An Energy Conserving Cooperative Caching Policy for Ad Hoc Networks

Yungho Leu and Hsin-Chang Lin
National Taiwan University of Science and Technology/ Department of Information Management, Taipei, Taiwan
Email: yhl@cs.ntust.edu.tw

Chi-Chung Lee*
Chung Hua University/Department of Information Management, Hsinchu, Taiwan
Email: leecc@chu.edu.tw

***Abstract*—Without pre-existing infrastructure, a Mobile Ad Hoc Network (MANET) can be easily deployed in a hostile environment such as a military operation area or an area undergoing disaster recovery. With limited bandwidth, the data caching technique may be useful to enhance the performance of a MANET. The cooperative caching scheme is shown to be suitable for MANETs. With cooperative caching schemes, each node shares its cached data items with other nodes in the network to improve the data availability of the network. Unfortunately, the existing cooperative caching schemes did not consider the energy consumption of a mobile node. A mobile node with limited residual energy may exhaust its power very quickly. As a result, the data availability of the network will be degraded. In this paper, we propose an energy conserving cooperative caching scheme which allows a mobile node with more residual energy to cache more data items for other nodes than a mobile node with less residual energy. Simulation results show that the proposed scheme enhance the data availability by reducing the response time of a query and extending the lifetime of the network.**

***Index Terms*—MANET, cooperative caching, energy conservation, data caching**

I. Introduction

A mobile ad hoc network (MANET) is a self-configuring infrastructureless network of mobile devices connected by wireless communication medium. The infrastructureless characteristic makes it easy to deploy a MANET in a hostile environment such as a battle field or an area undergoing disaster recovery [1]. Many research issues on MANETs have been addressed by researchers [2-5]. One important issue that receives a lot of attention from the researcher community is data caching [5-11]. With data caching, a mobile node stores its previously retrieved data items from the data source in its cache to answer the subsequent queries. In a MANET, the data communication between any two mobile nodes is of multi-hop nature. That is, the data transmission between any two mobile nodes is done by mean of hop by hop forwarding of the nodes in between the two mobile nodes. It is obvious that the response time, measured by the number of hop in answering a query, is reduced if a mobile node answers a query using the data items from its local cache or from the caches of its neighboring mobile nodes. Data caching technique can be categorized as local caching and cooperative caching. In local caching, each mobile node caches it's the data items that is only used for answering its own queries while in cooperative caching each mobile node not only caches data items for its own queries but also caches data items for its neighboring mobile nodes [8,9]. Notice that since the cache space of a mobile node is limited, to cache all the required data items of a mobile node in its local cache is impossible. Fortunately, the cooperative caching technique is well-suited for tackling this problem. In cooperative caching, a mobile node first checks its local cache then checks its neighboring mobile nodes for the required data items to answer a query. In this way, the cache space of all the neighboring mobile nodes can be treated as the cache space of the mobile node so as to extend the size of the cache of the mobile node.

The power of a mobile node is limited because it is powered by a battery. Therefore, the energy conservation issue becomes an important research problem in MANETs [12-15]. The cooperative caching also benefits the energy conservation of a mobile node. In MANETs, data items are usually located at a data server and the mobile nodes request data items from the data server. In answering a query, the multi-hop nature of communication in MANETs forces the mobile nodes in between the requesting mobile node and the data server to consume energy for relaying the request and the data items. With the cooperative caching, a requesting mobile node may find its required data items from the caches of its nearby mobile nodes so as to reduce the hop count from the requesting node to the hosting mobile node of the required data items. This, in turn, reduces energy consumption.

Much work [5-15] has been done on cooperative caching in MANETs in the recent years. However, only a few researches take the energy consumption of the mobile nodes into consideration [12-15]. If a mobile node caches many data items that are of high interest for its neighboring mobile nodes, it will consume its energy

* Corresponding author

doi:10.4304/jsw.8.9.2127-2133

very quickly due to the sharing of the cached data items. Very soon, the mobile node will consume all of its energy so as to degrade the performance of the MANET. In this paper, we define the lifetime of a MANET to be the period of time between the starting of the MANET to the time when the first mobile node consumes all of its energy. To extend the lifetime of a MANET, we propose to consume the energy of all mobile nodes uniformly when deciding which mobile node to cache a data item. In the proposed policy, a node with more residual energy caches more data items for its neighbors than a mobile node with less residual energy. The experiments show that the proposed policy not only reduces the response time but also extends the lifetime of the MANET.

The rest of this paper is organized as follows. In Section 2, we review the related works of this paper. In Section 3, we present the energy conserving cooperative caching policy. The performance analysis of the proposed policy is given in Section 4 and finally Section 5 concludes this paper.

II. RELATED WORK

Many methods have been proposed for cooperative caching in MANETs in the recent years [5-11]. For example, in [7] Hara and Madria [7] proposed the static access frequency data replication method (SAF) and the dynamic access frequency and neighborhood data replication method (DAFN) to enhance the data accessibility of a MANET. With SAF, a mobile node caches its top-n highly accessed data items in its cache. Note that in SAF the data caching is independent in the sense that a mobile node does not consider the data requirements of the other mobile nodes. The drawback of SAF is that two neighboring nodes may cache the same data so as to reduce the utility of the cache spaces of the mobile nodes. In DFAN, the data items are first cached according to the SAF policy. Then, in every T seconds, a mobile node communicates with its neighbors for their cached data items; once duplication on a cached data item is found, the node with the least access frequency on the data item deletes the data item from its cache. To utilize the free cache space, the node then caches the data item whose access frequency is the highest among its own data items that are still not in the cache. In [8-9], two cooperative caching methods, the cache data and the cache path, were proposed by Yin and Guo. In the cache data method a mobile node caches an accessed data items for later query request while in the cache path method a node caches the path to a cached data item for a later query request. In [10-11], the stability of the wireless connection between two mobile nodes is taken into account. If the connection between two mobile nodes is unstable, the two nodes do not cooperate in data caching. In contrast, if the connection is stable, the two nodes cooperate in data caching.

The researches on cooperative caching that consider energy consumption of the mobile nodes are reviewed in the following. In [12], Hara proposed a set of energy aware data replication methods. In WEA-B (Weighted Expected Access-Battery), a mobile node can adjust the proportion of the cache space for itself to the cache space for its neighbors according to the amount of its remaining battery power. In WEA-H (Weight Expected Access-Hop), besides giving weights to favor caching local accessed data items or remote accessed data items, a mobile node can also prevent to cache data items for far away mobile nodes. Pushpalatha et al. [13] proposed a replication model which relocates replicas of a mobile node to the mobile nodes with high remaining battery energy when the energy of the node is less than a predefined threshold. Mukilan and Wahi [14] developed a data replication algorithm based on node energy and mobility, in which query delay, energy consumption and data availability are considered. Chan et al. [15] analyzed the impact of energy consumption on designing a cache replacement policy and formulated the proposed cache replacement problem as a 0-1 knapsack problem. A dynamic programming algorithm and a heuristic algorithm are presented to solve the formulated problem.

III. ENERGY CONSERVING COOPERATIVE CACHING POLICY

In this section we present the energy conserving cooperative caching policy. We will first present the system model of a MANET with data caching. Then, we will give the details of the proposed policy.

3.1 The System Model

To ease the presentation of the proposed policy, we present the assumptions and notations in the following.

1. Assume that a MANET consists of n nodes. The ith mobile node is denoted by M_i. We use M to denote the set of all mobile nodes. That is, $M = \{M_1, M_2, \ldots, M_n\}$. Each node is equipped with the same initial power in its battery, which is denoted by E. Furthermore, each node has the same cache size, denoted by C.
2. There are k data items in the MANET. These data items are of the same size and are read-only. The data items are numbered and we use the number of a data item as its identifier. The set of all data items in MANET is denoted by D. That is, $D = \{1, 2, \ldots, k\}$.
3. Every mobile node has an access pattern over the data items in D. The access pattern for mobile node i, denoted by AP_i, is a vector of probabilities ($P_{i1}, \ldots, P_{ik}$), where P_{ij} is the probability for mobile node i to request data item j. Notice that the summation of the access probabilities of a mobile node on all data items is equal to one. That is, $\sum_{j=1}^{k} p_{ij} = 1$.
4. To ease the discussion, when a mobile node requests a data item, it is called the center node, while the mobile nodes within the communication range of the requesting node are called the neighbors of the center node. Note that the center node and its neighbors will cooperate to determine where to cache a data item in the caches of mobile nodes in the MANET.

3.2 The Energy Conserving Cooperative Caching Ploicy

In cooperative caching, a mobile node acting as a neighbor of a certain center node may face the dilemma that most of its energy is consumed in the activities of sharing its cached data items with the center node. When the remaining power of the mobile node is low, to continue sharing its cached data items may endanger the lifetime of the network. To counter this dilemma, we propose to allocate the cache space of a mobile node for its neighbors according to the remaining power of the mobile node.

To distinguish the cache space for a mobile node itself from the cache space for its neighbors, we call the cache space for the mobile node *the local cache space*, and the cache space for the neighbors *the cooperative cache space*. The center node caches an accessed data item in its cooperative cache if it will share the data item with its neighbors. In contrast, the center node caches the accessed data item in its local cache if the accessed data item is not shared. In the proposed policy, the size of the cooperative cache space is determined by the remaining power of the mobile node.

1) The Size and Contents of a Cooperative Cache

To realize the above idea, for any node, we let the size of its cooperative cache be proportional to its remaining power. Therefore, the size of the cooperative cache of node i, denote by $C_c(i)$, is given by

$$C_c(i) = \left\lfloor C * \frac{E_i}{E} \right\rfloor \quad (1)$$

where E_i is the current remaining power of node i, C and E are the total cache size and the initial battery power of node i, respectively. In Equation (1), because the cache size C and initial battery power E are constant, the size of the cooperative cache, $C_c(i)$, is directly proportional to its current remaining power E_i. When the energy of node i is almost empty, the ratio E_i/E approaches zero. On the other hand, E_i/E approach one when the remaining power of i is the same as the initial power.

For every data item in the cooperative cache, each of them meet the criterion which is used to evaluate whether the data item is the target to be cooperatively cached or not. This criterion is referred to as *profit*. The profit of node i caches data item j cooperatively is denoted by *profit(i,j)* and can be written as

$$profit(i, j) = P_{ij} + \sum_{i=1}^{k} P_{ij}(1 - \frac{R_{ij}}{R}) \quad (2)$$

where P_{ij} is the probability of node i to accesses data item j; P_{lj} is the probability a neighbor node l of node i to access data item j; k is the number of neighbors of node i; R is the transmission range of node i; R_{li} is the distance between node i, and node l. If R_{li} is greater than R, R_{li} is set to R. The term $\left(1 - R_{ij}/R\right)$ in Equation (2) is an indicator of the strength of the connection between node i and node l. If node i and node l are close to each other, their connection is strong. On the other hand, if node i and node l are far away from each other, their connection is weak. The value of the expression $\left(1 - R_{ij}/R\right)$ approaches one when node i and node l are at about the same location. On the other hand, the value of the expression $\left(1 - R_{ij}/R\right)$ approaches zero when node l is out of the communication range of node i.

In fact, the profit for a center node to cache a specified data item in its cooperative cache is the probability of the center node to access the data item plus the summation of the probabilities of all its neighbor nodes to access the data item multiply by their corresponding connection strength indicators. Since the purpose of cooperative caching is to benefit the center node and its neighbors, to cache the data items with high profit values is appropriate. This research therefore uses the profit values of the accessed data items to decide which data items should be cached in the cooperative cache. Based on the above description, the list of the cached data items in the cooperative cache of node i, denoted by $DC_c(i)$, can be represented as

$$DC_c(i) = Top[C_c(i)]\{profit(i,1), profit(i,2),..., profit(i,k)\} \quad (3)\#$$

That is, the data items to be cached in the cooperative cache of node i consist of the data items whose profit values are among the top $C_c(i)$ profit values of the set of $\{profit(i,1), profit(i,2), \ldots, profit(i,k)\}$.

2) The Size and Contents of a Local Cache

The idea of caching data on local cache is straightforward. In the local caching, a node caches the data items that only benefit its own data accesses. Notice that a node does not share its local cached data items with their neighbors. Since the size of the cooperative cache, $C_c(i)$, is determined in Equation (1), the size of the local cache of node i, $C_{nc}(i)$, is easily calculated as

$$C_{nc} = C - C_c(i) \quad (4)$$

After the size of the local cache of node i is calculated, the list of the local cached data items, denote by $DC_s(i)$, can be obtained as follows:

$$DC_s(i) = Top[C_{nc}(i)]AP_i \quad (5)$$

That is, the data items to be cached at the local cache of node i consist of the data items whose access probabilities are among the top $C_{nc}(i)$ access probabilities of access pattern AP_i.

3) Eliminating Overlapped Cooperative Cached Data Items

In performing cooperative caching, a data item may be cached in the center node and also in some of its neighbors. A situation that wastes cache memories. This research therefore introduces another function to determine whether a data item should be cached in the cooperative cache or not. Here, this function is referred to as *fitness*. There are two considerations when deciding where to store a cached data item. The first consideration is that the cached data items should be cached at the node that makes the largest profit value according to Equation (2). The second consideration is that the cached data item should be cached at the node with the most remaining battery power. To balance the two considerations, we define a fitness function for node i to cached data item j in its cooperative cache as follows:

$$fitness(i, j) = profit(i, j) \times \frac{E_i}{E}, \quad (6)\#$$

where E_i is the remaining power of node i, and E is the initial power of node i.

/*Initialization */
For each node j in *MANET* do
  broadcasts message $\{j, loc_j, E_j, AP_j\}$ to its neighbors; /* loc_j is the location of j*/
For each node i receiving message (j, loc_j, E_j, AP_j) from node j do {
  calculates $\{profit(i,w)|1\leq w \leq k\}$ and R_{ij}
  calculates $C_c(i)$ and $C_{nc}(i)$
  determines $DC_c(i)$ and $DC_s(i)$
  calculates $\{fitness(i,w)\ |w\in DC_c(i)\}$
  broadcasts $\{(w, fitness(i,w))|w \in DC_c(i)\}$
} /* End of initialization */
/* Replace overlapped cooperative cached data items */
When receives $\{(w, fitness(i,w)|\ w \in DC_c(i)\}$ Node j do {
  if $(\{DC_c(i) \cap DC_c(j)\} == \varnothing)$
    keeps track of $\{(w, fitness(i,w)|\ w \in DC_c(i)\}$;
  if $(\{DC_c(i) \cap DC_c(j)\}\ != \varnothing)$, for any data item w in $\{DC_c(i) \cap DC_c(j)\}$ do {
    if $(fitness(j,w) < fitness(i,w))$
    replaces *data item w* with the maximal_profit_data_item in $(\{D-\{DC_s(i) \cup DC_s(j)\}\})$;
  }
}

Figure 1 The energy conserving cooperative caching policy

Having determined the list $DCc(i)$ of the data items to be cached in its cooperative cache, node i, now plays in the role of a center node, calculates the fitness values of every data item in the list and broadcasts a message containing the identifiers of all the data items and their corresponding fitness values to its neighbors. When a neighbor node j of node i receives this message, it checks whether any data item in the message is also cached in its own cooperative cache. If the answer is no, node j keeps track of the message and records the data items that are stored in node *i's* cooperative cache. Otherwise, node j compares the fitness value of each overlapped data item, say w, in its cooperative cache and the fitness value of its counterpart in *i's* cooperative cache. If the fitness value of j on data item w is less than that of i on w, i.e., $fitness(j,w)< fitness(i,w)$, the duplicated data item w should be store in node i. Accordingly, node j replaces data item w in its cooperative cache with the maximal profit data item which is not yet stored in either i's or j's cooperative cache. The pseudo code of the cooperative caching policy is shown in Figure 1.

IV. Experimental Evaluation

In this section, we report the simulation results on the performance comparisons of the proposed policy with the other existing policies. In the following, we first give the parameters of the experiment. Then, we introduce the metrics for performance comparisons. Finally, we report the results of the simulation.

4.1 Parameters and Their Settings

The simulation model was implemented using NS2 on a PC running SUSE LINUX. The parameters and their setting are listed in table 1.

The access frequencies of a mobile node on the data items in MANET are modeled by Zipf-like distribution. The data items are numbered from 1 to k. The probability for data item l to be accessed by a mobile node is given by the following equation.

TABLE I.
Parameters and Their Settings

Descriptions	**Default Values**
Number of mobile nodes	50
Number of data sources	1
MAC	IEEE 802.11
Transmission bandwidth	2 M bits/sec
Communication range	250 meters
Data item size	2 K bytes
Number of data items	100
Cache space size	5,10,15,20,25 data items
Zipf-like theta (θ)	0.2 – 1.0
Simulation area of the MANET	1200×800 m^2
Routing protocol;	DSDV
Moving model	Random waypoint
Moving speed of a mobile node	6 m/sec
Energy of a mobile node	5 J
Mean time between queries	1 sec

$$P(k)=\begin{cases}\dfrac{\left(\frac{1}{k}\right)^{\theta}}{\sum_{i=1}^{n}\left(\frac{1}{i}\right)^{\theta}}, & 1\leq k\leq n,\\ 0, & otherwise.\end{cases} \tag{7}$$

In Equation (7), the degree of access skew is controlled by θ. A small value of θ represents a more uniform access pattern while a large value of θ represents a higher skewed data access pattern. In practice, $\theta > 0.85$ represents a high skewed access pattern while $\theta = 0$ represents a uniform access pattern. In addition, for the

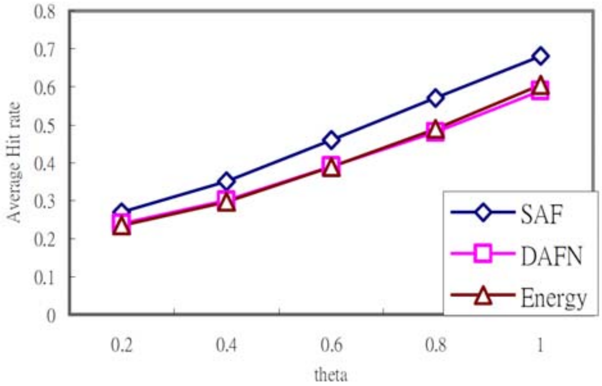

Figure 2. Effects of data skew on hit rate

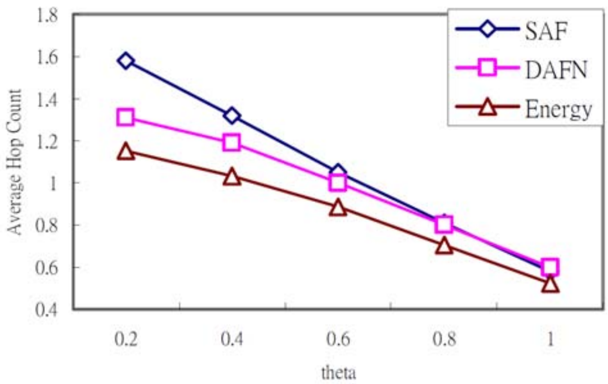

Figure 3. Effects of data skew on request delay

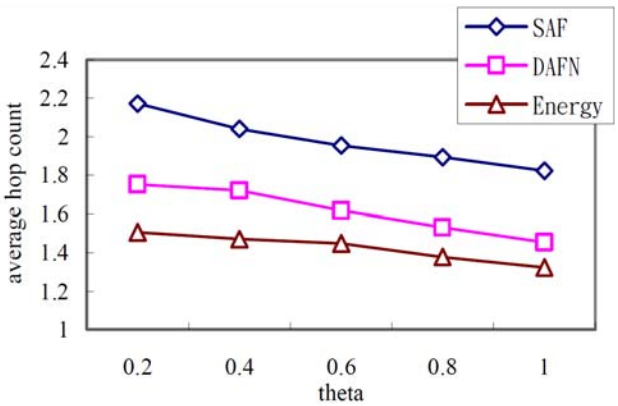

Figure 4. Effects of data skew on request delay of cache miss requests

energy consumption model in our experiments, we refer to the results of the paper in [16].

4.2 Performance Metrics

The metrics used in comparison are *hit rate*, *request delay* and *network lifetime*. The hit rate is the ratio of the number of the requested data items found in the local cache of a mobile node over the number of the data items queried by the mobile node. The request delay is the average hop counts from a mobile node that requests a data item to the mobile node that hosts the required data item. Finally, the network lifetime is defined to be the elapsed time since the starts operation of the MANET to the time when the first mobile node depletes its battery power [17].

4.3 Experimental Results

We compare the performance of the proposed policy, abbreviated to Energy, and the other two existing methods by extensive simulation experiments. The two methods to be compared are the static access frequency method [7], abbreviated to SAF, and the dynamic access frequency and neighborhood method [7], abbreviated to DFAN. The results of the experiments are reported in the following subsections.

1) Experiment 1: Effects of Data Skew on Average Hit Rate

In this experiment, we studied the effects of the data skew on hit rate for SAF, DAFN, and Energy. In the experiments theta is varied from 0.2 to 1 with an increment of 0.2. The result is depicted in Figure 2. It is shown that the hit rate increases as theta increases for all the policies. Note that as *theta* increases, a mobile node tends to query a small number of data items called hot data items. As the hot data items are queried frequently, they are easily cached in the cache space. Furthermore, as less number of data items is queried, the cache size also tends to be abundant for data caching. These effects result in high hit ratios for all policies. Also note that the hit rate of SAF is greater than those of DAFN and Energy. The reason is that SAF only performs local caching while both DAFN and Energy perform local caching and cooperative caching.

2) Experiment 2: Effects of Data Skew on Request Delay

Figure 3 shows the average hop counts for different levels of data skew. Similar to the reasons for the result of experiment 1, high *theta* values make the queried data items to be cached in the local cache of the querying mobile node or in the cooperative caches of the neighbors of the querying mobile node. This helps to reduce the average hop counts. Note that while the Energy policy has low hit rate compared to SAF and DFAN, it has the lowest hop counts among the three policies. This is because with the Energy policy a mobile node accesses a requested data item from the cooperative caches of its neighbors when a cache miss occurs. Since the neighbors are just one-hop away from the mobile node, it incurs little cost to access the cached data item from the cooperative caches. This helps to reduce the hop counts for queries with the Energy policy.

The low average hop count of the Energy policy can be better explained by examining the hop counts for a data item that is not in the local cache of the querying mobile node. Figure 4 shows the hop counts required for a data item that is not in the local cache. It shows that both the Energy and the DAFN have low hop counts for data items that incur cache miss. This is because the cooperative caching policy helps to bring the cache-missed data items to the nodes that are close to the querying mobile node.

3) Experiment 3: Effects of Data Skew on Data Accessibility

In this experiment, we varied the cache size from 5% to 25% with an increment of 5% to study the effects of cache size on the request delay. Here, 5% means that the

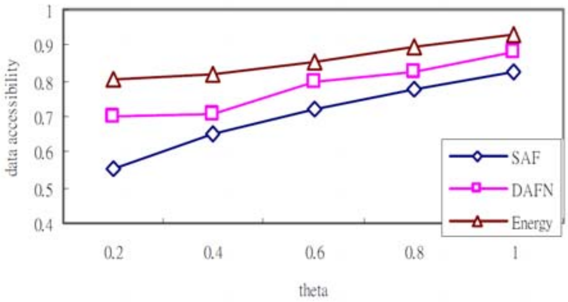

Figure 5 Effects of data skew on data accessibility

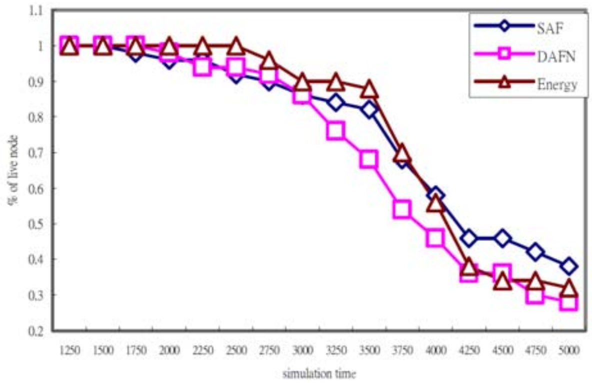

Figure 6 Simulation time and live node rate

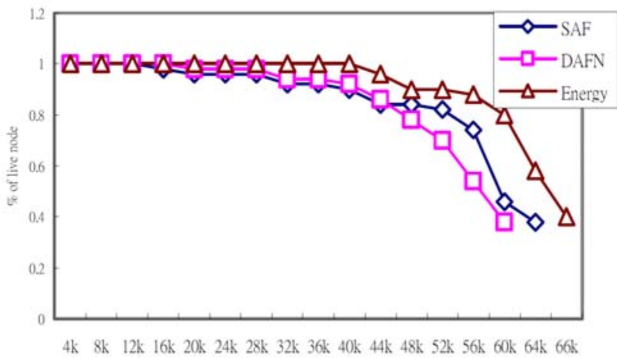

Figure 7 Number of completed requests and live node rate

cache size is set to be 5% of the number of all data items. The experiment result is shown in Figure 5. Figure 5 shows that the average hop counts decreases as the cache size increases for all the three policies. It shows that the Energy policy has the lowest average hop count among the three policies.

4) Experiment 4: Effects of Cache Size on Request Delay

In this experiment, this research studied the effects of the cache size on average hop count among SAF, DAFN, and Energy. In the experiments we varied the cache size from 5% to 25% by steps of 5% to study the effects of cache size on request delay. Here, 5% means the cache size was set to 5% of the number of data items. The experiment result was given in Figure 6. In this figure, the x-axis represents the cache size while the y-axis represents the data accessibility, that is, the average hop count. The result show that the data accessibility decreased as the cache size increased among all evaluated methods. In addition, Energy had the lowest average hop count, DAFN had the middle, SAF had the highest.

5) Experiment 5: Simulation Time and Live Node Rate

In this experiment, we study the live node rate over the simulation times. The total simulation time of the experiment is 5000 time unit and the live node rates are reported at simulation time from 1250 to 5000 with an increment of 250. In this experiment, we call a node that depletes its power a dead node. Figure 6 shows the percentages of the live nodes for the three policies over the execution times. It shows that the first node to be dead for the Energy policy occurs at simulation time 2500, which by far later than those of the DFAN (at simulation time 1750) and SAF (at simulation time 1500). Based on this result, we conclude that the Energy policy has the longest lifetime of the MANET among the three policies. This is because only the Energy policy considers the remaining power of a mobile node when determining the size of the cooperative cache. While the power of a node is low, the node reduces the size of the cooperative cache to decrease the energy consumption rate. In this way the mobile nodes evenly consume their energy in performing cooperative caching so as to prolong the lifetime of the network. Another fact to be noticed is that the live node rates of the Energy policy start to decrease dramatically when the simulation time reaches 3500. This is because in the time interval between 3500 and 5000 the Energy policy finishes more requests than the DFAN and SAF. This makes the Energy policy to spend more energy than the DFAN and SAF. The next experiment gives more details on relationship of the number of completed requests and the energy consumption.

6) Experiment 6: Number of Completed Requests and Live Node Rate

In the experiment, we study the relationship of the number of completed requests and the live node rate of different policies. This experiment result is given in Figure 7. In this figure, the x-axis represents the number of completed requests while the y-axis represents the live node rates. The result shows that the Energy policy has the highest live node rates compared to the other two policies at the same number of completed requests.

V. Conclusion

Unlike the existing cooperative caching methods which focus on reducing query delay in MANETs, we present a policy that focus on both reducing query delay and energy conservation. The proposed policy not only reduces the query delay but also enforces uniformly consumption of energy among the mobile nodes. Comparing with the existing methods, the simulation results showed that the proposed policy not only reduces the query delay but also extends the lifetime of the network.

Acknowledgment

This work was supported by the National Science Council of Taiwan grant NSC 101-2221-E-011-135.

REFERENCES

[1] T. G. Subir Kumar Sarkar and C. Puttamadappa Basavaraju, *Ad hoc mobile wireless networks*: *principles, protocols, and applications*, Auerbach Publications, 2008.

[2] L. Abusalah, A. Khokhar, M. Guizani, “A survey of secure mobile ad hoc routing protocols,” *IEEE Communications Surveys* & *Tutorials*, Volume 10, Issue 4, pp. 78 – 93, 2008.

[3] L. Hanzo II, R. Tafazolli, “Admission control schemes for 802.11-based multi-hop mobile ad hoc networks: a survey,” *IEEE Communications Surveys* & *Tutorials*, Volume 11, Issue 4, pp. 78 – 108, 2009.

[4] Shushan Zhao, A. Aggarwal, R. Frost, Xiaole Bai, “A survey of applications of identity-based cryptography in mobile ad-hoc networks,” *IEEE Communications Surveys* & *Tutorials*, Volume 14, Issue 2, 2012.

[5] A. Derhab and N. Badache, "Data replication protocols for mobile ad-hoc networks: a survey and taxonomy," *IEEE Communications Surveys* & *Tutorials*, Vol.11, No.2, pp.33-51, 2009.

[6] M. K. Denko, Jun Tian, T. Nkwe, M.S.Obaidat, “Cluster-based cross-layer design for cooperative caching in mobile ad hoc networks,” *IEEE Systems Journal*, Volume 3 , Issue 4, pp. 499 – 508, 2009.

[7] T. Hara, and S. K Madria, "Data Replication for Improving Data Accessibility in Ad Hoc Networks," *IEEE Trans. on Mobile Computing*, Vol. 5, No. 11, pp. 1515-1532, 2006.

[8] L. Yin and G. Cao, “Supporting cooperative caching in ad hoc networks,” *IEEE Transaction on Mobile Computing*, vol. 5, no. 1, pp. 77–89, Jan. 2006.

[9] 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.

[10] T. Hara, Y.-H. Loh and S. Nishio, ”Data replication methods based on the stability of radio links in ad hoc networks,” in *Proceedings of the 14th International Workshop on Database and Expert Systems Applications*, pp. 969 - 973, Sept. 2003.

[11] L. Yin and G. Cao, “Balancing the Tradeoffs between Data Accessibility and Query Delay in Ad Hoc Networks,” in *Proceeding of IEEE symposium on reliable distributed systems* (SRDS), 2004.

[12] Takahiro Hara, “Energy efficient data access in mobile ad hoc networks,” in *proceedings of the Eleventh International Conference on Mobile Data Management* (MDM), pp. 326 – 330, 23-26 May, 2010.

[13] M. Pushpalatha, Revathi Venkataraman and T. Ramarao, “Energy aware data sharing in mobile ad hoc networks,” *Communications in Computer and Information Science*, Volume 90, Part 2, pp. 378-387, 2010.

[14] P. Mukilan and A. Wahi, “EENMDRA: efficient energy and node mobility based data replication algorithm for MANET,” *International Journal of Computer Science*, Issues, Vol. 9, Issue 3, No 1, May 2012.

[15] E. Chan, W. Li and D. Chen, “Energy saving strategies for cooperative cache replacement in mobile ad hoc networks,” *Pervasive Mobile Computing*, 5(1):77–92, 2009.

[16] 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.

[17] 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.

Yungho Leu received his B.S. and M.S. degrees in Electrical Engineering in 1981 and 1983, respectively, from National Taiwan University, Taiwan. He received his Ph.D. degree in Computer Science in 1991 from Purdue University, West Lafayette, IN. He is now a professor of the Department of Information Management, National Taiwan University of Science and Technology. He is interested in mobile computing including VANETs, MANETs and Sensor Networks. He is also interested in data mining including Fuzzy Sets, Fuzzy Time Series, Genetic Algorithms, Association Rules, Bioinformatics and Computational Intelligence in Financial Engineering. He has published many papers in the referred journals and conferences in the above areas. He is a member of the IEEE.

Hsin-Chang Lin received his B.S. degree in Applied Mathematics from Chung Yuan Christian University, Taoyuan, Taiwan, in 2003. He received his MBA degree from National Taiwan University of Science and Technology, Taipei, Taiwan in 2005. He is interested in Mobile Computing.

Chi-Chung Lee was born in Hsinchu, Taiwan. He received his B.S. degrees in Industrial Engineering from Chung Yuan Christian University, Taoyuan, Taiwan, in 1991. He received his MBA and Ph.D. degree in Information Management from National Taiwan University of Science and Technology, Taipei, Taiwan, in 1993 and 2004 respectively. His research interests include database systems, data management for mobile computing, and RFID systems. He is currently an assistant professor of the Department of Information Management, Chung Hwa University, Hsinchu, Taiwan.