無線隨意網路中考量連結穩定度的協同資料快取方法

呂永和
國立臺灣科技大學資訊管理研究所
台北市基隆路四段四十三號
E-mail: yhl@cs.ntust.edu.tw

劉威呈
國立臺灣科技大學資訊管理研究所
台北市基隆路四段四十三號
E-mail: M9309208@mail.ntust.edu.tw

摘要

在無線隨意網路的環境下，因為網路拓樸經常變動 ，節點間的連線時常會中斷，導致網路斷裂的情形發生。為了要解決網路斷裂的問題，可以使用資料備分。資料備份的策略考量每個節點該備份哪些資料項，以提升整體的資料存取的效率。現有資料備份策略中，各節點只考慮快取對自己最有益處資料；或是有考慮周圍節點的資料需求，但並未考慮節點間連線的穩定度。

在本論文中，我們提出一個考慮節點連線穩定度的資料備份策略，我們考量節點間過去一段時間的連接穩定度，決定該備份哪些資料項。若兩節點過去的連接情況越穩定，則兩節點會備份對彼此都有益的資料項。經過實驗結果顯示，我們的資料備份方法對於資料的可存取性和請求延遲，皆有不錯的表現。

關鍵詞：無線隨意網路、資料快取、資料備份、穩定度。

Abstract

In a Mobile Ad Hoc Network (MANET), a mobile client moves freely in the network. The mobility of a mobile client causes the rapidly change on the network topology, unstable connection between mobile nodes and frequent network division. To solve these problems, data caching is proposed. In a data caching method, each node considers which data items should be cached in its local storage to improve the system performance. In the existing data caching approaches, a mobile client usually stores data items that favors the client itself and, sometimes, its neighbors. However, very few researches take into account of the link stability between mobile clients. In this paper, we propose a data caching scheme that takes into account of link stability. We measure link stability between two mobile clients by their link conditions in a recent period of time. If two nodes connect frequently in this period of time, the link is deemed to be more stable and the two mobile clients tend to be more cooperative in caching data items. Through an experiment using NS2, we showed that our scheme offers better performance in terms of data accessibility and query delay.

Keywords: Mobile Ad Hoc Network, Cooperative Data Caching, Link Stability

1 緒論

無線隨意網路(Mobile Ad Hoc Networks，簡稱MANET)是一個沒有基礎建設、由許多行動裝置所組成的無線網路環境，在這個環境下的行動裝置像是 PDA、Notebook 等可以利用無線電波來跟周圍其它的行動裝置來溝通，能夠跟鄰近的行動裝置來拿到自己所需的資料，甚至能夠透過其他裝置來跟

更遠的裝置來拿資料。

無線隨意網路可以由行動裝置臨時組成，又不需要網路基礎建設，這些行動裝置可以快速的形成一個無線網路環境，每個行動裝置可以任意地移動，並且能夠跟溝通範圍內的其他裝置利用無線電波來作通訊，所以多可以應用在特殊目的的任務行動上，例如軍事行動、災難救援等。

例如在一個執行特定軍事行動的一個小隊中，隊長跟所有士兵皆裝備有著無線通訊能力的行動裝置，士兵可以隨時將周圍最新的情報資料藉由其他士兵的行動裝置傳回給隊長，而隊長可以利用傳回來的資料情報來做出決策，並且指揮命令某個士兵應該執行某個新的任務。

在無線隨意網路的環境下，由於節點不規則的移動，使得原本在彼此通訊範圍內的節點隨時都可能會離開彼此的通訊範圍，而造成節點間因為失去連線無法取得彼此的資料項，會導致資料可存取性的下降，影響到資料快取的效益。這種情況我們稱之為網路斷裂(Network Division)。

為了要解決網路斷裂所產生的問題，我們會採取資料備份(Data Replication)的策略。在[7]的論文中有提到，在一個無線隨意網路的環境下會有許多的行動裝置，像是 PDA，行動電腦等，這些行動裝置的快取空間其實都是有限的，因此無法備份整個網路裡所有的資料，所以每個裝置要選擇對自己或是周圍的人最有效益的資料來備份。

在[1]的論文中，T.Hara 提出了三個資料備份的策略：SAF、DAFN 和 DCG。在作者假設的環境下，每個節點都可以備份 c 個資料項，而且每個節點經過一段重置時間，都會進行一次快取空間的重置。資料內容重置的考量因素是節點對資料存取頻率以及資料重置時的網路拓樸狀態，也就是會考慮到多個節點之間的快取合作。

在[3]的論文中 Cao 提出了一個觀點，在資料備份的策略下，因為每個節點的快取空間有限，沒辦法把系統環境中所有的資料快取到自己的空間內，必定只能選擇對自己或是周圍的節點最有效益的資料來備份。當節點大部分的快取空間都為周圍的人備份資料時，以自己為考量所備份的資料也就會相對的減少，這樣整體網路的資料可存取性會較大，不過節點的請求延遲也會相對的較高；相反地，當節點都以自己為考量來備份資料，以周圍節點為考量所備份的資料就會減少，這樣節點的請求延遲會較低，不過整體網路的資料可存取性也會相對地較低。因此，作者想要找出一個資料快取方法，在節點的資料可存取性和請求延遲之間取得平衡。

目前在無線隨意網路下有關資料備份的研究[1][2][3][4][5][6]中，對於節點間穩定性的議題探討不多，當節點間的連接狀況不穩定時，採取的資料備份策略就會因而失去效益，不僅無法使整體網路的資料可存取性提高，同時也浪費了節點的快取空間。因此，在資料備份的策略中，節點間的穩定度是一個很重要的考量因素。接下來我們要介紹我們的快取空間分配策略。

2 考量穩定度的快取空間分享策略

2.1 隨意網路環境假設

以下是在我們系統中所會使用到的符號，我們將一一說明之。

1. N_i：節點 i 的標示方法。我們假設在系統中共有 m 個節點，每個節點皆有屬於自己的標示方法，而且絕對不會跟其他節點重複，節點編號從數字 1 到數字 m，例如節點 5 的標示方法為 N_5，而節點 20 的標示方法為 N_{20}，而所有的節點的集合為$\{N_1,N_2,\ldots,N_m\}$。
2. D_j：資料項 j 的標示方法。在我們的假設下，共有 n 個資料項存在於系統中，每個資料項也都有自己的標示方法，而且絕對不會跟其他資料項重複，資料項的編號是由數字 1 到數字 n，例如資料項 1 的標示方法為 D_1，而資料項 10 的標示方法為 D_{10}，而所有的資料項的集合為$\{D_1,D_2,\ldots,D_n\}$。
3. C：假設每個節點的快取空間大小為 C，意即每個節點至多可以快取 C 個資料項，而且每個節點所快取的資料項以不重複為原則。C 的大小必須小於 n，代表每個節點不可能將

系統內所有的資料項快取到自己的快取空間中。

4. a_{ij}：表示節點 N_i 對資料項 D_j 的存取頻率，在我們的系統中，我們假設每個節點對於資料項的存取頻率是已知，且不會變動。

2.2 問題描述

我們想出一個以穩定度為考量的資料快取策略，衡量節點間過去一段時間的連接狀況，來量化我們所需要的穩定度。當節點間過去一段時間的穩定度越大時，對於彼此資料快取的合作影響會越大；相反的，如果節點間過去一段時間的穩定度較小時，對於彼此的資料快取的策略影響也就沒那麼顯著。

2.3 穩定度的探討

2.3.1 穩定度的定義

我們將穩定度定義為節點間在過去一段時間連接情況的穩定程度，所以我們認為良好的穩定度的情況應該為：

1. 過去一段時間內，兩節點保持連接時間越長，表示穩定度越大。
2. 兩節點過去連接時間越靠近現在，未來短時間內也可能仍舊保持連接，表示兩節點穩定度越高。

2.3.2 穩定度的衡量

在系統環境下，每個節點每隔一段不定長度的時間會進行一次資料備份的重置。當節點進行資料備份重置時，會考慮與附近節點的穩定度，來決定合作考量的對象與快取空間備份內容。為了衡量穩定度，每個節點會一直定時的發送 Hello 封包，以利其他節點算出連接時間，進而求出穩定度。當節點對資料的分享快取空間命中率小於一定的比率時，節點就會進行資料備份重置的動作。我們這邊所指的分享快取空間命中率為節點在本身及周圍節點的快取空間中可以拿到資料的機率，因為我們資料備份重置的策略是考慮節點本身和周圍的節點，所以考慮分享快取空間命中率也是以節點本身及周圍節點一起納入考量因素。

為了要算出節點的穩定度，我們要算出節點間的連接時間。每個節點會在資料重置間隔內定時的廣播 Hello 封包，讓周圍的節點能夠利用收到 Hello 封包的資訊來計算連接時間，Hello 封包的內容資訊包含了封包發送者 ID、封包發送時間及封包序號，而當節點 N_i 收到節點 N_j 的 Hello 封包時，可利用收到的 Hello 封包內容資訊來進行以下的動作：

1. 根據封包發送者 ID 判斷是哪個節點送來的封包。
2. 記錄節點的最後封包序號。每個節點發送 Hello 封包的序號順序都是有著連續的編號，所以假設當節點 N_i 收到節點 N_j 的 Hello 封包時，會先記錄下來該封包的序號，當成是節點 N_i 收到節點 N_j 最後的封包序號。
3. 接下來節點 N_i 會判斷之前是否有收過節點 N_j 的 Hello 封包。
   - 是，判斷這次收到節點 N_j 的 Hello 封包序號和上次收到的 Hello 封包序號是否連續，如果前後兩封包的序號是連續的，代表兩節點沒有離開過彼此的溝通範圍過，如果序號是不連續的，代表兩節點有離開彼此的溝通範圍過。
   - 否，就不做任何判斷動作。
4. 如果節點 N_i 之前收過節點 N_j 的封包，根據封包序號是否連續來決定是否要累加兩節點的連接時間。
   - 如果封包序號連續，代表兩節點在兩個封包的時間內一直都在彼此的溝通範圍內，所以要繼續累加兩節點的連接時間。
   - 如果封包序號不連續，代表兩節點離開過彼此的溝通範圍，兩節點的連接有中斷過，所以兩節點要重新開始計算一段新的連接時間。

接下來要討論的是連接時間的時間權重問題。我們要給予每段連接時間一個時間權重值，來代表該連接時間對於過去一段時間的重要性。一段連接時間的結束時間點越靠近現在的時間點，它的時間權重會越大；相反的，連接時間的結束時間點離現在的時間點越遠，它的時間權重會越小。

以圖 1 為例，我們來說明一段連接時間的時間權重算法。假設連接時間 ct 是某兩節點過去一段時間的連接時間，連接時間 ct 的時間結束點為時間點 t_2；而過去一段時間的時間計算起始點為時

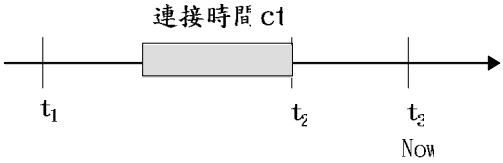

圖 1　時間權重算法解說圖

間點 t_1，計算的結束點為現在的時間點，也就是時間點 t_3，則這段連接時間 ct 的時間權重 tw_{ct} 的算法為：

$$tw_{ct} = \frac{\text{連接時間計算結束點}t_2 - \text{計算時間起始點}t_1}{\text{過去一段時間總長度}t_3 - t_1}$$

最後一個部分我們要解釋如何利用連接時間結合時間權重來算出節點間的穩定度。當我們要計算兩節點的穩定度，我們要先算出過去一段時間內兩節點的每段連接時間的穩定度權重，再將所有連接時間的穩定度權重加總起來，可得到一個穩定性權重的總和，代表了在過去一段時間兩節點的穩定度。

我們用圖 2 的例子來說明如何計算穩定度。圖中說明了在過去一段時間內，節點 N_i 和節點 N_j 的連接情況。假設過去一段時間的計算起始點為 t_1，計算的結束點為 t_5，而在這段時間內，N_i 和 N_j 共有三段的連接時間，分別是連接時間 ct_1、連接時間 ct_2 和連接時間 ct_3，這三段時間的長度分別為 l_1、l_2、l_3，而這三段連接時間的結束點分別為時間點 t_2、t_3 和 t_4。

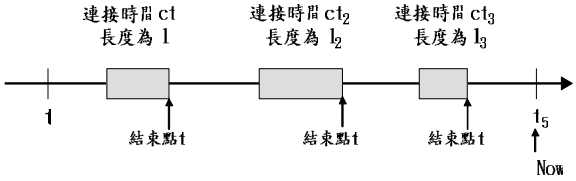

圖 2　穩定度計算解說圖

我們對於某一段的連接時間 ct 的穩定性權重 sw 的算法為：

$$sw_{ct} = \frac{\text{ct 的時間長度}}{\text{總計算時間長度}} * \text{該 ct 的時間權重}$$

則節點 N_i 和節點 N_j 在過去一段時間內的總穩定度權重 sw_{ij} 的算法為

$$sw_{ij} = \sum (\text{每一段連接時間的穩定度權重})$$

$$= \left(\frac{l_1}{t_5 - t_1} * \frac{t_2 - t_1}{t_5 - t_1}\right) + \left(\frac{l_2}{t_5 - t_1} * \frac{t_3 - t_1}{t_5 - t_1}\right) + \left(\frac{l_3}{t_5 - t_1} * \frac{t_4 - t_1}{t_5 - t_1}\right)$$

算出來的 sw_{ij} 即為節點 N_i 和節點 N_j 在過去一段時間的穩定度。

2.4　快取空間分享與穩定度

接下來我們將討論如何把之前計算出來節點間的穩定度運用在我們的快取空間策略中。在討論快取空間的分配運用之前，我們希望能夠找出對於某一節點相對穩定的節點，來當作該節點的快取空間分配考量對象，也就是穩定節點的概念。因為當我們的快取空間分配的考量對象都是較穩定的節點時，節點所備份的資料比較容易會被周圍的較穩定的節點來存取，所以資料的可存取性會比較高，才可以使我們的分配策略發揮較好的效益。

所謂的穩定節點，是指對於某一節點來說，過去一段時間內與該節點穩定度較高的節點，通常我們會設一個穩定度的門檻值 $sw_{threshold}$，來當作符合穩定節點的條件。通常 $sw_{threshold}$ 為一系統變數，系統使用者可依照自己的需求來調整該數值。接下來所討論的快取空間分配策略都是以穩定節點來做考量。

2.4.1　快取空間的大小配置

如之前章節所說的，節點間的穩定度越大時，在快取分配策略中的影響權重也應該越大。所以我們的快取空間分配策略中是以周圍所有的穩定節點為考量，來決定每個節點應該如何分配快取空間給周圍的穩定節點。

我們假設節點 Ni 本身的穩定度為 1(穩定度的範圍介於 0 到 1)，因為節點本身一定可以拿到自己快取空間內的資料。以節點 Ni 本身為考量的快取空間 Cs(i)大小的算法為：

$$C_s(i) = s * \frac{1}{1 + \sum_{Nj \in sn(i)} sw_{ij}}$$

其中 s 代表節點 N_i 的總快取空間大小，sn(i)代表節點 Ni 所有穩定節點的集合，$\sum_{Nj \in sn(i)} sw_{ij}$ 代表 Ni 附近所有穩定節點穩定度的總和，而節點 Ni 分享給周圍其他節點的快取空間 $C_c(i)$的大小為 N_i 的總快取空間 s 減去 Cs(i)的大小。

從我們決定快取空間大小的公式來看，當節點附近穩定節點的穩定度的總和越大時，附近穩定節點所能分到的快取空間相對的就較大；相反的，如果附近穩定節點的穩定度總和不大，節點就要多保留快取空間來供自己使用。

2.4.2 快取空間的內容配置

在我們決定了節點的快取空間大小配置後，接下來要討論如何決定快取空間的資料內容。快取空間的內容配置共分為兩個部份：a.節點 Ni 本身所使用的快取空間 Cs(i)，以及 b.分享給附近周圍穩定節點的快取空間 Cc(i)。

a. 節點本身所使用的快取空間的內容配置 Cs(i)

因為是節點本身所要使用的快取空間，快取空間的內容配置要以節點本身為考量，所以我們依照節點本身對資料項存取頻率由大到小來配置資料，直至節點本身可用的快取空間配置已滿。

接下來用例子來說明如何配置節點本身所使用的快取空間。表 1 為假設在過去一段時間內，節點 N1 與其他節點的穩定度。

表 1　節點 N_1 與穩定節點的穩定度表

在過去一段時間與節點 N 有連接的節點	與節點 N 的穩定度
N_2	0.2
N_3	0.4
N_4	0.6
N_5	0.8

假設節點 N_1 總共有 4 個快取空間，穩定節點的門檻值 $sw_{threshold}$ 為 0.3，所以穩定節點的集合 sn(1)={N_2,N_3,N_4}。假設節點 N_1 的本身穩定度為 1，而以 N_1 為考量的快取空間 Cs(1)大小的計算方法為

$$Cs(1) = \left\lfloor 4 * \frac{1}{1 + (0.4 + 0.6 + 0.8)} \right\rfloor = 1$$

所以以節點 N_1 為考量的快取空間為 1 個快取空間，可以存放一個資料項。接下來我們參考表格 2 中，節點 N_1 對所有資料項的頻率來決定要備份哪些資料到快取空間。

表 2　節點 N_1 的資料存取頻率表

資料項	N 對資料項的存取頻率
D_1	0.41
D_2	0.21
D_3	0.09
D_4	0.13
D_5	0.16

所以我們將表格 2 的資料項存取頻率由大到小排列，並挑選一個最大的值來當作 Cs(1)的內容，排列結果依序為 D_1、D_2、D_5、D_4、D_3，最後 Cs(1)會備份的資料內容為{D_1}。

b.分享給附近周圍穩定節點的快取空間 Cc(i)

Cc(i)分享的對象主要是周圍穩定的節點，所以關於內容的配置也是以周圍穩定的節點對於資料的需求來當作主要的考量因素，同時我們也要考

慮節點本身跟周圍節點的穩定度。穩定度越高的節點，對於資料內容的配置影響越大。

接下來我們用例子來說明如何節點 N_1 配置分享快取空間 Cc(1)內容的三個步驟。

步驟 1.　計算節點 N_1 的所有資料項的資料權重 dw

資料權重的算法為利用節點 N_1 周圍的穩定節點的穩定度和 N_1 周圍的穩定節點對資料的存取頻率。在這個例子中我們同樣利用滿足穩定節點條件的門檻值 $sw_{threshold}$ 來篩選出穩定節點 N_3、N_4 和 N_5。

計算 N_1 的某資料項 D_i 的算法為計算 N_1 的穩定節點集合 sn(1)中每個節點對資料項 D_i 存取頻率乘以穩定度的平均值。我們利用表格 1 中 N_1 穩定節點的穩定度和表格 3 中 N_1 的每個穩定節點 N_j 對資料項 D_i 的存取頻率 f_{ji} 來計算出 N_1 每個資料項的資料權重 dw。

表 3　節點 N_1 穩定節點的資料存取頻率表

資料項	N 的穩定節點		
	N_3	N_4	N_5
D	0.05	0.17	0.23
D_2	0.29	0.36	0.39
D_3	0.11	0.04	0.16
D_4	0.42	0.25	0.14
D_5	0.13	0.18	0.08

我們以計算 N_1 的資料項 D_1 的資料權重 dw_1 為例，dw_1 的計算方法為

$$
\begin{aligned}
dw_1 &= \sum(\text{穩定節點的穩定度} * \text{該節點對 D 的存取頻率}) \\
&= \frac{\sum_{j=2}^{4}(sw_{1j} * f_{j1})}{3} \\
&= (sw_{13} * f_{31} + sw_{14} * f_{41} + sw_{15} * f_{51})/3 \\
&= (0.4*0.05 + 0.6*0.17 + 0.8*0.23)/3 \\
&= 0.102
\end{aligned}
$$

我們依照同樣的方法可以算出資料項 D_2、D_3、D_4 和 D_5 的資料權重 dw_2、dw_3、dw_4 和 dw_5。表 4 為計算出來後的資料權重。

表 4　節點 N_1 的資料權重表

N 的資料項	資料權重
D	0.102　dw
D_2	0.214　dw_2
D_3	0.065　dw_3
D_4	0.143　dw_4
D_5	0.075　dw_5

步驟 2.　配置 N_1 的快取空間 Cc(1)

我們從上面算出節點本身所使用的快取空間 Cs(1)大小為 1，而節點 N_1 的快取空間大小 s 為 4，所以分享的快取空間 Cc(1)大小為總快取空間大小 s 減去 Cs(1)的大小，也就是 Cc(1)可以存放 3 個資料項。

然後我們將表 4 中所有資料項的資料權重由大到小排序，並挑選 3 個資料權重最大的 3 個資料項來存放到 Cc(1)的快取空間中。資料項的資料權重排序結果為 D_2、D_4、D_1、D_5、D_3，所以備份到 Cc(1)的資料內容為$\{D_2,D_4,D_1\}$。

步驟 3.　將 Cs(i)和 Cc(i)重複的資料項移除

為了增加資料的可存取性，節點本身使用 Cs(1)的資料內容跟分享的空間 Cc(1)的資料內容不得重複。我們觀察到節點 N_1 的快取空間中，Cs(1)的內容為$\{D_1\}$，而 Cc(1)的內容為$\{D_2,D_4,D_1\}$，所以有一個重複的資料項 D_1 被兩個快取空間同時備份。這時我們要找一個資料權重次低的資料項來替換 D_1，資料項的資料權重排序結果為 D_2、D_4、D_1、D_5、D_3，因此我們會選擇 D_5 來替換掉 D_1。

最後節點 N_1 本身使用 Cs(1)的資料內容為$\{D_1\}$，而分享的空間 Cc(1)的資料內容為$\{D_2,D_4,D_5\}$。

3 實驗結果與分析

3.1 實驗參數

在本章中我們將用 NS2[8]模擬我們的資料快取策略，並分析我們的實驗結果。NS2 是一個離散事件引發的網路模擬器，NS2 可以提供一個整合式的環境，用來模擬傳輸層的通訊協定(例如 TCP)、網路層的路由協定(routing protocol)、節點間移動行為、傳送資料的行為等在無線網路環境下所需要的架構元件。以下是實驗參數設定。

表 5 模擬實驗的參數設定

參數	值
節點數量(個)	50
資料數量(個)	100
傳輸距離(公尺)	250
快取空間大小(%)	20
資料傳輸速度(M/秒)	2
Zipf-like theta(θ)	0.2~1.0
網路範圍大小(公尺*公尺)	1200*800
節點移動速度(m/s)	0~4
節點暫停時間(sec)	3
平均請求時間(sec)	1
路由協定	DSDV

每個節點對每個資料的需求服從 Zipf-like[14]分配，每個節點存取資料 k 的機率如下：

$$p(x=k)=\begin{cases}\dfrac{(\frac{1}{k})^{\theta}}{\sum_{i=1}^{n}(\frac{1}{i})^{\theta}} & ,k=1..n,\\ 0 & ,\text{otherwise.}\end{cases}$$

我們可以藉由調整 Zipf-like 的分配中的 θ 值來控制實驗中資料存取頻率的偏斜程度。

3.2 實驗分析結果

在接下來的實驗中，我們將把我們的策略跟 Cao 的 RN、Hara 的 DAFN 分別以實驗模擬之，並把實驗結果來做比較。我們的策略取名為 Stability。每個實驗的模擬時間為 7200 秒，而前 1000 秒為系統的暖機時間；在系統的暖機時間時，節點不進行任何的資料快取重置作業。我們的策略是根據 2.3.2 小節中的分享快取空間命中率來決定節點快取空間的重置時機，而 RN 和 DAFN 的節點快取空間的重置時間則是設定為 50 秒，而節點每隔 50 秒會跟周圍的節點來溝通一次。接下來我們將介紹三個實驗。

實驗一：節點本身快取空間命中率(Hit rate)的比較

我們這邊所說的快取空間命中為節點在自己的快取空間中取得資料，而我們的快取空間命中率的定義為

$$\text{Hit rate}=\frac{\text{從本身快取空間取得資料的次數}}{\text{所有發出請求資料的次數}}$$

這裡所說的快取空間命中率與我們在 2.3.2 小節中談到的分享快取空間命中率是不同的。快取空間命中率是指節點在本身的快取空間能夠拿到資料的機率，而分享快取空間命中率是指節點在本身及周圍節點能夠拿到資料的機率。

圖 3 為我們的方法 Stability、RN 和 DAFN 比較 Hit rate 的比較圖。圖中的 X 軸為我們所使用的 Zipf-like 分配所設定的參數 θ，而 Y 軸代表了在每種不同設定的 θ 時的快取空間命中率。

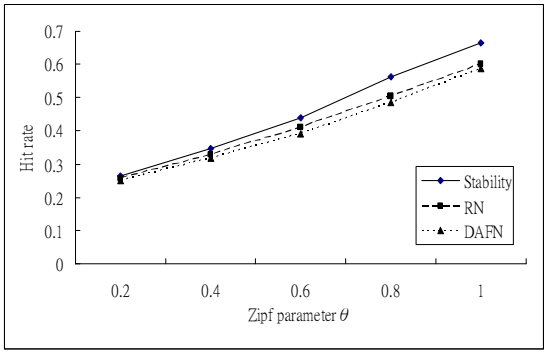

圖 3 Hit Rate 的比較圖

Stability 的分享快取空間命中率是設為 0.6，也就是當節點在本身及周圍節點拿到資料的機率小於 6 成，節點會進行資料快取內容的重置。在這個實驗中，我們藉由改變 Zipf-like 分配中的 θ 值來

觀察每個方法的 Hit rate 值，當 θ 值從 0.2 調整到 1 的過程中，每個方法的 Hit rate 都是逐漸升高的。這代表當資料項的需求機率的變異性越大的時候，節點本身快取空間的命中率也會相對地提升。因為每個節點所需求的資料項就會集中在某些特別熱門的資料上，而節點會把這些熱門的資料放在自己的快取空間中，因此節點在自己空間內的快取命中率也相對地提升。

從圖 3 中可以發現，不論 Zipf-like 分配的 θ 值如何改變，我們的方法的 Hit rate 都是最高的。代表當我們考慮到節點間過去一段時間內的連接狀況來當作我們快取資料的依據時，每個節點從本身的快取空間能拿到資料的機率會比其他兩個方法來得高。

雖然三個快取的資料備份策略都有考慮到節點周圍的其他節點，但是在 DAFN 的方法中，每個節點會替換掉跟周圍節點重複的資料項，而沒有考慮到本身節點是否也是需要該資料項，所以當某節點需要某個已被替換掉的資料項時，該節點就無法從自己的快取空間取得。而在 RN 的方法中，當某節點周圍的節點穩定度都很高的時候，該節點會分享很多的快取空間給周圍的穩定節點，甚至會分享自己所有的快取空間，因此會導致每個節點無法有效的從自己的快取空間中取得資料，常常要從周圍節點來取得自己所需要的資料。

在我們的方法 Stability 中，無論周圍的節點穩定度如何，節點本身所佔的權重都為 1(每個節點穩定度的權重範圍是 0 到 1 之間)；也就是說，每個節點在資料備份時都是擁有絕對的權重，這樣每個節點才可以更有效的利用到自己的快取空間，而不是都要從周圍的節點來取得自己所需的資料。

實驗二：資料可存取性(Data accessibility)的比較

接下來要比較的是 Stability、RN 和 DAFN 三個方法的資料可存取性。我們對於資料可存取性的計算方法為

$$\text{Data accessibility} = \frac{\text{成功請求資料的總次數}}{\text{所有請求資料的總次數}}$$

資料可存取性是指在無線隨意網路下，節點發出請求資料的封包後，能夠成功收到有該資料的其他節點的回應封包，這就代表了一次的成功請求資料。

資料可存取性代表了整個網路環境中，資料可被成功存取的程度，也可以代表節點可以成功拿到資料的比率。所以當資料可存取性較高的時候，在整個網路中資料被利用的程度也是較高的。而當資料可存取性較低的時候，代表整個網路的資料被利用度並不是很好，也就是節點所備份的資料沒有被有效存取。

圖 4 是三個方法 Stability、RN 和 DAFN 的資料可存取性的比較。其中 X 軸代表了 Zipf-like 分配所用的參數 θ，Y 軸代表了每個方法在不同的 θ 下的資料可存取性。

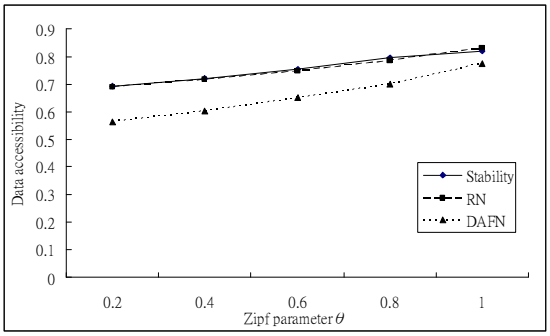

圖 4　Data accessibility 的比較圖

在這個實驗中 Stability 的分享快取空間命中率設為 0.6。在這個實驗中，當 θ 值越小的時候，資料的可存取性也會越小，代表了當節點對資料的存取頻率越平均時，每個節點所備份的資料內容差異性會越大，所以每個節點要從其他節點成功要求到自己所需要的資料的機率也會下降。

在圖 4 中我們發現 Stability 的資料可存取性在不同的 θ 下，也就是在節點對資料的存取頻率的離散程度不同時，Stability 的結果都比 DAFN 好，因為 DAFN 沒有考慮到節點間過去一段時間的連接情況。假如兩個節點之間的連接不穩定，但兩節點卻以對方為資料備份的考量對象，當未來兩節點在很短的時間內離開了彼此的溝通範圍後，兩節點以彼此為考量所備份的資料被存取的機會就會減

少，導致資料可存取性的下降。而我們的方法 Stability 跟 RN 比較的結果是不分上下的。我們認為原因是節點間的距離跟節點間過去一段時間的連接情況有著某些程度的關連，當兩節點的距離很近的時候，這兩節點連接的情況相對地會比較穩定，因為他們一直都在彼此的溝通範圍內，所以兩個方法的資料可存取性會很接近。不過我們認為當兩節點距離很遠的時候，不絕對代表這兩節點的穩定度在過去一段時間內都是很差的，只能說明這兩節點在測量距離的一瞬間時的穩定度沒有很好。

實驗三：請求延遲(Query delay)的比較

第三個實驗要比較的是三個方法的請求延遲的情形。請求延遲的計算方法為

$$\text{Query delay} = \frac{\text{成功回應請求的總跳躍(Hop)數}}{\text{成功請求資料的總次數}}$$

請求延遲這個權數是用來分析在整個網路環境下，每個節點平均要經過幾個跳躍數才可以拿到資料。當請求延遲的值較小時，代表每個節點可以從其他較近的節點取得自己需要的資料，這樣可以耗費掉較少的網路成本，對於整體網路是比較有效益的。

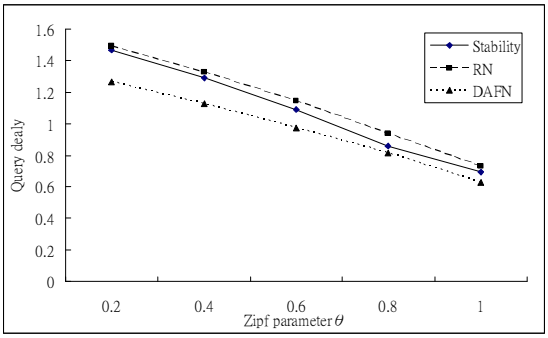

圖 5　Query delay 的比較圖

圖 5 是三個方法的請求延遲的比較。X 軸是 Zipf-like 的參數 θ，而 Y 軸是節點要拿到資料的平均請求延遲時間，也就是要平均要經過多少跳躍數，節點才可以拿到自己需要的資料。我們可以從圖中觀察到，隨著 θ 值的上升，環境中每個節點所備份的資料考量會偏向於較熱門的資料，所以三個方法的請求延遲都會逐漸下降，這是一個合理的現象。

從實驗結果可以發現，我們的方法 Stability 的效能是介於 RN 和 DAFN 之間，Stability 的平均請求延遲是比 RN 好，而略差於 DAFN。Stability 比 RN 的請求延遲較好的原因是 Stability 考慮了節點間過去的一段時間內的連接情況，而 RN 只有考量到兩節點一瞬間的距離來當作兩節點的穩定度。在 Stability 的方法下，過去節點間連接較頻繁或是較穩定時，節點就會為彼此來備份資料。在未來的短時間內兩節點也很有可能在連接，這樣兩節點就可以有效的取得彼此的資料，因此節點間的請求延遲也會相對地降低。而 RN 的請求延遲的結果就沒有 Stability 來得好。

而我們的方法 Stability 的請求延遲的表現略差於 DAFN。原因在於在 DAFN 的方法中，節點間有替換掉重複的資料項，所以可以備份較多不同的資料項；在我們的方法中，節點間不會去替換掉重複的資料項，所以當某節點需要較冷門的資料項時，就比較難從節點本身或是周圍節點的快取空間取得。我們的方法 Stability 沒有去除節點間重複的資料項是因為避免資料可存取性的下降。當兩鄰近的節點有重複的資料項，這兩個節點只會有一個節點會保留該資料項，不過假如這兩個節點之間連接的穩定度不好，可能會造成其中一個節點無法取得該資料項，會導致資料可存取性的下降。因此，在實驗二中 Stability 和 DAFN 資料可存取性的比較來說，我們的方法是優於 DAFN 的。

4　結論

在之前探討資料備份的文獻中，對於討論節點間穩定度的討論不多，而節點間的穩定度對於資料備份的效益來說又是不容忽視的，所以我們提出了以節點間過去一段時間的連接情況來當作衡量節點間穩定度的考量因素。我們認為，兩節點在過去一段時間的連接情況會影響到兩節點未來短時間內的連接情況。因此，我們提出了一個叫做 Stability 的資料備份策略。實驗結果顯示我們考量

穩定度的方法 Stability 在某些方面比 RN 和 DAFN 還來得好。

參考文獻

1. T. Hara, "Effective replica allocation in ad hoc networks for improving data accessibility, " IEEE INFOCOM 2001, vol.3, pp.1568 - 1576, April 2001
2. T. Hara, "Replica allocation in ad hoc networks with periodic data update, "Proc. of Mobile Data Management 2002. pp. 79 – 86, Jan. 2002
3. L. Yin and G. Cao, " Balancing the Tradeoffs between Data Accessibility and Query Delay in Ad Hoc Networks," IEEE Symposium on Reliable Distributed Systems (SRDS), 2004.
4. T. Hara, N. Murakami, S. Nishio: "Replica allocation for correlated data items in ad hoc sensor networks, ". SIGMOD Record 33(1): 38-43 (2004)
5. S. Ishihara, M. Tamor,T. Mizuno,T. Watanabe, "Replication of data associated with locations in ad hoc networks, "in proc of IEEE international conference on Mobile Data Management (MDM2004),p.172.2004
6. J. Cho; S. Oh; J. Kim; H. Ho Lee; J. Lee "Neighbor caching in multi-hop wireless ad hoc networks, " IEEE Communications Letters, Vol. 7 , Issue. 11, Pp. 525 – 527,Nov. 2003
7. Vittoria Gianuzzi: Data replication effectiveness in mobile ad-hoc networks. PE-WASUN 2004: 17-22
8. The Network Simulator - ns-2: http://www.isi.edu/nsnam/ns/