

一個以序列號碼為主的無線通訊計算之暫存資料驗證法 A Sequence Number Based Cache Invalidation Scheme for Mobile Computing Environments

呂永和
Yungho Leu

翁國彬
Kuopin Weng

李之中
Chi-Chung Lee

台灣科技大學資管系
Department of Information Management
National Taiwan University of Science and Technology
yhl@cs.ntust.edu.tw

摘要

在無線通訊計算環境之下，行動主機各保有資料，供日後繼續使用，但一旦伺服器將資料異動，則客戶端的資料便會產生不一致的情況。在本篇論文中，我們提出一種稱為序列號碼法的資料驗證方法，在保持暫存資料一致的情形下，盡量降低所需傳輸的驗證資料量，而達到有效使用網路頻寬的目的。

Abstract

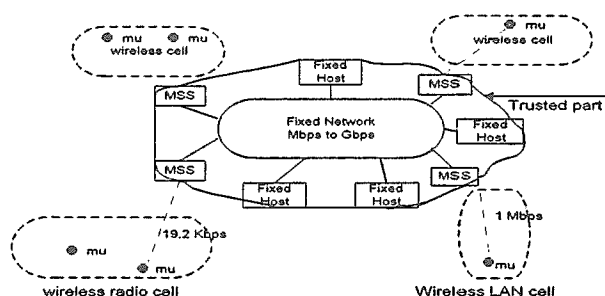
In a mobile computing environment, a mobile host usually caches data which are transmitted from a data server for future applications. While caching reduces data transmission cost, it induces data inconsistency when the data server updates the data source of the corresponding cached data. In this paper, we propose an efficient cache invalidation scheme to maintain cache consistency of the mobile client. Unlike previous schemes which incur a large invalidation report, our scheme effectively reduce the size of the invalidation report using serial numbers. By both analysis and simulation, we show that significant performance gains are attained by our scheme.

關鍵字：暫存資料驗證法 (Cache Invalidation Scheme)、暫存資料的一致性 (Cache Consistency)、無線通訊計算環境 (Mobile Computing Environment)

1 簡介

隨著電腦科技與無線網路 (Wireless Network) 技術的進步，人們可以利用手提式電腦在任何地方、任何時間，利用無線通訊技術擷取存在公司中伺服器 (Server) 內的相關資料，處理公司的日常業務。這種結合電腦與無線通訊技術而成的系統，稱之為無線通訊計算環境 (Mobile Computing Environment)。

This work was supported by the Republic of China National Science Council under Contract NCS 87-2213-E-011-002



MU: 行動主機 (Mobile Unit)

MSS: 支援無線通訊工作站 (Mobile Support Station)

Fixed Host: 固定式主機

圖1、無線通訊計算環境

無線通訊計算環境如圖1所示，主要是由固定式主機 (Fixed Host) 與行動主機 (MU) 所組成 [5]，各個固定式主機間是以有線網路來完成資料通訊或資料傳輸。而固定式主機如果擁有無線通訊傳輸功能者，我們又稱它為支援無線通訊工作站 (MSS)。

每一個行動主機在同一時間，只能隸屬於一個區域。行動主機可以從一個區域移動到另一個區域，這個動作叫做區域轉移 (HANDOFF) [3,4]。為了使行動主機在各個區域都能夠繼續接收資料且能維持資料之一致性，每一個支援無線通訊工作站中所含的資料庫將完全是一樣的 [9]。

無線計算環境依是否能提供雙向溝通，可分為非對稱式通訊環境 (Asymmetric Communication Environments) 與對稱式通訊環境 (Symmetric Communication Environments)。在非對稱式通訊環境之下 [1]，只有伺服器 (Server) 可以傳送訊息給客戶端 (Client)。而在對稱形通訊環境下 [9]，客戶端與伺服器可以作雙向溝通。本篇採用對稱式的架構。

行動計算環境，跟以往架構在有線網路上的環境有多處明顯的不同，而這些不同處主要有列幾點 [1,6]：

- 客戶端的運作必須依賴電池的電力，但電池的電量有限，所以客戶端必須時常維持在睡眠模式下 (離線狀態) 或甚至會將電源關掉，以節省電量。
- 此環境底下其頻寬 (Bandwidth) 非常有限，所以如何有效地利用這有限的頻寬是很重要的。

也就因為頻寬有限，所以我們希望客戶端能將伺服器送來的資料暫存(CACHE)，以供未來的應用繼續使用，從而降低網路上的資料流量。然而當伺服器端修改對應的資料後，其暫存資料就必需予以更新，以保持暫存資料的一致性(Cache Consistency)，達成暫存資料一致性的方法稱為暫存資料驗證法(Cache Invalidation Scheme)

在[9]中，作者對行動計算環境作了詳細的分類。首先，根據伺服器是否知道客戶端所保有的暫存資料，將伺服器分為有狀態伺服器(Stateful Server)及無狀態伺服器(Stateless Server)。有狀態伺服器知道客戶端保有那些資料；而無狀態伺服器則否。再者，根據伺服器傳送異動資料訊息(Invalidation Report，以下將以IR作為簡稱)的方式又將伺服器分為同步式(Synchronous)及非同步式(Asynchronous)兩類。在同步式中，伺服器週期性地傳送出IR。而在非同步式中，一當資料被異動，其IR馬上被送出。

本論文是採用同步式與無狀態伺服器的行動計算環境下提出一個有效率的暫存資料驗證法。而我們的方法稱為序列號碼法(Serial Number scheme 簡稱SN法)。在我們的方法中對每一個異動資料就給定一個序號(流水號)，客戶端利用此序號完成暫存資料驗證工作。由第4節中模擬的結果與分析，我們發現在許多狀況之下，序列號碼法都比其它方法有效率。

本篇論文的架構如下：第2節討論相關的研究。第3節介紹我們的方法。在第4節中我們對我們的方法及相關方法進行效益評估及比較分析。第5節提出本論文的結論及未來研究方向。

2 相關研究

在介紹相關研究之前，必須先加以說明的是，在我們所假定的行動計算環境之下，當使用者對客戶端作資料查詢時，客戶端會將此使用者的查詢放在一個佇列之中；而當收到伺服器IR時，客戶端會去驗證暫存資料，留下正確的暫存資料，然後才將使用者的查詢由佇列中拿出處理，如查詢的資料存在暫存資料之中，則直接將此暫存資料送給使用者，否則客戶端會向伺服器申請資料，並將此資料送給使用者。

我們再定義一下網路傳輸成本，網路傳輸成本會包含下列兩種，一是IR傳輸成本，另一則是資料傳輸成本。IR傳輸成本是伺服器送IR給客戶端時，所需傳輸的資料量。而資料傳輸成本，則是當使用者向客戶端查詢資料時，此資料不在暫存資料之中，而需向伺服器下載資料，所需傳輸的資料量。以下我們介紹幾種常見的暫存資料驗證方法。

2.1 BROADCASTING TIMESTAMPS

我們簡稱此法為TS法[9]。伺服器每隔一段時段(如L秒)，就送一個IR給客戶端，供客戶端去驗證暫存資料是否仍有效。而IR會包含異動資料的資料代

號與其異動時間的時間戳記。且每一個IR會往前包含數個時段內的異動記錄，以避免客戶端遺失少數幾個IR時，就要將所有的暫存資料丟棄。而客戶端也就是依照IR中的資料代號與資料異動時間的時間戳記，驗證客戶端的暫存資料是否仍有效。

此法的精神是以資料最後被異動的時間來驗證客戶端的暫存資料是否有效，它的缺點是所付出的IR成本也相當高，因為用來表示資料異動的時間戳記需要很大的位元數。

2.2 Amnesic Terminals

我們簡稱此法為AT法[9]。伺服器會每隔一段時段(如L秒)，就送一個IR給客戶端，供客戶端去驗證暫存資料是否仍有效。而IR的內容則只有包含上個時段被異動的資料代號。而客戶端收到此IR時，則直接用IR中的資料代號來移除(Invalidate)暫存資料中相對應的資料。

這種方法其網路傳輸IR的成本極低，它不再會有像TS要傳送資料異動的時間戳記成本，而且AT法的IR也僅僅涵蓋了上一個時段所異動資料的訊息，所以其IR傳輸成本非常低。但一旦有遺失IR時，它就得將所有暫存資料丟棄，而使得資料傳輸成本會很高。

2.3 Bit Sequences

我們簡稱此法為BS法[10]。伺服器會每隔一段時段(如L秒)，就送一個IR給客戶端，供客戶端去驗證暫存資料是否仍有效。此法它將資料庫分成很多頁(page)，而每頁都用一個位元相對應，一當頁被異動，則在此頁相對應的位元上會被設為1(代表頁被異動)。而每次IR傳送時，會將所有記錄頁的位元送出，除此之外，還會送出若干的時間戳記，供客戶端比較並判斷，有那些頁其相對應位元為1者，需被從客戶端中移除(詳見[10])。

此法不管伺服器上有無資料異動也不管資料異動的頻率高低，其所送出的IR傳輸量幾乎都是一樣的，所以當伺服器上的異動頻率很低時，此法的IR傳輸成本就相對的大。而且BS法因為沒有對每一個資料都給一個驗證的資訊，所以有可能發生誤判而將正確的暫存料丟棄(False Invalidation)，而這種誤判也將造成資料傳輸成本的增加，使得整個效率變差。

2.4 Grouping with Cold Update-set Retention

我們簡稱此法為GCORE法[8]。此法的主要目的是處理離線太久之客戶端的暫存資料驗證。對TS法、BS法及AT法而言，當客戶端離線過久，或者是過多的IR沒收到時，則不管暫存資料是否仍有效，都必須將暫存資料丟棄，而GCORE的作法則是希望盡量補救這一部份的資料，使得正確的暫存資料得以保留。

2.5 SIGNATURES[2,7,9]

可將檔案或客戶端中的一群相關資料項透過特定的演算法等，求出一個數值(SIGNATURES)，然後與伺服器的數值比較，從而獲知那些暫存資料需被移除。此法的主要目的是濃縮(Compress)驗證資訊以降低IR的傳輸量。

GCORE法和SIGNATURES法所討論的重點與與本論文有所不同，因此在第4節中未將其列入比較。

3 序列號碼法(Serial Number scheme)

以下我們將介紹我們所提出的有效減低IR成本的暫存資料驗證方法，我們稱之為序列號碼法，簡稱SN法。每當有資料異動發生時，伺服器就會產生一個流水序號，並將流水序號、被異動的資料的資料代號與異動時的時間戳記等放到一個專門存放異動資訊的異動佇列中，等到要廣播IR時，便會將此異動佇列拿出處理，然後將必要的驗證資訊廣播出去，而客戶端在接到此IR時，則會先對IR做必要之處理，之後才用相關資訊來驗證暫存資料的內容是否是正確的。

在此我們要特別強調IR的內容，因這也是我們有效降低IR資料流量的精神所在，此IR的內容只會包含所有循序發生資料異動的資料代號，外加一個流水序號，如此一來將減少如TS法時間戳記所需大的位元數，同時也不會像BS法發生資料誤判(Fulse Invalidation)的情形。下面我們介紹序列號碼法的伺服器與客戶端的演算法與實例。

3.1 序列號碼法之伺服器演算法

我們假設SN法每隔一固定時段(假設為L秒)，便會將IR廣播一次，而此IR會涵蓋數個時段(假設為W秒)內的所有異動記錄。底下我們就來說明伺服器的演算法與舉例說明。

異動時的演算法：

- 1.當有資料異動情形發生，伺服器就產生一個流水序號(每次加一)。
- 2.將這個異動資料的資料代號、異動資料的時間戳記與流水序號放入一個節點(Node)，此節點架構如下。

```
struct node{
    INT pageid;(資料代號)
    INT serial_number;(流水序號)
    DOUBLE timestamp;(時間戳記)}
```

- 3.將此節點放入異動佇列之中。

這裡有一個問題要特別提出說明，每一次產生一個異動，系統就會給予此異動動作一個流水序號，但隨著時間的增加，此流水序號可能會變成很大，所以在此我們對流水序號做了一個限制，我們會給流水序號一個上限值，且假設其至少在數個W秒的時間內，流水序號不會有重複的情形發生。而當流水序號到了一個上限值時，下一個產生的流水序號會再從頭開始產生流水序號，我們可以透過取餘數的方法(MOD)來達成。例：我們給流水序號的上

限值為100,000，當流水號編到100,000時，則下一次發生資料異動時，我們所產生的流水號會是0。

在經過固定時段(如L秒)之後，伺服器就會將異動資訊廣播出去，以下就是廣播IR時的演算法：

- 1.檢查異動佇列內的節點看其發生的時間是否在W秒以前，是則將此節點從異動佇列中移除。
- 2.從異動佇列中找出擁有最小時間戳記的節點，將此節點的流水序號取出，此序號我們又稱之為基底序號(Base Number)。
- 3.將異動佇列中所有節點中的資料代號循序取出。
- 4.將基底序號與所有從異動佇列中循序取出的資料代號組成一個IR，廣播出去。

例一、假設伺服器每隔L時間(此L為一小時)做一次IR廣播，而每次的IR會含蓋最近三個L時段(在此W即為三小時)所發生的異動記錄，且給定流水序號的上限值為300。而伺服器發生資料異動的情形如圖2

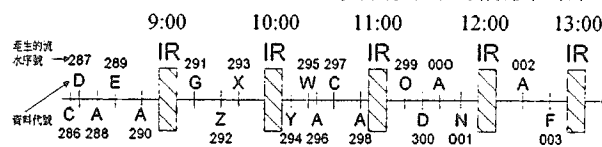


圖2、伺服器端發生資料異動的情形

當資料發生異動時，伺服器端便會將異動資料代號、異動時間戳記與產生的流水序號匯整到一個節點，然後將此節點放入異動佇列之中，而異動佇列內容如表一所示。

當12:00時伺服器準備要廣播IR時，伺服器會去異動佇列(如表一所示)中，找出在9:00以前(12:00減3小時)所發生的異動的節點，並將這些節點移除，移除後的異動佇列內容如表二所示。接著從此異動佇列中找出擁有最小時間戳記的節點(此時間戳記為9:13、資料代號為G、流水序號為291的節點)，將此節點的流水序號取出(這個流水序號我們稱他為基底序號)放到IR中，然後並將所有異動佇列中的節點的資料代號也循序取出放入IR之中，並將此IR廣播出去。此時IR內容如表三所示。

表一、12:00時異動佇列內容如下

資料代號	C	D	A	E	A	G	Z	X	Y	W	A	C	A	O	D	A	N
流水序號	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	000	001
時間戳記	8:05	8:10	8:21	8:35	8:55	9:13	9:31	9:48	10:00	10:10	10:10	10:30	10:50	11:10	11:20	11:30	11:50
									3	3	8	2	6	5	5	5	0

表二、在12:00時經過比較，去除在W時間(三小時)前所異動的訊息，其異動佇列內容如下

資料代號	G	Z	X	Y	W	A	C	A	O	D	A	N
流水序號	291	292	293	294	295	296	297	298	299	300	000	001
時間戳記	9:13	9:31	9:48	10:03	10:13	10:18	10:32	10:56	11:15	11:25	11:35	11:50

表三、12:00時要被廣播的IR內容如下

資料代號	G	Z	X	Y	W	A	C	A	O	D	A	N
------	---	---	---	---	---	---	---	---	---	---	---	---

基底序號=291

3.2 序列號碼法之客戶端演算法

我們在伺服端的演算法中可以看到，伺服端有一個異動佇列專門用來存放異動資訊；相對地在客戶端也有一個暫存佇列，用來記錄客戶端現在擁有那些資料與他們的流水序號之暫存節點。而此暫存節點結構如下：

```
struct cache_node{
    INT cpageid;(資料代號)
    INT cserial_number;(流水序號)
    INT flag;(旗標)}
```

其中旗標是用來判斷此暫存節點是有效資料或是應被移除的資料，在此我假設當旗標值為0時，代表此資料是有效的；當旗標值為1時，代表此資料必須被移除，而旗標的初始值為0。

假設客戶端最後一次收到IR的時間為 T_{ib} ，而目前收到的IR其在伺服端的廣播時間為 T_i ，而流水序號的上限假設為 $MaxN$ ，當使用者向客戶端要求查詢資料時，客戶端會先將此查詢需求放到一個查詢佇列之中，待收到IR時，才去處理此查詢佇列的東西。

客戶端收到IR時的演算法：

- 1.當客戶端收到IR後，便比較 $T_i - T_{ib}$ 是否大於 w (斷線是否過久)，若是，則將所有暫存資料移除並結束；否則，執行2。
- 2.將所有IR中的資料代號取出，並用基底序號來將所有資料代號所對應的流水序號還原。(第一個資料代號的流水序號就等於基底序號，第二個資料代號的流水序號就等於基底序號+1然後對 $MaxN+1$ 取餘數，第 n 個資料代號的流水序號就等於基底序號+ $n-1$ 然後對 $MaxN+1$ 取餘數。
- 3.將由IR所得之資料代號，從小的流水序號開始，將其資料代號與暫存佇列的節點的資料代號比較，如果找到資料代號相同的節點，但其流水序號不同(資料已過時)，便點此節點的旗標設為1，否則設為0。
- 4.將所有旗標為1的節點移除，並將此節點中資料代號所指的暫存資料一併移除
- 5.檢驗暫存佇列中的所有暫存節點，找出資料代號沒包含於此次IR中者，將此類暫存節點的流水序號設為-1
- 6.去執行查詢佇列的查詢動作，當查詢的資料有在暫存佇列中，則將暫存資料直接送給使用者；否則執行7。
- 7.客戶端先向伺服器取得資料後，並檢驗此資料的資料代號是否有在IR中出現過，若有則將此資料代號的最新流水序號(可由IR中取得)與資料代號匯整到暫存節點。否則將此資料代號的流水序號設為-1，然後匯整到暫存節點，之後才將資料送給使用者。

例二、承例一，假設客戶端有12筆暫存資料，其資訊如表四，且假設客戶端在10:00離線、在11:30重新連線，且在11:30到12:00之間使用者向客戶端要求查詢三項資料，如表五所示。而在12:00收到IR，則收到IR時客戶端會有以下的動作：

客戶端在12:00時收到的IR原始內容如表六所示，之後客戶端便利用基底序號291來恢復所有IR中資料代號所相對應的流水序號，結果如表七。

然後客戶端便拿此恢復流水序號後的IR去與暫存佇列中的節點做比較(IR中的資料代號與流水號取出方式，是從流水序號小的開始取)，當發現IR中的資料代號與暫存佇列的資料代號相等的情況下，就去檢驗他們的流水序號是否相等，當流水序號相等則將暫存節點的旗標設為0(表示暫存資料仍然有效，例如資料代號為X、流水序號為293的暫存節點)，流水序號不相等則將旗標設為1(表示暫存資料已過時，例如資料代號為A、流水序號為290的暫存節點)，其比較完成後結果如表八所示。

接下來將所有旗標為1的暫存節點從暫存佇列中移除。並將剩餘的暫存節點中，其資料代號並沒有被此次IR所包含的暫存節點的流水序號改為-1，如表九所示。最後便去執查詢佇列的查詢動作，第一個節點是查詢N，資料代號N的節點不在暫存資料中，而N的資料代號有包含在此次的IR之中，所以客戶端向伺服器取得N資料，並從IR中將資料代號為N的最新流水序號取出，與N資料代號匯整，放入暫存佇列之中。而第二個節點是查詢資料R，其不存在於暫存資料中，且IR中不包含R的資料序號，所以當向伺服器取得資料後，會將資料代號R與一個值為-1的流水代號匯整到暫存節點中。而第三個節點是查詢資料X，其存在於暫存資料中，所以直接將資料送給使用者。處理完查詢佇列後，暫存佇列的內容如表十所示。

表四、收到12:00的IR以前，暫存佇列的資訊如下

資料代號	Q	Z	B	D	N	M	H	I	J	P	A	X
流水序號	-1	292	-1	287	286	282	279	-1	277	273	290	293
旗標	0	0	0	0	0	0	0	0	0	0	0	0

表五、12:00時查詢佇列之內容如下

資料代號	N	R	X
------	---	---	---

表六、收到的IR內容如下

資料代號	G	Z	X	Y	W	A	C	A	O	D	A	N
------	---	---	---	---	---	---	---	---	---	---	---	---

基底序號=291

表七、經過基底序號還原IR的流水序號後，其內容如下

資料代號	G	Z	X	Y	W	A	C	A	O	D	A	N
流水序號	291	292	293	294	295	296	297	298	299	300	000	001

表八、經過IR與暫存佇列比較資料後，將資料代號相等而流水序號不相等的暫存節點之旗標設為1。內容如下

資料代號	Q	Z	B	D	N	M	H	I	J	P	A	X
流水序號	-1	292	-1	287	286	282	279	-1	277	273	290	293
旗標	0	0	0	1	1	0	0	0	0	0	1	0

表九、將無效的資料移除，並將流水序號修正後，暫存佇列內容如下

資料代號	Q	Z	B	M	H	I	J	P	X
流水序號	-1	292	-1	-1	-1	-1	-1	-1	293
旗標	0	0	0	0	0	0	0	0	0

表十、處理完查詢佇列後，暫存佇列的內容如下

資料代號	Q	Z	B	M	H	I	J	P	X	N	R
流水序號	-1	292	-1	-1	-1	-1	-1	-1	293	001	-1
旗標	0	0	0	0	0	0	0	0	0	0	0

4 效益分析

在這第4節中我們會分別針對BS法、TS法、AT法、與我們的SN法做分析比較，第一部份我們將針對網路傳輸成本的IR傳輸成本做式子與例子分析，第二與第三部份則是設定系統模擬的環境，以系統模擬所得出的結果對所提的方法之傳輸成本做分析比較。

4.1 網路的IR傳輸成本分析

網路傳輸成本包含下列兩種，一是IR傳輸成本，另一則是資料傳輸成本。IR傳輸成本是伺服器送IR給客戶端時，所需傳輸的資訊量。而資料傳輸成本，則是當使用者向客戶端查詢資料時，此資料不在暫存資料之中，而需向伺服器下載資料，所需傳輸的資料量。

以下會各別針對TS法、AT法、BS法與我們所提的SN法，在IR傳輸成本上作分析比較。假設一個時段為L秒，而W則為包含k個時段的時間， n_w 代表平均在每個w秒(k個時段)內有幾個不同的資料被異動、N代表此資料庫中總共有的資料數量、 $\log(N)$ 為資料代號所需的位元數、T為表示一個時間戳記所需的位元數、 n_L 代表在L秒內(一個時段)有幾個不同的資料被異動。

a. TS法其IR的傳輸成本為經由[10]中得知為 $n_w(\log(N)+T)$

b. AT法其IR的傳輸成本經由[10]中得知為 $n_L(\log(N))$

c. BS法其IR的傳輸成本經由[10]中得知為 $2N + T\log(N)$

d. SN法的IR傳輸成本則為， $n_{w,sn}(\log(N))+\log(M)$ ，其中 $n_{w,sn}$ 表示在w秒內一共發生了多少次的資料異動的動作。M是表示流水序號的上限值。而 $\log(M)$ 流水序號所需的位元數。

4.2 系統模擬環境設定

為了能有效釐清每個方法的優劣，所以我們勢必要做一個系統模擬來分析比較。我們的系統模型如下：

參考[8]中的系統模型，我們假設所有的寫(write)的動作都是在伺服器端完成的，而客戶端資料只能進行資料讀取動作，客戶端只有在連線狀態時，才可以接受使用者的資料查詢申請，而當客戶端收到資料查詢申請時，並不會馬上給使用者資料，而是會將此資料查詢申請先放到一個查詢佇列中。然後等待下一次收到IR時，會先去驗證暫存資料是否有效，然後再比較使用者所需的資料是否有存在於暫

存資料中；若有，則將此資料訊息送給使用者，若沒有則向伺服器端取得資料後，再送給使用者。

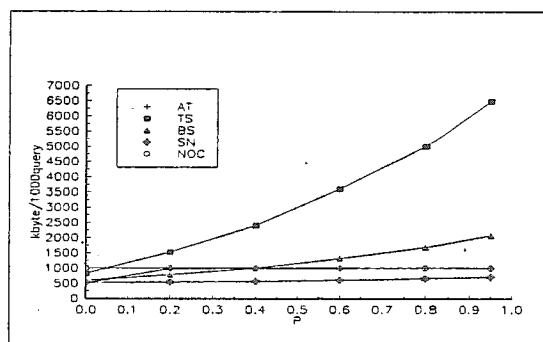
假設在收到IR之後，下個時段客戶端要成離線狀態機率為P，每次離線的時間長度為平均數是 L_{disc} 、標準差為 $L_{disc}/2$ 的常態分配(Normal distribution)。

假設此資料庫共有N個資料、而客戶端最多可擁有B個暫存資料、假設每個資料內容有1k個位元組、資料代號需 $\log(N)$ 位元來表示、T表示一個時間戳記所需的位元數。並假設任兩個異動動作的間隔時間，是一個平均數為u的指數分配(Exponential distribution)。且假設每一個資料被異動的機率都相同。而任兩個查詢動作的間隔時間，是一個平均數為q的指數分配。N為伺服器端資料庫的資料數量、B為客戶端最多可以擁有的暫存資料個數(暫存資料區的大小)。M是表示流水序號的上限值。而 $\log(M)$ 為流水序號所需的位元數。

表十一、系統模擬環境參數如下

變數名稱	定義	給值
N	伺服器端資料庫的資料總數	10,000 個資料
B	客戶端最多可擁有多少筆暫存資料	5,000 個資料
u	任兩個異動動作的平均間隔時間，呈指數分配	10 秒/異動資料
q	任兩個查詢動作的平均間隔時間，呈指數分配	0.5 秒/查詢動作
P	一個條件機率，當客戶端是連線狀態之下，下一個時段會開始離線的機率	0.2
L_{disc}	每一次離線平均會離線多久，呈常態分配，其標準差為 $L_{disc}/2$	200 秒
L	廣播IR的間隔時間	20 秒
W	一個IR涵蓋的時間長度	2,000 秒(100個L時段)
page	一個資料的內容有多少位元組	1 K位元組(1k byte)
$\log(N)$	為資料代號的位元數	14 位元
T	表示一個時間戳記所需的位元數	512 位元
M	流水序號的上限	100,000
$\log(M)$	流水序號所需用的位元數	17 位元

4.3 模擬結果分析

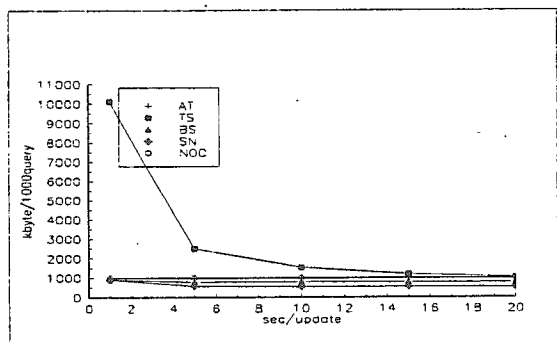


(N=10,000, BN=5,000, w=2,000(100個L), L=20, u=10 秒/異動資料 q=0.5 秒/查詢資料, $L_{disc}=200, M=100,000$)

圖3、客戶端斷線機率大小對網路傳輸成本的影響

在圖3中我們模擬P值(客戶端斷線機率)對網路傳輸成本的影響,我們取平均客戶端完成1000次的查詢動作時,其相對應的網路傳輸成本為多少k位元組為比較的標準。而網路傳輸成本包含1)IR的傳輸成本、2)當查詢時在客戶端的暫存資料中找不到所需資料,而向伺服器端要求資料所需的資料傳輸成本。

由此模擬結果我們可以看到當客戶端一直在維持在連線狀態時,AT法擁有最小的網路傳輸成本,而SN法與BS法的網路傳輸成本也不錯。且可以看到不管用那一種暫存資料的方法,其網路傳輸成本都會比沒有使用暫存資料(圖中以NOC代表沒有使用暫存資料的結果曲線)的情形好。但當斷線機率逐漸增加時,網路傳輸成本的IR部份將逐漸成為網路傳輸的主要部份,最後使得網路傳輸成本爆增,甚至比不使用暫存資料的情形還糟。我們可以從圖二中看到TS法與BS法當斷線機率高時,其網路傳輸成本會超過不使用暫存資料的方法高。而SN法因其能保持暫存資料的正確性,且其IR成本又低,所以SN在一般情況之下,網路傳輸成本都比其他方法好。



(N=10,000, BN=5,000, w=2,000(100個L), L=20, P=0.2 q=0.5 秒/查詢動作, $L_{disc}=200, M=100,000$)

圖4、伺服器端異動資料的頻率對網路傳輸成本的影響

在圖4中我們驗模擬伺服器端異動資料的頻率對網路傳輸成本影響。我們是取平均客戶端完成1000次的查詢動作,其相對應的網路傳輸成本來比較。

由此模擬結果我們可以看到,當伺服器端異動資料的頻率越高時(異動動作的間隔時間越短時),TS法效能越差(因IR成本大),而SN法與BS法則有相對較佳的網路傳輸成本,而AT法因為受限於斷線機率等於0.2的影響,使得客戶端的暫存資料時常因沒收到IR,而導致暫存資料被移除,造成資料傳輸成本大增,以致於網路傳輸成本接近於不使用暫存資料的情形。

綜合以上,我們可以發現序列號碼法(Serial Number Scheme),在許多的狀況上都得到相對較佳的網路傳輸成本。而AT法在一直維持連線的情況,其網路傳輸成本最佳。而TS法則在大部份的網路傳輸成本上的表現都不理想。至於BS法在異動頻率越高的情形下(異動資料的間隔時間越短),相對於其他方法,會有很不錯的表現。

5 結論

在無線通訊網路之下,我們必須盡量節省電池電力與有效的利用有限的頻寬。在本篇論文中,我們提出一個使用序列號碼來驗證暫存資料的有效性的方法。經由詳細的系統模擬分析發現,我們所提的方法,在大部分的情形下,都比其它的方法好。

而在未來研究的方向上,我們將去把序列號碼法與GCORE法做結合,而使得當斷線時間過長時(超過一個IR所能包含的時間長度時),也能儘可能地減少正確的暫存資料被丟棄的情形。而另一方面我們也將致力於無線通訊網路上資料並行控制的研究,以期達到不但能使客戶端擁有正確的資料,而且還能使客戶端能擁有異動資料的功能。

參考文獻

- [1] T. Imielinski, S. Viswanathan and B. R. Badrinath. "Data on Air: Organization and Access." IEEE Transactions on Knowledge and Data Engineering, VOL.9, NO. 3, MAY/JUNE 1997.
- [2] J. J. Metzner. "A Parity Structure for Large Remotely Located Replicated Data Files." IEEE Transactions on Computers, Vol.C-32, No.8, August 1983.
- [3] Y. B. Lin. "Cellular Inter-System Handoff(Tutorial)." PCS-NCTU-96-02.
- [4] A. R. Noerpel and Y. B. Lin. "Handover Management for a PCS Network." To appear in IEEE Personal Communications Magazine PCS-NCTU-96-12.
- [5] J. Ioannidis and G. Q. Maguire. "The design and implementation of a mobile internetworking architecture." In USENIX Winter 1993 technical conference January 1993.
- [6] T. Imielinski. and B. R. Badrinath. "Mobile Wireless Computing: Challenges in Data Management." In Communications of ACM, October 1994.
- [7] D. Barbara, B. Feijoo, and H. Garcia-Molina. "Exploiting Symmetries for Low-cost Comparison of File Copies." In Proc. Int. Conf. Distributed Comput. Syst., San Jose, Jun 1988.
- [8] K. L. Wu, P. S. Yu and M. S. Chen. "Energy-Efficient Caching for Wireless Mobile Computing." In Data Engineering 1996.
- [9] D. Barbara and T. Imielinski. "Sleepers and Workaholics: Caching Strategies in Mobile Environments." IN Proceedings of the ACM SIGMOD Conference on Management of Data, pages 1-12, 1994.
- [10] J. Jing, O. Bukhres, A. Elmagarmid, and R. Alonso. "Bit-Sequences: A New Cache Invalidation Method in Mobile Environments." Technical Report CSD-TR-94-074, Department of Computer Sciences, Purdue University, May 1995.