

隨意網路中節省能源的協同資料快取方法

An energy conserving cooperative caching policy for Ad Hoc Networks

呂永和、林新展

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

摘要

由於隨意網路不需要基礎建設的支持，隨意網路可以應用於許多環境。為了提升整體網路的效能，最常採用的方法是快取技術。隨意網路在資料的傳遞方式是透過多點跳躍方式(multi-hop)，所以隨意網路以合作快取的策略來討論是較好的，節點間彼此分享快取空間，更進一步提升系統效能。

由於目前隨意網路中所提出來的合作快取策略中，並沒有以考量每個節點剩餘的能源，如此很可能會發生過度的使用某些節點的狀況，造成縮短某些節點的生命，導致整體網路的完整功能運作的時間縮短，縮短了網路生命時間。因此我們提出了考量能源的合作快取策略，避免過度使用了某些節點能源，以達到延長網路生命的目的，模擬實驗的結果說明我們的合作策略確實能延長網路的生命時間，並且降低了一些請求延遲，以及提高了資料可獲得性。

關鍵詞：隨意網路、合作快取、能源、快取資料

Abstract

Since no pre-existing infrastructure is required, a Mobile Ad Hoc Network (MANET) can be easily deployed in a hostile environment such as a military operation or a disaster recovery. Since the bandwidth of a MANET is very limited, data caching may be useful to enhance the performance of a MANET. Due to the multi-hop nature of a MANET, the existing data caching policies for mobile computing cannot be readily applied. The cooperative caching concept is shown to be suitable for MANETs. With cooperative caching, each node shares its cache space with other nodes in the network to improve the data accessibility of the network.

Since existing cooperative caching policies do not consider the residual energy of each node, nodes with limited energy may exhaust its power very quickly. As a result, the data accessibility of the network will be reduced. In this study, we propose to consider each node's leftover energy to determine the amount of data to be cached. Simulation results show that our method increases the data accessibility and reduces the data access time

Key words: MANET, Cooperative Caching, Energy,

1 緒論

無線隨意網路(Mobile Ad Hoc Network 稱為 MANET)是由二個以上的行動裝置所構成的區域網路,任意兩個行動裝置只要在傳輸的範圍內就可以直接的傳輸資料,不需要透過基地台的協助。由於不需要基礎建設的支持,隨意網路可以應用於許多環境。無線隨意網路最常應用於軍事上,例如位於戰場前方的士兵發現敵人基地,可以把此訊息告訴其他的士兵,彼此可以藉由行動裝置來完成通訊,不需基礎建設的支持完成通訊。

由於在隨意網路中每個客戶端可以任意的移動,隨意網路的拓撲是時常變動的,而且節點間斷線狀況時常發生,這是隨意網路的一個重要的特性。另外在隨意網路中資料的傳輸上是透過多點跳躍(multi-hop),多點跳躍是指傳送時資料需由多人轉送的現象,請求資料的節點與提供資料節點的距離可能有數個跳躍的距離。

由於行動裝置的能源是有限的,為了節省能源花費最常使用快取技術,資料快取(Caching)是常被使用在降低行動計算客戶端的資料請求延遲時間(Query Delay),以及存取資料時的能源損失的技術。目前在傳統行動計算的環境裡,有許多的快取策略,但是這些策略並不適用在隨意網路,由於在傳統行動計算的環境中,資料的傳輸是透過單點跳躍(single-hop),每一個節點只需要考慮快取哪些資料項,對自己會帶來最大的利益,並不需要考量其他節點的需求。相對的而在隨意網路中傳輸資料是透過多點跳躍的,節點與節點之間是有互動的,所以每個節點在快取資料項時,只考量對自己最有利的策略並不是一個好的方式。所以隨意網路的快取策略採用行動計算的快取策略並不恰當。因此研究者提出合作快取的方式[1][2][3][4][5][6][7][8][9]來更進一步的提升快取空間所帶來的利益。合作快取指每個節點不以快取對自己最有利益的資料項,而是去快取對整體最有利益的資料,藉由節點與節點之間彼此分享快取空間,將可減少節點彼此之間快取過多重複的資料項,讓更多的資料項可以更快取

得,達到更進一步的降低請求延遲以及節省能源的花費。

雖然有線網路裡面也有使到合作快取[10]的概念,由於有線網路的合作協定,通常是使用於存在有基礎架構的環境裡,適用於節點是固定、穩定的環境下的,所以當節點移動頻繁時,這些方法並沒有辦法適用於這樣的環境。

在合作快取策略相關研究中,有幾個因素是合作時所需考量的,在[1,2]中作者利用紀錄過去存取資料的資訊,判斷是否要合作快取,紀錄哪些節點擁有比資料來源端更近的資料項,如此以設計合作快取的策略。[3]作者提出了考量節點存取資料的活躍程度,作者提出判斷節點是否較活躍的方法,較不活躍的節點則幫忙較活躍的節點快取資料。上面所介紹的方式是屬於被動的合作快取策略。在[4]中作者三個方法為 SAF、DAFN、DCG,SAF 為不合作策略,DAFN 方法避免過多的重複資料、此文章主要考量的因素為節點對於資料的存取機率,以及不與周圍節點有過多重複資料的考量。[6][7]考量了節點的連結穩定度,若連結程度不穩定,則降低合作程度,若連結程度穩定,則提高合作程度。[8]也是考量了節點的連結穩定程度,作者以考量某些節點之間,彼此會有相同的移動特性,因為在隨意網路中通常以小隊為單位去完成任務,所以作者考量了節點間存在相同的移動特性,以此觀點設計合作策略。

由於目前相關於合作式快取的研究[1][2][3][4][5][6][7][8][9]中,如上面所介紹的想法,並沒有考量每個節點剩餘的能源作為合作快取策略。若是沒有考量每個節點剩餘的能源時,很可能會發生某些熱門的資料項都由特定的節點來合作快取,這樣的話很容易會導致這些特定的節點,必須時常傳輸這熱門的資料項來提供給其他的節點,造成這些節點將很快把能源消耗完,使這些節點的生命較早結束。更嚴重的是,這些節點的陣亡會造成整體網路的完整功能運作的時間縮短,換句話說就是縮短了網路的生命時間。所以我們將避免發生如此的情形,我們希望公平的使用每個節點的能源,能量多的節點負責工作量多一點,能量少的

節點負責工作量少一點，藉此達到延長節點的生命。所以我們提出一種考量能源使用的合作快取策略。在我們的策略中，除了能有合作快取所帶來的提升成功取得資料的比例以及降低請求的延長外，也希望達到平均的使用每個節點的能源，使整個網路的生命延長。在下一章我們將介紹考量能源的合作快取策略。

2 考量能源的合作快取策略

2.1 隨意網路環境假設

我們首先介紹環境的假設以及一些基本的符號說明。

1. 我們假設網路的節點數量有 $|M|$ 個，每一個節點都有編號，例如節點 i 我們標號成 M_i ，每個節點的編號如下 M_j ($1 \leq j \leq |M|$)，所有的節點集為 $M = \{M_1, M_2, \dots, M_{|M|}\}$ ，每個節點具有相同的能力，且節點可以任意的移動，每個行動裝置的初始能源為 $|E|$ 。
2. 所有的資料項目總共有 $|D|$ 個，每一個資料項的大小相同，每一個資料項有一個唯一的編號，例如資料項 i 標號成 D_i ，每個資料項的編號如下 D_j ($1 \leq j \leq |D|$)，所有的資料項目集合為 $D = \{D_1, D_2, \dots, D_{|D|}\}$ 。
3. 每個客戶的快取記憶體空間大小為 $|C|$ 個空間，換言之每個客戶可以快取 $|C|$ 個資料項，我們假設快取記憶體空間小於總資料項目 $|D|$ 。
4. 每個客戶對於每一個資料的需求機率是已知的，且機率是不會變動的。
5. 每個資料項的內容是不會異動的。

我們的環境假設在無線隨意網路中，每個節點會對於資料項提出需求，對於資料項的需求，可以使用自己的快取資料來回應需求，也可以送出請求給其他節點，藉由其他的節點傳送資料來回應這個需求。

2.2 問題描述

我們的問題是討論在隨意網路中，藉由合作快取的概念，決定每個節點的快取空間中，需快取哪些資料項目，使得延長隨意網路的生命時間，提升在隨意網路的資料可得性，以及降低請求延遲的目的。根據[7]說明，在隨意網路中決定每一個節點的快取內容，以增加合作快取的效益，想找出最佳解此問題是一個 NP-hard 的問題。例如，由於移動性的因素，我們在某個時間點經過複雜的計算找出了最佳解，但不久後可能因為節點的移動導致最佳解又改變了。由於行動裝置計算能力有限，太複雜的計算並不適用。因此我們的策略是找出一個啟發式(heuristics)的合作快取策略，其目的是找出一個不需要複雜的運算卻有不錯的效果的合作方式。

2.3 合作快取與能源使用

合作快取的基本概念，每個節點不一定會快取對自己最有利益的資料項，而是去快取對周圍整體最有利益的資料，這是合作的目的與精神。但是我們觀察到，當某一個節點快取了對周圍整體有高利益的資料項，該節點就會常常需要把快取資料傳給其他節點，會花費較多的能源。也就是說，若是一個快沒能源的節點，卻快取了大量有價值的資料項時，容易縮短節點的生命，所以我們認為節點剩於能源較少應該要降低合作關係，避免過度使用節點能源。

所以我們的策略為將總快取空間分成兩部分，一部分是合作快取空間，另一部分是不合作的快取空間，我們將利用決定節點的合作快取空間大小，來促使能源較多的節點就負責多點工作，而節點能源較少就負責少一點的工作量。如此以達到公平的使用每個節點剩餘的能源，而不會造成過度的使用某些節點，好處就是藉由公平的使用節點能源來達到延長節點的生命時間以及到整個網路的生命延長，接著我們將介紹如何決定合作快取的空間。

2.3.1 合作空間的決定

如上討論，合作快取空間的大小，將受到節點剩餘能源的影響，為延長網路的生命剩餘能源較多的節點，應該拿出較多的空間來合作快取，負責工作量多一點，剩餘能源較少的節點，應要拿出較少的快取記憶體空間來合作快取，負責工作量少一點，拿出的合作空間比例，要與剩餘的能源成正比。所以當我們想知道節點 i 要拿出多少合作空間時，我們利用下面的式子決定合作的空間，

$$Cc(i) = \left\lfloor |C| * \frac{E_i}{|E|} \right\rfloor,$$

$Cc(i)$ 為節點 i 拿出來合作快取的空間大小， $|C|$ 為節點 i 快取記憶體空間大小， E_i 為節點 i 目前剩餘的能源， $|E|$ 為每個節點剩餘的能源。當節點 i 剩餘能源多時，帶入式子則會拿出較多的合作空間，當節點 i 剩餘能源少時，帶入式子則會拿出較少的合作空間，所以我們剩餘的工作是決定合作空間與不合作空間所要快取的資料項。

2.3.2 決定不合作空間的快取內容

首先我們先討論如何決定在不合作的快取空間，要快取哪些資料項。對於這些不合作的快取空間，節點只需考量哪些資料對自己最有利益的，而不考量周圍節點的任何需求。因此節點便把自己需求機率較高的資料項，至於不合作區域中，直到不合作的快取空間大小 $|C| - Cc(i)$ 滿了為止。換句話說，假設節點 i 不合作的快取空間所快取的資料項集合是 $DCs(i)$ ，其內容為 $DCs(i) = D_{\max} [|C| - Cc(i)] \{ P_i(1), P_i(2), \dots, P_i(|D|) \}$ ，其中 $\{ P_i(1), P_i(2), \dots, P_i(|D|) \}$ 為節點 i 對每一個資料的需求機率， $|D|$ 為資料項目總個數。 $D_{\max} [k] \{ \text{set } A \}$ 的意思就是從集合 A 中，取出前 k 大的元素其對應的編號。也就是說 $D_{\max} [|C| - Cc(i)] \{ P_i(1), P_i(2), \dots, P_i(|D|) \}$ 的意思是取 $P_i(1), P_i(2), \dots, P_i(|D|)$ 中，這 D 個資料機率值排名前 $(|C| - Cc(i))$ 個較大的資料項。對於不合作部份的資料，我們將不回應其他節點的請

求。

2.3.3 決定合作空間的快取內容

對於合作快取空間內，我們對於每一個資料項都給一個值，當這一個值越高表示合作快取帶來的利息越高，是較適合合作快取的資料項，這個值越低表示合作快取這一個資料項的效益較低。而合作快取的資料項就是由這個值來決定。當我們合作快取價值高的資料，表示這個資料項對於整體是比較有利的，我們就把這個資料項快取起來。決定此利益值的方式如下：假設節點 i 周圍的節點有 k 個，節點 i 周圍的節點表示為節點 j ， $j = \{1, 2, \dots, k\}$ ，總共有 k 個鄰居，我們用 $profit_i(w)$ 來表示節點 i 快取資料項 w 合作快取的利益值， $profit$ 值算法如下

$$profit_i(w) = P_i(w) + \sum_{j=1}^{j=k} P_j(w) * (1 - \frac{R_{ij}}{R}),$$

其中 $P_i(w)$ 的意思就是節點 i 對資料項 w 的需求機率， $P_j(w)$ 的意思就是節點 i 的一個周圍節點 j 對資料項 w 的需求機率， R 為每個節點的傳輸距離半徑， R_{ij} 為節點 i 與周圍節點 j 兩點的距離，若節點 i 與節點 j 距離越近則表示鏈結較穩定，則 $(1 - \frac{R_{ij}}{R})$ 的值便越高。相反的，若節點 i 與節點 j

的距離越遠則值便越低，則在未來兩個節點的鏈結較不穩定。所以乘上些數值就是考量斷線的比重，

$$P_i(w) + \sum_{j=1}^{j=k} P_j(w) * (1 - \frac{R_{ij}}{R})$$

的意義為節點 i 對資料項 w 的需求，加上所有周圍節點對此資料項 w 的需求，再乘上考量斷線的權重。因為合作快取的資料項是對整體而言是較有利益的，當自己以及周圍使用頻率較高的，將此資料快取起來對整體而言是較有利益的，所以我們使用 $profit$ 值來當做合作快取的指標。假設節點 i 合作的快取空間所快取的資料項集合是 $DCc(i)$ ，其快取資料項目為

$$DCc(i) = D_{\max} [Cc(i)] \{ profit_i(D_1) \},$$

$profit_i(D_2), \dots, profit_i(D_{|D|})$ } ,

意思為從所有資料項選出前 $Cc(i)$ 大的 $profit$ 值。由於合作快取的資料集合, 需與不合作空間快取的資料項沒有重複下, 所以需有條件限制

$$\forall D_x \in DCs(i), \forall D_y \in DCc(i), D_x \cap D_y = \phi .$$

2.3.4 替換與周圍合作快取重複的資料項

因為上一小節決定了合作空間快取資料項目後, 因我們很可能與周圍的節點有快取了重複的資料項, 所以我們這一小節將介紹處理與周圍節點, 快取相同資料的方式。假設節點 i 合作快取空間, 快取了資料項 w , 周圍節點 j 合作快取空間, 也快取了資料項 w , 為了充分使用快取記憶體空間, 我們將使這兩個快取空間, 快取不相同的資料項。因此我們將節點 i 合作快取資料項 w , 給定合適指標為 $Cval_i(w)$, 我們將節點 j 合作快取資料項 w , 給定合適指標為 $Cval_j(w)$, 接著我們將比較 $Cval_i(w)$ 以及 $Cval_j(w)$ 這兩個指標, 若節點 i 快取 w 的 $Cval_j(w)$ 值較高, 我們認為為由節點 i 快取 w 較適合; 若當節點 j 快取 w 的 $Cval_j(w)$ 值比較高, 我們認為由節點 j 快取 w 比較適合。 $Cval$ 值較低的節點就使用其他的資料項來替換, 我們之後會說明要用哪一個資料項替換。

對於 $Cval$ 的設計, 我們第一個考量因素為哪個節點快取 w 對整體較有利益, 若當節點 i 快取 w 的 $profit$ 值, 大於節點 j 快取 w 的 $profit$ 值時, 我們認為由較有利益的節點 i 來快取是較合適的。另外一個考量因素, 假設節點 i 最高的 $profit$ 值為資料項 w , 合作快取最有價值的資料項為 w , 假設周圍節點 j 合作快取最高 $profit$ 值也是資料項 w , 若兩個 $profit_i(w)$ 與 $profit_j(w)$ 是一樣大的, 表示說 w 對兩個節點來說都是一樣重要的資料項。假設節點 i 有較多的剩餘的能源, 假設節點 j 剩餘的能源很少, 若熱門的資料項 w , 由節點 j 合作快取時, 在不久之後, 節點 j 的能源就會消耗完了, 因為熱門的資料項 w 要經常傳輸給其他節點, 若節點 j 能源使用完了, 同時也會導致對整體重要的資料項 w 遺失。所以我們認為熱門的資料項 w , 應該由剩於

能源較多較穩定的節點 i 來快取, 較不容易沒電較穩定, 避免由能源少較不穩定的節點快取熱門資料。經由這兩個因素的考量, 所以我們設計節點 i 快取資料項 w 的 $Cval_i(w) = profit_i(w) * \frac{E_i}{|E|}$, 其中 E_i

為節點 i 剩餘的能源, $\frac{E_i}{|E|}$ 為節點剩下多少能源轉

換對應到 0 與 1 中的範圍內。 $Cval$ 值越大表示這個節點合作快取這個資料的利益越高, 且剩餘能源越大較穩定較佳。所以當有重複的合作快取資料時, 我們採用 $Cval$ 當指標, 決定哪個節點較合適。

所以當節點 i 藉由 $profit$ 值決定好合作快取的資料項後, 將合作快取的資料項, 以及每個合作快取資料的 $Cval$ 值, 廣播一個跳躍給周圍的所有節點 j 。當某周圍節點 j 收到廣播訊息後, 則比較自己的合作快取的空間內, 與節點 i 合作快取的資料有沒有重複, 如果沒有重複的資料時, 節點 j 則紀錄節點 i 合作快取了哪些資料, 為了知道周圍節點已經快取了哪些資料。如果合作快取的資料有重複時, 則比較相同資料項的 $Cval$ 值, 若節點 j 的 $Cval$ 值較大則表示較合適, 則不做任何處理; 若重複的資料項節點 j 的 $Cval$ 值較小, 則節點 j 將該重複的資料項去除, 並且找出自己以及周圍節點尚未合作快取的資料項, 選出 $profit$ 值最高的資料項來替換的重複的資料。

2.4 合作快取策略

每個節點每 L 秒定期與周圍的節點溝通一次, 每當 L 秒時在新的溝通前, 會將上次所紀錄的資訊清除, L 為系統的參數, 溝通內容與步驟如下

1. 每個節點 i 將節點編號 M_i 和所在位子, 以及本身需求的機率 $\{P_i(1), P_i(2), \dots, P_i(|D|)\}$ 廣播一個跳躍傳送給所有周圍的節點。
2. 當每個節點 i 收到周圍節點的剩餘電量以及需求機率時, 計算兩點的距離, 並將對每一個資

料項 w 的需求機率加總起來計算 $profit_i(w)$ ，對每個資料項 w ， $1 \leq w \leq |D|$ 。

- 當收完所有周圍節點的訊息後，每個節點 i 計算要拿出多少空間來合作，計算

$$Cc(i) = \left\lceil |C| * \frac{E_i}{|E|} \right\rceil。$$

- 決定快取的資料項，在不合作的空間裡 $DCs(i)$ 就去快取對自己需求機率最高的資料項，

$$DCs(i) = D_{\max} [|C| - Cc(i)] \{ P_i(1), P_i(2), \dots, P_i(|D|) \}$$

在合作的空間裡快取的資料項內容如下。

$$DCc(i) = D_{\max} [Cc(i)] \{ profit_i(D_1), profit_i(D_2), \dots, profit_i(D_{|D|}) \}$$

where $\forall D_x \in DCs(i), \forall D_y \in DCc(i)$ 對每個 $D_x \cap D_y = \phi$ 。

- 決定好合作快取資料項後，計算 $Cval_i(w)$ ，where $w \in DCc(i)$ ，將 $DCc(i)$ 及 $Cval_i(w)$ 廣播一個跳躍(Hop)出去。

- 當節點 j 收到周圍節點 i 的溝通信息時，判斷與周圍的節點有沒有合作快取重複的資料項目，

若沒有重複的資料項

紀錄周圍節點合作快取了哪些資料項。

若有重複的資料項 w :

判斷周圍與自己的合作快取資料是否快取了所有的資料項。

否，與周圍節點未合作快取了所有的資料項

若 $Cval_j(w) < Cval_i(w)$

從周圍尚未合作快取的資料項中找出 $profit$ 最大的資料項快取。

若 $Cval_j(w) \geq Cval_i(w)$

不做任何處理。

是，與周圍節點已合作快取了所有的資料項

在與節點 i 合作快取的資料，以及節點 j 自己已經快取的資料中，在

不重複條件下找出最大 $profit$ 值的資料取代資料 w 。

3 實驗結果與分析

3.1 實驗參數

我們使用的模擬軟體是 ns2[11]，ns2 是一種離散事件的模擬器，可以用來模擬 TCP，路由協定，群播等有線無線網路的環境。接著我們列出模擬的參數設定。

項目	值
傳輸的距離(R)	250 公尺
資料的傳輸速度	2M/s
總節點	50 個
網路的範圍	1200 公尺*800 公尺
資料總數量	100 個
資料大小	2Kb
快取空間大小	5% ~ 25%
Zipf-like theta (θ)	0.2~1.0
每個節點的能源	5 焦耳
平均請求時間	1 秒
MAC	IEEE 802.11
資料來源端個數	1
節點移動速度	6 公尺/每秒

表格 1 模擬參數設定

每個客戶端對於資料項 k 的需求機率為

$$P(X = k) = \begin{cases} \frac{\left(\frac{1}{k}\right)^\theta}{\sum_{i=1}^n \left(\frac{1}{i}\right)^\theta} & 1 \leq k \leq n, \\ 0 & otherwise. \end{cases}$$

θ : Zipf distribution 的參數，用來表示資料項請求模式的請求次數的分配偏斜程度。我們假設 $L=50$ 秒。傳輸資料能源的花費設定由[13]可知。

3.2 衡量指標

由於要比較我們的合作策略有沒有提升資料的可獲得性，以及請求延遲以及延長網路的生命時間，我們將利用下面的衡量指標來判斷，

$$1. \text{hit ratio} = \frac{\text{自己快取空間取得資料的總數量}}{\text{所有請求資料的總數量}}$$

$$2. \text{請求延遲} = \frac{\text{所有成功回應請求資料的總跳躍數(HopCount)}}{\text{所有成功回應請求資料的數量}}$$

$$3. \text{資料可獲得性} = \frac{\text{所有成功回應請求的總數量}}{\text{所有請求資料的總數量}}$$

4. 網路生命時間: 由[12]定義為網路上第一個節點將電源消耗完的時間。我們在實驗部分將會把網路的模擬時間以及對應的節點的存活數量列出來比較，接著我們也會列出系統完成的工作量與存活節點的數量做比較。

3.3 實驗分析

我們與 T. Hara 所提出來的 SAF 以及 DAFN 方式做比較[4]，SAF 就是節點不合作的策略，節點指考量對自己最有利益的資料，DAFN 是加強合作的策略，假設我們所提出來的快取策略稱為 Energy。

我們首先觀察節點使用到自己快取空間的機率(Hit rate)，如圖 1，我們可以觀察到只考量自己需求的 SAF 方法 hit rate 最高，因為不考量其他節點的需求 hit rate 是最高的，我們的方法與 DAFN 的 hit rate 差不多，雖然 SAF hit rate 最高，但我們來觀察請求延遲是不是就較佳呢？

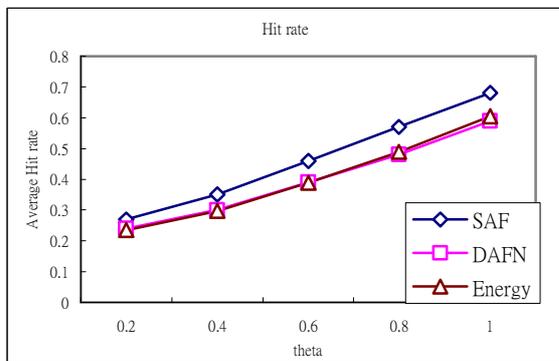


圖 1 Hit Rate 的比較(cache size=20%)

在圖 2 中，X 座標是指 zipf-like 的 theta，Y 座標是指平均拿到資料需要經過多少距離。觀察得知我們的合作策略與 SAF 以及 DAFN 相較之下，我們平均拿到資料項的距離是較短的，我們認為雖然不合作的策略 SAF，雖有部分的資料可以較快的從自己的快取空間取得，但是對其餘部分的資料項需到較遠的節點擷取資料。而我們合作的策略，雖有部分資料項取得時間較差於不合作的策略，但是對於 hit miss 的資料項是會較快取得的。由模擬的結果顯示，我們的合作快取策略請求延遲是較低的。還有一個原因是因為我們有設計節點不合作的部分，會讓請求延遲降低。另一方面我們也可以觀察到隨著 theta 值的降低，三個方法的資料平均取得距離就會提高，這也是合理的現象。

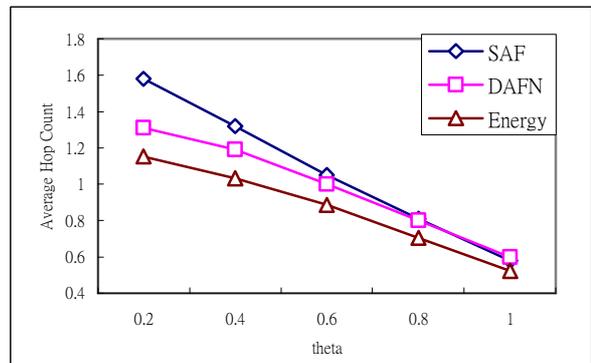


圖 2 theta 對於資料請求延遲圖，(cache size=20%)

我們的策略與其他策略比較起來有較低的請求延遲，其原因我們認為是對於在自己快取無法取得的那一部分資料，可以在周圍取得可以在較近的地方取得，因為有較佳的合作快取效果有較佳的區域性。所以接著我們觀察對於 hit miss 的部分，需要經過多少的距離可以取得，圖 3 X 座標是 zipf-like 的 theta，Y 座標是對於自己快取空間無法取得的資料那部份，其平均拿到資料需要經過多少距離，可以看到在我們的方法中，對於 hit miss 那部分資料可以較快取得，所以我們的方法才會有較佳的請求延遲。

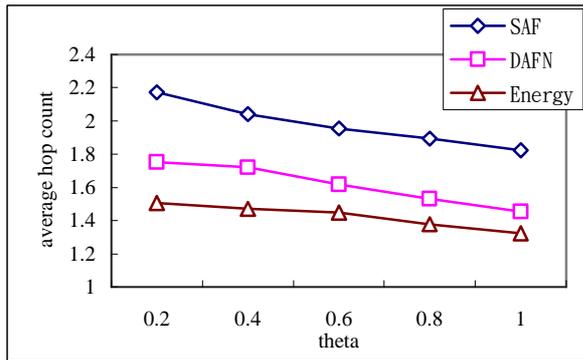


圖 3 對於 hit miss 資料之請求延遲圖，(cache size=20%)

圖 4.是在不同的快取空間對於資料請求延遲的影響，由模擬結果發現我們的方法在每個快取空間都有較低的請求延遲，我們認為跟上述是相同的原因，我們可以發現增加快取空間可以降低請求延遲，有相當大的幫助。

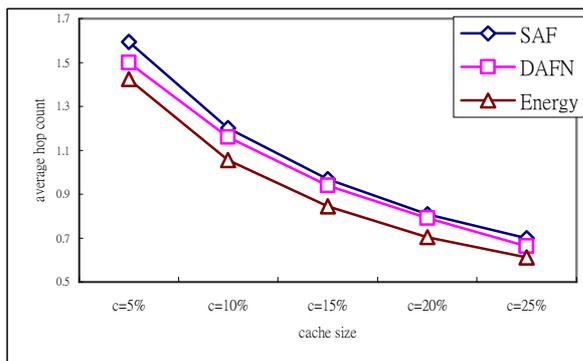


圖 4 快取空間對於資料請求延遲的圖，(theta=0.8)

接下來我們來觀察節點存活的狀況，圖 5 是節點的存活狀況以及模擬時間的對照圖，X 座標是系統的模擬時間，Y 軸是還有多少百分比的節點還存活著(live)，我們觀察到我們方法剛開始節點的死亡時間較晚，以第一個節點死亡時間為網路的生命時間的定義，我們的方法是有較長的網路生命時間的，有較長時間擁有完整節點功能的運作時間。我們認為方法中有考量節點的能源，能源較少時降低合作空間，負責工作量少使得存活時間較長，以及因為節點間有較多的合作能源能使用較平均。雖在後半段的時間我們的方法有較多的節點死亡了，我們認為是因為我們的方法請求延遲較短，所以完成的工作量速度快，所以我們完成了較多的資料請求，因此能源的消耗也相對的較多，所以會產生在後半段時間有較多的死亡節點。

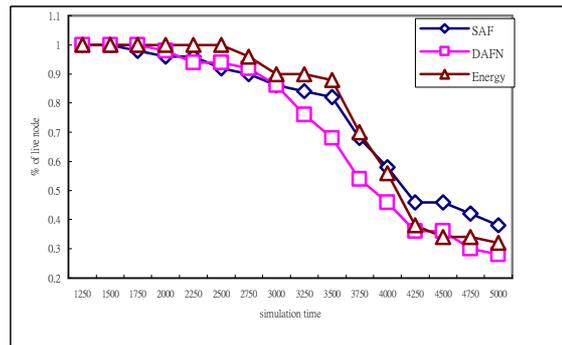


圖 5 節點的存活狀況，(c=20%,theta=0.8)

所以我們換個角度來看網路的生命時間，如圖 6，我們 X 座標是完成請求的總數量，Y 座標是還有多少百分比的節點存活著，1k 是指成功完成 1000 個請求，可以發現在相同的工作量下我們的方法都是有較多的節點存活著的。在完成相同的工作量下，我們有較多的節點存活，我們的網路生命時間較長。

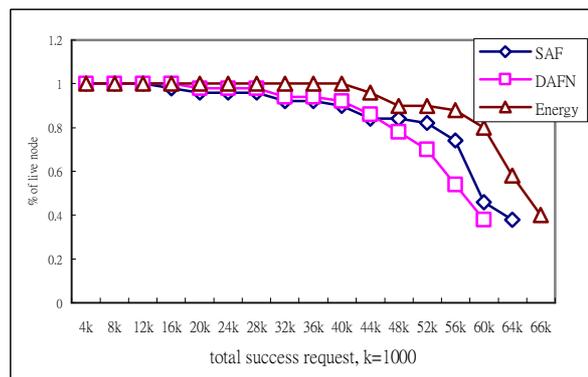


圖 6 相同工作量下的節點的存活狀況，(c=20%,theta=0.8)

最後我們比較資料可獲性的結果，圖中 7 X 座標是 zipf-like 的參數 theta 的變化，Y 座標是節點平均資料可獲得性的值，由圖我們可得知在變動的 theta 值中，我們方法的資料可獲得性都較優於 SAF 以及 DAFN 來，其原因我們認為是我們的方法中，有處理與周圍重複合作快取資料，所以有較多種類的資料可以在周圍節點取得。圖 4.12 中可知當資料的熱門程度與冷門程度偏移越小，即使用者對資料項的喜好程度較平均時，我們可以發現三個方法的資料可獲得性都會往下降，我們認為因為

快取空間有限，當每個資料需求越平均會讓快取空間的功效降低，所以三個方法都會往下降。

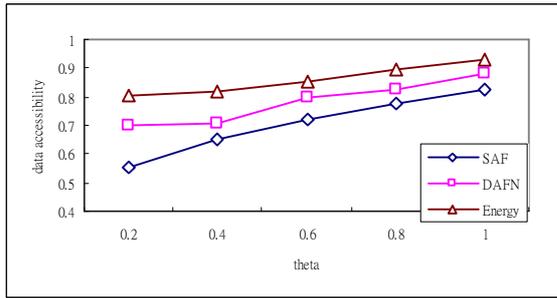


圖 7 theta 對於資料可獲得性的圖，(cache size=20%)

4 結論

為了提升隨意網路的系統效能，許多文獻提出許多合作快取策略，但是除了提升系統效能之外，公平的使用每個節點的能源，延長節點的生命時間也是很重要的議題。在目前隨意網路的合作快取策略，並沒有以能源為出發點做考量，本篇論文提出以考量能源的合作快取策略，由模擬結果顯示，我們的合作快取策略不僅減少了請求延遲以及資料的可獲得性之外，也延長了網路的生命時間。

ACKNOWLEDGEMENT

本論文由國科會研究計畫補助完成，計畫編號為 NSC-94-2213-E-011-064

參考文獻

1. L. Yin and G. Cao, "Supporting cooperative caching in ad hoc networks," INFOCOM 2004, Volume 4, pp. 2537 - 2547, 7 March 2004.
2. G. Cao, L. Yin and C.R. Das, "Cooperative cache-based data access in ad hoc networks," IEEE Computer, Volume 37, Issue 2, pp. 32 - 39, Feb 2004.
3. J. Cho, S. Oh, J. Kim, H. Lee and J. Lee, "Neighbor caching in multi-hop wireless ad hoc networks," IEEE Communications Letters, Vol. 7, Issue 11, Pp. 525 - 527, Nov. 2003.

4. T. Hara, "Effective replica allocation in ad hoc networks for improving data accessibility," IEEE INFOCOM 2001, vol.3, pp.1568 - 1576, April 2001.
5. T. Hara, "Replica allocation in ad hoc networks with periodic data update," Proc. of Mobile Data Management 2002. pp. 79 - 86, Jan. 2002.
6. T. Hara, Y.-H. Loh and S. Nishio, "Data replication methods based on the stability of radio links in ad hoc networks," DEXA 2003, pp. 969 - 973, Sept. 2003.
7. 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.
8. J.-L. Huang and M.-S. Chen, "Exploring Group Mobility for Replica Allocation in a Mobile Ad-Hoc Network," Proc. of the ACM 12th International Conference on Information and Knowledge Management (CIKM-03), Nov. 2003.
9. F. Sailhan, V. Issarny, "Cooperative Caching in Ad Hoc Networks," The 4th International Conference on Mobile Data Management, MDM'03. pp. 13 - 28, 2003.
10. D. Wessels and K. Claffy, "ICP and the Squid Web Cache," IEEE Journal on Selected Areas in Communication, pp. 345-357, 1998.
11. The Network Simulator -ns-2: <http://www.isi.edu/nsnam/ns/>.
12. C.F. Chiasserini, "An energy-efficient method for nodes assignment in cluster-based Ad Hoc networks," Wireless Networks Volume 10, Issue 3, pp. 223 - 231, May 2004.
13. L. Feeney and M. Nilsson, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment," in Proceedings of IEEE INFOCOM, Anchorage, AK, 2001.