Performance Evaluation Of CLIR, LDIS And LDCC Cooperative Caching Schemes Based On Heuristic Algorithms

R. K. Ravindra Raju[1], B. Santha Kumar[2], Nagaraju Mamillapally[3]

[1]M.Tech(CSE), Sri Kottam Tulasi Reddy Memorial College of Engineering, Kondair, India.
[2]Asst.Professor, Sri Kottam Tulasi Reddy Memorial College of Engineering, Kondair, India,
[3]Asst.Professor, Adarsh PG College of Computer Sciences, Mahabubnagar,

ABSTRACT

Location Dependent Information Services (LDISs), through which mobile clients can access location sensitive data such as weather information, traffic reports, and local news, are gaining increasing popularity in recent years.

Simulations show that the proposed policies can significantly reduce energy consumption and access latency when compared to other replacement policies. In this paper, we studied the proposal of a cache replacement policy called Location Dependent Cooperative Caching (LDCC) and also discussed the issues of developing energy-efficient coordinated cache replacement policies in mobile ad hoc networks.

Keywords:LDIS, Simulation, Sensitive Data, LDCC, Ad Hoc

1. INTRODUCTION

We address cooperative caching in wireless networks, where the nodes may be mobile and exchange information in a peer-to-peer fashion. We consider both cases of nodes with large and small-sized caches. For large-sized caches, we devise a strategy where nodes, independent of each other, decide whether to cache some content and for how long. In the case of small-sized caches, we aim to design a content replacement strategy that allows nodes to successfully store newly received information while maintaining the good performance of the content distribution system. Under both conditions, each node takes decisions according to its perception of what nearby users may store in their caches and with the aim of differentiating its own cache content from the other nodes'. The result is the creation of content diversity within the nodes neighborhood so that a requesting user likely finds the desired information nearby. We simulate our caching algorithms in different ad hoc network scenarios and compare them with other caching schemes.

We evaluate the performance of the CLIR (Cross-Layer Interception and Redirection) cooperative caching scheme for ad hoc networks. Although this caching scheme was developed for Mobile Ad Hoc Networks (MANET) we

study the application of this kind of algorithms in static grid ad hoc networks. By means of simulations, we evaluate the mean traffic generated in the wireless network, the delay perceived by the users and the percentage of failed searches as a function of the mean time between requests, the Time

To Live (TTL) of the documents, the traffic pattern and the cache sizes. We compare the performance cooperative caching schemes as well as the option of no using a caching scheme.

Location Dependent Information Services (LDISs), through which mobile clients can access location sensitive data such as weather information, traffic reports, and local news, are gaining increasing popularity in recent years. Due to limited client power and intermittent connectivity, caching is an important approach to improve the performance of LDISs. In this paper, we propose a cache replacement policy called Location Dependent Cooperative Caching (LDCC). Unlike existing location dependent cache replacement policies, the LDCC strategy applies a prediction model to approximate client movement behavior and a probabilistic transition model to analyze the communication cost. These models are used in the design of a cache replacement policy to improve system overall performance. Simulation results demonstrate that the proposed strategy significantly outperforms existing caching policies in providing LDIS in mobile ad hoc networks.

Data caching on mobile clients is widely seen as an effective solution to improve system performance. In particular, cooperative caching, based on the idea of sharing and coordination of cache data among multiple users, can be particularly effective for information access in mobile ad hoc networks where mobile clients moving frequently and network topology changing dynamically. Most existing cache strategies perform replacement independently, and they seldom consider coordinated replacement and energy saving issues in the context of a mobile ad hoc network. In this paper, we analyze the impact of energy on designing a cache replacement policy and formulate the Energy-efficient Coordinated cache Replacement Problem (ECORP) as a 0-1 knapsack problem. A heuristic algorithm called ECORP-Greedy and an optimal solution called ECOR-POPT are presented to solve the problem. Simulations show that the proposed policies can significantly reduce energy consumption and access latency when compared to other replacement policies.

2. RELATED WORK

Providing information to users on the move is one of the most promising directions of the infotainment business. The ubiquity and ease of access of third- and fourth-generation (3G or 4G) networks will encourage users to constantly look for content that matches their interests. However, by exclusively relying on downloading from the infrastructure, novel applications such as mobile multimedia are likely to overload the wireless network. It is thus conceivable that a peer-to-peer system could come in handy, if used in conjunction with cellular networks, to promote content sharing using ad hoc networking among mobile users. For highly popular content, peer-to-peer distribution can, indeed, remove bottlenecks by pushing the distribution from the core to the edge of the network.

In such an environment, however, a cache-all-you-see approach is unfeasible, because it would swamp node storage capacity with needless data that were picked up on the go. Thus, several techniques of efficiently caching information in wireless ad hoc networks have been investigated in the literature. The solution that we propose, called Hamlet, aims at creating content diversity within the node neighborhood so that users likely find a copy of the different information items nearby (regardless of the content popularity level) and avoid flooding the network with query messages.

The cooperative caching schemes for ad hoc networks can be classified into four groups: broadcast-based, information - based, role-based and direct-request. The broadcast-based caching schemes employ broadcast messages as the first choice in order to find the documents in the network. These broadcast messages can be sent to the entire network, as in the case of MobEye. Other schemes such as SimpleSearch [4], follow a more restrictive approach that limits the distance of the messages to four hops. ModifiedSS is an evolution of SimpleSearch that employs GPS (Global Positioning System) in order to send the requests to the direction where the servers are located. Similarly, the caching scheme proposed by Moriya in sends the broadcast messages to the neighborhood so that, if the document is not found, the request is transmitted to the server.

The information-based cooperative caching schemes employ information of the location of the documents in the network. Nodes obtain this information by analyzing the messages that they forward. As examples of this category of caching schemes we can mention: DGA (Distributed Greedy Algorithm), Wang, Cho and POACH (POware Aware Caching Heuristic).

Under a role-based caching scheme, each node in the wireless network has a predefined role. That is, they can be caching nodes, requesting nodes, coordinator nodes, gateway nodes, etc. The role-based caching schemes are usually applied to cluster networks.

Finally, the direct-request caching schemes directly send the requests to the server with the hope of being served by an intermediate node in the route from the requester to the server.

Location Dependent Information Services (LDISs), by producing the answer to a query according to the geographical location from where the query originates, are becoming increasingly popular. Traditional location dependent applications, such as weather report and directory services, have brought many benefits in our daily lives.

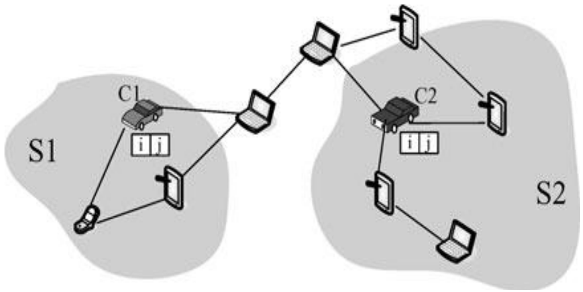


Figure 1: Cache replacement in a mobile environment

With the increasing popularity of mobile and pervasive computing, plenty of new applications, just like traffic reports and vehicle navigation, show great potential in mobile environments where users can enjoy unrestricted mobility and ubiquitous information access. However, inherent constraints in the realm of mobile computing such as low bandwidth and limited battery life must be addressed before LDIS can be deployed on a large scale. Effective cache management has the potential to deal with many of these issues. Caching allows users to fetch data from a nearby cache rather than from a distant server, and can bring the following benefits:

(1) Saving the energy of mobile equipments
(2) Reducing the access latency;
(3) Reducing the bandwidth consumption and the workload of the server;
(4) Enhancing the robustness of the server.

One of the main design issues in cache management, particularly in mobile ad hoc networks (MANET), is how to formulate a suitable cache replacement strategy to handle mobility and resource constraints. An efficient cache replacement scheme for LDISs needs to consider a lot of impact factors, including access probability, update frequency, object sizes, and location correlated factors. In order to use location information for cache replacement decision, most existing policies consider three location properties: the direction of mobile clients, the data distance, and the data valid scope. However, their concern involved only a few simple aspects of movement. Other movement characteristics like the probability that a mobile node moves into/out a valid scope also produces a great influence on the efficiency of cache strategies, which has not been paid enough attention in the literature.

The recent years have witnessed a rapid development of mobile computing and wireless communication techniques. However, constraints such as scarce bandwidth, frequent network disconnections, and limited local resources (energy and computing power) must still be overcome to enable efficient information access in wireless network environments. Data caching on mobile clients has been shown to be effective in improving system performance, i.e. terminal latency, server workload, connectivity, bandwidth and energy consumption. However, caching on individual systems cannot be scaled easily. Cooperative caching, based on the idea of sharing and coordination of cache data among multiple users, has emerged as an approach to overcome the limitation. There are several important issues involved in cache management: cache replacement, cache prefetching and cache invalidation. In this paper we study the cache replacement problem. Most existing cache replacement strategies perform replacement independently. Such a local replacement strategies are not very efficient in the cooperative caching, because terminals cannot make full use of each other’s caching space. For example, consider a region where mobile terminals have the same or similar access characters, if each terminal performs cache replacement independently, it’s very likely that the same data items are stored in each terminals. Forwarding requests to the neighbor terminals that hold the same data items is inefficient. As a result, coordinated replacement strategies are more effective in such a condition. We address the issues of developing energy-efficient coordinated cache replacement policies in mobile ad hoc networks. The contributions of our papers are as follows:

(1) As far as we know this is the first cache replacement policy based on coordinated replacement designed specifically with energy efficiency as the main performance objective;
(2)We analyze the impact of energy on cache replacement, and pose it as a 0-1 knapsack problem;
(3) A heuristic algorithm and an optimal algorithm using dynamic programming are presented to solve the energy efficient cache replacement problem, and the algorithms are shown to outperform existing approaches.

3. PROPOSED CACHING SCHEMES AND REPLACEMENT POLICIES

This section describes some of the caching schemes and replacement policies for data retrieval.

CLIR implements a local cache in every node in the network. This local cache is managed using the LRU (Least Recently Used) replacement policy. Using this cache, every node stores the received documents. Therefore, further requests to the same document will be resolved by the local cache. This is called a local cache hit. As the requests must be forwarded hop by hop from the requester node to the server node, the intermediate nodes in the route from the source to the destination of the requests can reply directly if the requested document is stored in their local cache. This is called an interception cache hit.

When the route from the source node of the request to the destination node has not been created, CLIR utilizes the routing protocol to piggy-back the request in the routing protocol messages. By using this technique, the routing protocol is able to create the route to the destination node and search for the requested document at the same time. If any node that receives the route request message has a copy of the requested document in its local cache, it will reply using the route reply message informing that this node has a copy of the document. When the requester node receives the route reply message, the route between both nodes is created and the requester node will forward the request to the node that has the copy of the document. This is called a cross-layer interception hit. This mechanism allows finding the documents in the network even if the server is not temporarily available.

CLIR also implements a redirection cache that stores information about where the documents are located in the network. This information is obtained from the messages that are forwarded by the mobile node. The redirection cache manages information about the source of the requests and the corresponding replies. It also stores the number of hops and the TTL of the documents and it estimates the time that the documents are stored in the local caches. The redirection cache is managed by means of two LRU lists, one for the documents whose TTL is known and the other with the documents with an unknown TTL. When a node receives a request and the redirection cache contains information of a node that is closer to the original destination of the request, the request is forwarded to this closer node. When the redirected node receives the request, it replies with the document. This is called a redirection cache hit. In the case that the redirected node has evicted the document from its local cache, a redirection error message is sent to the redirection node in order to update the information of the redirection cache.

Finally, CLIR also implements the storage of the replied document in the node located in the middle of the route from the source and destination of the reply. So, the documents can be easily disseminated along the network. In order to avoid the excessive replication of documents, this mechanism is performed if the distance between both nodes is greater than four hops.

We first describe our cooperative caching model and then formulate our location dependent caching scheme. In our cooperative caching model, when a mobile client generates a request, it first checks its local cache. If a valid data item value has been cached, the client will get the value from the local cache, called a *local hit* and stop the search. Otherwise, the client will broadcast a request to all the mobile clients in its communication range. If one of these neighbors has a valid value, it will send the answer back; this is called a *range hit* or *remote cache hit*. If none of the neighbors has the answer, the client will send the request to the server to retrieve the valid value (a *server hit*). In this paper, we assume that each answer is attached with complete/partial invalidation information. We now describe our proposed policy, called LDCC (Location Dependent Cooperative Caching). In our policy, each client performs its cache replacement using its local information. We use the below to get the approximate time sequence {0, . . . ,*AT*out*jk*− 1 ,*AT*in*jk* , *AT*out*jk* , *AT*in*jk*+1, . . . , *fw*}. So, the cost function of data item *di,j* is as follows:

$$cost_{i,j} = p_i((\varepsilon(sup)+\varepsilon(s_i))P^{V}_{i,j} - (\varepsilon(sup)+\varepsilon(sdw))P^{PV}_{i,j})$$

$$= \frac{\lambda_i^a p_i}{\lambda_i^a+\lambda_i^u}\Bigg(\varepsilon(sup)+\varepsilon(s_i)$$

$$- \frac{(\varepsilon(sup)+\varepsilon(sdw))\sum_k (AT_k^{\mathrm{out}_j} - AT_k^{\mathrm{in}_j})}{fw}\Bigg)$$

$$\times \sum_k e^{-\lambda_i^u AT_k^{\mathrm{in}_j}}\left(e^{-(\lambda_i^a+\lambda_i^u)AT_k^{\mathrm{in}_j}} - e^{-(\lambda_i^a+\lambda_i^u)AT_k^{\mathrm{out}_j}}\right)$$

Now the cache replacement problem becomes an optimization problem: how to choose a set of objects *V* to be replaced by the new object *dk* to maximize the retrieve cost saving, or in other words, to minimize the value of $_{i\in\ V}$*costi*. has shown that such problem can be mapped to a 0-1 knapsack problem, which is known to be NP-hard.

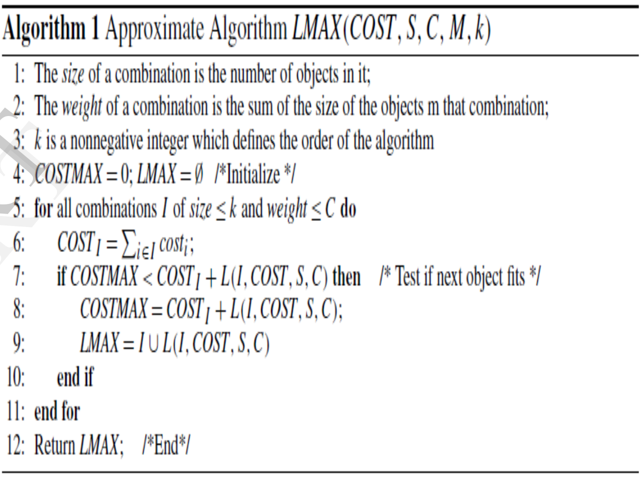
The problem can be formulated as follows.

Notations:

di is the data item value cached in Client N0, i = 1, . . . , M;
costi is the cost value of di , i = 1, . . . , M;
si is the size of di , i = 1, . . . , M;
C is the capacity of Client N0;
The problem is to

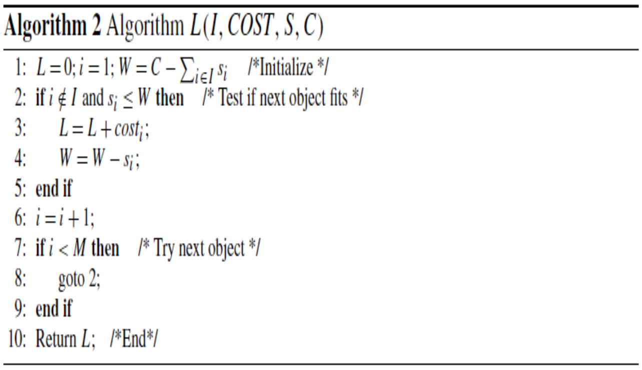
$$\text{maximize} \qquad \sum cost_i\,\delta_i$$

$$\text{subject to} \quad \sum s_i\delta_i \le C \text{ and } \delta_i \in \{0,1\}$$



Algorithm 1 Approximate Algorithm $LMAX(COST, S, C, M, k)$

```
The size of a combination is the number of objects in it;
The weight of a combination is the sum of the size of the objects m that combination;
$k$ is a nonnegative integer which defines the order of the algorithm
$COSTMAX = 0; LMAX = \emptyset$   /*Initialize */
for all combinations $I$ of size $\le k$ and weight $\le C$ do
    $COST_I = \sum_{i\in I} cost_i$;
    if $COSTMAX < COST_I + L(I, COST, S, C)$ then   /* Test if next object fits */
        $COSTMAX = COST_I + L(I, COST, S, C)$;
        $LMAX = I \cup L(I, COST, S, C)$
    end if
end for
Return $LMAX$;   /*End*/
```



Algorithm 2 Algorithm $L(I, COST, S, C)$

```
1: $L = 0; i = 1; W = C - \sum_{i\in I} s_i$   /*Initialize */
2: if $i \notin I$ and $s_i \le W$ then   /* Test if next object fits */
3:     $L = L + cost_i$;
4:     $W = W - s_i$;
5: end if
6: $i = i + 1$;
7: if $i < M$ then   /* Try next object */
8:     goto 2;
9: end if
10: Return $L$;   /*End*/
```

THE ECORP-GREEDY AND ECORP-OPT ALGORITHMS

A. ECORP-Greedy Algorithm

Suppose the objects in C are ordered according to decreasing value of the benefit-to-size ratio, so that

$$benefit(c_1)size_{c_1} \geq benefit(c_2)size_{c_2} \geq \cdots \geq benefit(c_m)\ size_{c_m}.$$

We define the *gain* of caching an object c_i as

$$gain(c_i) = benefit(c_i)\ size_{c_i}.$$

The ECORP-Greedy strategy can be described as follows. In each step the object with the smallest *gain* value is chosen as a victim, until the available space is enough for caching the new object O_k.

B. ECORP-OPT Algorithm

The ECORP problem can be solved in pseudo-polynomial time using dynamic programming. Given a pair of integer $\mu(1 \leq \mu \leq m)$ and $v(0 \leq v \leq S - size_{O_k})$, consider the sub instance of ECORP consisting of objects $O_1, \cdots, O_\mu$ and capacity v. Let $g(\mu, v)$ denote its optimal solution value, i.e.

$$g(\mu,\nu) = \\ max\left\{\sum_{j=1}^{\mu} benefit(c_j)x_j : \sum_{j=1}^{\mu} size_{c_j}x_j \leq \nu, x_j \epsilon \{0,1\}\right\} \\ for\ j = 1, \cdots, \mu$$

We have

$$g(1,\nu) = \begin{cases} 0, for\ \nu = 0, \cdots, size_{c_1} - 1; \\ benefit(c_1), for \nu = size_{c1}, \cdots, S - size_{O_k} \end{cases}$$

$$g(\mu,\nu) = \\ \begin{cases} g(\mu - 1, \nu), \\ \qquad for\ \nu = 0, \cdots, size_{c_m} - 1 \\ max(g(\mu - 1, \nu), g(\mu - 1, \nu - size_{c_m}) + benefit(c_m)), \\ \qquad for\ \nu = size_{c_m}, \cdots, S - size_{O_k}. \end{cases}$$

The optimal solution of the problem is $g(m, S - size_{O_k})$.

4. PERFORMANCE EVALUATION

In this section, we evaluate the performance of CC caching through simulation experiments and their evaluation impacts.

- Simulation Model
    - Client Model
    - Server Model
- Time between requests
- TTL of the documents
- Zipf slope
- Cache Size
- The Movement Prediction Model
- Analytical Model for Location Dependent Caching
- System Execution Model
- Mobile Client Model
- Evaluation Metrics
- Impact of Movement Interval
- Impact of Query Interval
- Impact of Update Interval
- Impact of Forecast windows
- Comparison
- Impact of Node Density

5. CONCLUSION

This paper exploits clustering for efficient data caching in mobile ad hoc networks. The proposed CC caching scheme is scalable and incurs lower overhead with increasing number of nodes. we have evaluated the performance of the CLIR caching scheme applied to static grid ad hoc networks. This evaluation has been performed using the metrics: mean traffic processed by the node, the delay perceived to obtain the requested documents and the percentage of mean timeouts. We have evaluated the influence of the traffic load in the network, the TTL of the documents, the traffic pattern (Zipf slope) and the local cache size. we propose a model to predict the relative movement of mobile clients to valid scopes, and we also provide a state transition model to analyze the communication cost in the location dependent information services. Cooperative caching is an effective approach for supporting efficient information access in mobile ad hoc networks. An important element of cooperative caching is the cache replacement scheme.

ACKNOWLEDGEMENT

We want to take this opportunity to thank all the people who helped me during my conference sojourn. I understand that it is rather late to acknowledge their contributions, but as the saying goes, better late than never!

REFERENCES

[1]. J. Wortham (2009, Sep.). **Customers Angered as iPhones Overload AT&T.** The New York Times. [Online]. Available:http://www.nytimes.com/2009/09/03/technology/companies/03att.html

[2]. A. Lindgren and P. Hui, **"The quest for a killer app for opportunistic and delay-tolerant networks,"** in Proc. ACM CHANTS, 2009, pp. 59–66.

[3]. P. Padmanabhan, L. Gruenwald, A. Vallur, and M. Atiquzzaman, **"A survey of data replication techniques for mobile ad hoc network databases,"** VLDB J., vol. 17, no. 5, pp. 1143–1164, Aug. 2008.

[4]. A. Derhab and N. Badache, **"Data replication protocols for mobile ad hoc networks: A survey and taxonomy,"** IEEE Commun. Surveys Tuts., vol. 11, no. 2, pp. 33–51, Second Quarter, 2009.

[5]. G. Dodero and V. Gianuzzi, **"Saving Energy and Reducing Latency in MANET File Access"**, Proc. 26th International Conference on Distributed Computing Systems Workshops (ICDCSW'06), 2006, pp. 16-20.

[6]. S. Lim, W.C. Lee, G. Cao and C.R. Das**, "A novel caching scheme for improving Internet-based mobile ad hoc networks performance",** Ad Hoc Networks, vol. 4, no. 2, 2006, pp. 225-239.

[7]. S. Lim, W.C. Lee, G. Cao and C.R. Das, **"Cache invalidation strategies for Internet-based mobile ad hoc networks"**, Computer Communications, vol. 30, no. 8, 2007, pp. 1854-1869.

[8]. T. Moriya and H. Aida, **"Cache Data Access System in Ad Hoc Networks"**, Proc. 57th IEEE Semiannual Vehicular Technology Conference (VTC 2003), April 2006, vol. 2, pp. 1228-1232.

[9]. B. Tang, H. Gupta and S.R. Das, **"Benefit-Based Data Caching in Ad Hoc Networks"**, IEEE Transactions on Mobile Computing, vol. 7, no. 3, 2008, pp. 289-304.

[10]. Y.H. Wang, J. Chen, C.F. Chao and C.C. Chuang, **"A Distributed Data Caching Framework for Mobile Ad Hoc Networks"**, Proc. 2006 International conference on Wireless communications and mobile computing, 2006, pp. 1357-1362.

[11]. J. Cho, S. Oh, J. Kim, K.H. Lee and J. Lee, **"Neighbor Caching in Multi-Hop Wireless Ad Hoc Networks",** IEEE Communications Letters, vol. 7, no. 11, 2003, pp. 525-527.

[12]. P. Nuggehalli, V. Srinivasan and C.F. Chiasserini, **"Energy-Efficient Caching Strategies in Ad Hoc Wireless Networks",** Proc. 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003), June 2003, pp. 25-34.

[13]. N. Chand, R.C. Joshi and M. Misra**, "Cooperative Caching in Mobile Ad Hoc Networks Based on Clusters",** International Journal on Wireless Personal Communications, no. 43, 2007, pp. 41-63.

[14]. M.K. Denko**, "Cooperative Data Caching and Prefetching in Wireless Ad Hoc Networks",** International Journal of Business Data Communications and Networking, vol. 3, no. 1, 2007, pp. 1-15.

[15].**"Movement prediction based cooperative caching for location dependent information service in mobile ad hoc networks"**, Edward Chan · YilinWang ·Wenzhong Li Sanglu Lu.

[16].Wenzhong Li, Edward Chan, Daoxu Chen – **"Energy- efficient Cache Replacement Policies for Cooperative Caching in Mobile Ad Hoc Network"**, WCNC 2007 Proceedings.