[ $t_a, t_d, cnt$ ] LIST;
3: PageNumber=0;
4: initial  $disk_i=1$ ; //initial from the highest frequency disk
5: while  $t_s \leq T$  do
6:   if DiskNum( $t_s$ ) =  $disk_i$  then
7:     if PageNumber <>  $num_i$  then
8:       PageNumber = PageNumber + 1;
9:     else
10:       $disk_i = disk_i + 1$ ;
11:      pageNumber = 0;
12:    end if
13:  end if
14:   $t_s = t_s + 1$ ;
15: end while
16: AccessModes[] = TranslateAccessMode( $t_s, DiskAccessMode[]$ );
17: NearOptimalPower = PowerConsumption(AccessModes[]);

```

Algorithm 1: Greedy Phase Algorithm

```

INPUT:  $E, t_s, NearOptimalPower, DiskAccessModes(num_1, num_2, \dots, num_k)$ ;
OUTPUT: MinPowerConsumption;
1: initial Frequency=0;
2: Compute Frequency of the DiskAccessModes( $num_1, num_2, \dots, num_k$ );
3: initial  $disk_k = 2$ ; // initial from the second highest frequency disk
4: for  $disk_k = 2$  to  $k$  do
5:   while DISK ( $disk_k - 1$ ) IS Not EMPTY do
6:      $num_{k-1} = num_{k-1} - 1$ ;
7:     while (Frequency  $\leq$  TotalFetchFrequency) and (PowerConsumption  $\leq$  NearOptimalPower) do
8:       if DISK  $disk_k$  Is Not FULL then
9:          $num_k = num_k + 1$ ;
10:        AccessModes[] = TranslateAccessMode( $t_s, DiskAccessMode[]$ );
11:        PowerConsumption = PowerConsumption(AccessModes[]);
12:        if PowerConsumption  $\leq$  NearOptimalPower then
13:          NearOptimalPower=PowerConsumption;
14:        end if
15:      else
16:         $disk_k=disk_k + 1$ ;
17:      end if
18:    end while
19:  end while
20: end for

```

Algorithm 2: Refine Phase Algorithm.

5. Performance Study

5.1. Experiment Design

圖 3 描述我們 performance study 的 simulation mode。行動資訊系統由 broadcast server 與 mobile client 兩個部份構成。詳細的運作情形說明如下。**Broadcast Server**: 在圖 3 中 broadcast server 下方是一包含所有要 broadcast data 的 database。這些 data pages 將被利用來產生我們實驗中所需的 broadcast program。產生 broadcast program 後送到下一模組 Index or FAMI generator，這 Index or FAMI generator 則跟據 Index or AMI approach 的不同，分別將 index data 加到 broadcast program 成爲 index packet，以及將由不同 T 組成的 FAMI table 加到 broadcast program 中每一個 data page。Index or FAMI generator 跟據 index 與 FAMI 不同的 approach 加完 index data or FAMI table 後的 broadcast program 則可以送上 public channel broadcast 給 MU。

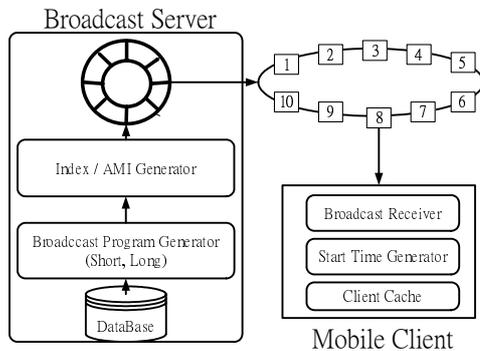


圖 3: Simulation Mode

表格 1 是 broadcast server 相關參數表格，說明有關我們 performance evaluation 中所使用的參數與其代表的意義。NumDisks 代表 broadcast server 中 broadcast disk 的個數，設定值爲 5。DiskSize 代表每一個 disk 的 page 數。至於每一個 disk 包含的 page 數如表 1 所示，其基本原則是 access probability 愈 hot 的 disk 其 page 數愈少，access probability 愈 cold 的 disk 其 page 數愈多。至於如何計算出每一 disk 所含的 page 數則是根據每一 disk 的 DiskFreq。按照產生 broadcast disks program 的程式產生，詳細的演算法可參考[1, 2]。

表格 1：broadcast server 的參數與設定。

Parameter	Description	Settings
NumDisks	廣播磁碟個數	5
DiskSize	各磁碟的大小	35, 90, 315, 420, 730
DiskFreq	各磁碟的廣播頻率	9 : 7 : 5 : 3 : 1

Mobile Client: 圖 3 的 broadcast receiver 是 client 用來從 wireless channel 抓取 broadcast pages。這些被抓取的資料被 client cache 單元管理與儲存。start time generator 用來產生不同的 start time (t_s) 與 MU turnaround time T。實驗中 client 端所用到的參數與預設值。T 的設定範圍從 100 到 1600。基本上 start time t_s 可以是整個廣播週期的任一時間點，但這將會造成 t_s 有太多的選擇，將造成實驗太複雜。爲了簡化此一工作，我們只考慮 MU 只從 hot, medium, cold 這 3 個時間點進入，也就是 MU 只會從 broadcast disk 1, 3, or 5 開始抓取資料。爲了能達到省電的目的，典型的行動裝置支援兩種操作模式：active mode 與 doze mode。以典型的 wireless PC card 在 doze mode 消耗 60mW 以及在 active mode 消耗 805-1400mW [13]。爲了簡化實驗的複雜度，在不影響實驗結果的前提下，我們忽略有關查詢處理其它會消耗電力的部份，假設 1200mW 包括全部抓取單一 data page 的總耗費電力。總電力的計算公式爲 $P = 1200 \times Time_{active} + 60 \times Time_{doze}$ 。實驗效能的比較，除了考慮總消耗電力的 metric 外，我們還考慮資料抓取的重要性，資料的重要性與資料播出的 frequency 成正比，故我們以抓取相異資料的 total frequency 來代表抓取資料的重要性。

5.2. Performance Results

5.2.1 Total Frequency under different T

這一個實驗，主要是變化 MU 的 turnaround time T，看看我們方法與 BNews-Index 在不同的 T 的設定下，所能抓取資料的重要性。因爲更長的 T 代表 MU 可以抓取更多資料故 total frequency 會隨 T 的增加而增加如圖 4 所示。這符合我們前面的分析結果。

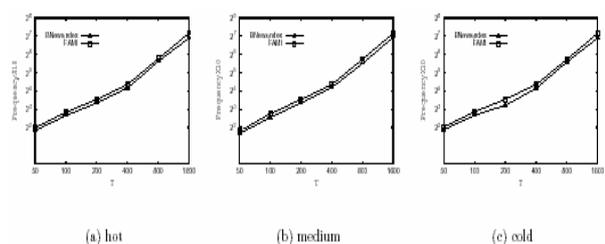


圖 4. The frequency affects under different T.

5.2.2 Power Consumption under different T

實驗結果如圖 5 所示，主要是變化 MU 的 turnaround time T，看看我們方法與 BNews-Index 在不同的 T 的設定下，所耗費的電力。因爲更長的 T 代表 MU 可以抓取更多資料故 power consumption 會隨 T 的增加而增加。在前一個實驗說明我們的方法在相同的 T 下可抓取更多更重要的資料，此一實驗則證實我們的方法不但能抓取更多更重要的資料而且比起 BNews-Index 還要更節省電力。

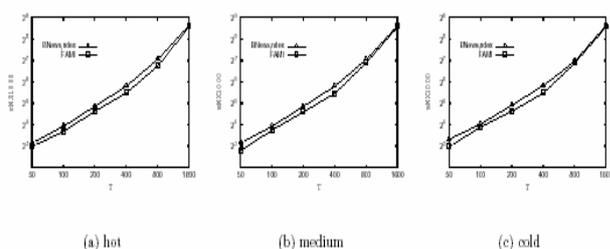


圖 5. The power consumption affects under different T.

6. Conclusions and Future Work

就 Server 端而言，broadcast 是一種很自然用來解決 scalability 與 bandwidth problems 的方法，特別適用於資源相當受限無線通訊環境提供 MU 資訊。而就 client 端 MU，其 power energy is limited 且具 mobility 特性。為了提供 MU 更好的資訊服務。因此一個好的 broadcast strategy 必需同時考量 MU 這兩項因素。在這篇論文中，我們研究如何安排 broadcast program 來讓 broadcast system 可以主動的提供 MU 適當的 access modes。MU 只要 follow 這樣的 access modes 就能以最省 power energy 的方式抓取足夠滿足其查詢所需資料。在此我們提出 AMI 的方法，藉由在 broadcast pages 加上一小小 AMI table，如此不管 MU 在何時 tune into public channel 皆能馬上經由該抓取的 data page 上的 AMI table 查詢滿足其查詢需求的 access modes。經由 performance evaluation 的驗證，我們提出 AMI 方法其各方面效能遠遠優於為了能解決相同問題的改良式 on air index approach。目前我們只處理 single channel broadcast 的問題，未來我們希望將此問題擴大到 multiple broadcast channels 上。

7. Reference

- [1] Swarup Acharya and Rafael Alonso and Michael J. Franklin and Stanley B. Zdonik, Broadcast Disks: Data Management for Asymmetric Communications Environments, in *Proceedings of the ACM SIGMOD Conference*, May 1995, San Jose, CA, USA, pp.199-210.
- [2] Swarup Acharya and Michael J. Franklin and Stanley B. Zdonik, Dissemination-based Data Delivery Using Broadcast Disks, *IEEE Personal Communications*, Vol.6, No.2, December 1996, p.p. 50-60.
- [3] Swarup Acharya, Michael Franklin, and Stanley Zdonik, Balancing Push and Pull for Data Broadcast, in *Proceedings of ACM SIGMOD Conference*, May 1997, Tuscon, Arizona, p.p. 183-194.
- [6] A. Datta, D.E. Vandermeer, A. Celik, and V. Kumar, "Broadcast Protocols to Support Efficient Retrieval from Database by Mobile Users", *ACM Transactions on Database Systems*, Vol. 24, No.1, March 1999, p.p. 1-79.
- [7] Quinglong Hu, Wang-Chien Lee, and Dik Lun Lee, A Hybrid Index Technique for Power Efficient Data Broadcast, *Distributed and Parallel Database*, Vol. 9, No. 2, 2001, p.p. 151-177.
- [8] Tomasz Imielinski, S. Viswanathan, and B. R. Badrinath, Power Efficient Filtering of Data on Air, in *Proceedings of the 4th International Conference on Extending Database Technology (EDBT)*, March 1994, pp. 245-258.
- [9] T. Imielinski, S. Viswanathan, and B. R. Badrinath, Data on Air: Organization and Access, *IEEE Transactions on Knowledge and Data Engineering*, 9(3), 1997, pp.353-372.
- [10] Shou-Chih Lo and Arbee L.P. Chen, Optimal Index and Data Allocation in Multiple Broadcast Channels, in *Proceedings of the 16th International Conference on Data Engineering*, February 2000, San Diego, California, pp. 293-302.
- [11] Sheldon M. Ross, Introduction to Probability and Statistics for Engineers and Scientists, *John Wiley & Son Inc.*, 2000.
- [12] M. A. Viredaz, L. S. Brakmo, and W. R. Hamburg, Energy management on handheld devices, *ACM Queue*, Vol. 1, No. 7, October 2003, p.p. 44-52.
- [13] Peter Triantafillou, R. Harpantidou and M. Paterakis, High Performance Data Broadcasting: a Comprehensive System's Perspective, in *Proceedings of the Second International Conference on Mobile Data Management (MDM01)*, Hong Kong, China, January 8-10, 2001, p.p. 79-90.
- [14] Nitin H. Vaidya and Sohail Hameed, Scheduling Data Broadcast in Asymmetric Communication Environments, *ACM Wireless Networks*, Vol. 5, No. 3, May 1999, p.p. 171-182.
- [15] Jianliang Xu, Wang-Chien Lee, Xueyan Tan, Exponential Index: A Parameterized Distributed Indexing Scheme for Data on Air, in *Proceedings of the Second International Conference on Mobile Systems, Applications, and Services (MobiSys04)*, June 2004, Boston, MA, p.p. 153-164.
- [16] Mark Weiser, Some Computer Science Problems in Ubiquitous Computing, *Communications of the ACM*, July 1993
- [17] <http://www.starband.com> on line
- [18] <http://www.direcway.com> on line
- [19] Mark Weiser, Some Computer Science Problems in Ubiquitous Computing, *Communications of the ACM*, July 1993.