ISSN:2249-5789

# A ROBUST AND EFFICIENT DATA TRANSMISSION IN ADHOC NETWORKS

[1]V.Narayanarao, M.tech student, Aitam college, Tekkali, India
[2]Promod kumar sahu, Associate professor, Aitam college, Tekkali, India
[1]narayanarao.mtech@gmail.com
[2]promod_sahu@yahoo.com

## Abstract

*Recently lot of interest has gained in ad hoc networks which are characterized as networks without any fixed topology due to high mobility. In ad hoc networks due to high mobility and lack of fixed infrastructure network disconnections occur frequently. Hence data accessibility in ad hoc networks is lower than in the fixed infrastructure networks. In this paper, we address cooperative caching mechanism to improve data accessibility in adhoc networks. Proposed solution is based on the size of the cache in nodes. For large-sized caches, nodes take decision independent of each other whether to cache some data and how long. In small-sized caches, a data replacement strategy allows nodes to store newly received data while maintaining the better performance in the data distribution system. Performance of the proposed caching algorithms is evaluated by simulating in NS2 under various ad hoc network scenarios. Simulation results reveal that, proposed solution successfully created data diversity in the adhoc networks which leads to an efficient data access.*

## 1. Introduction

In ad hoc networks, nodes communicate each other using multi-hop wireless links. Due to high mobility and lack of fixed infrastructure, each node acts as a router and forwards data to other nodes. Even if the source and the destination nodes are not in communication range to each other, data will be transferred to the destination node by forwarding through intermediate nodes. Since no special infrastructure is required, in various fields such as military affairs and commerce, many applications are expected to be developed in ad hoc networks.

In such applications, general caching strategies is not feasible, because it would swamp node storage capacity with needless data that were picked up on the go. Several techniques have been proposed by various authors for efficient data caching in ad hoc networks which is discussed in the literature survey [1] [2]. In this paper, we address cooperative caching mechanism to improve data accessibility in adhoc networks. Our proposed algorithm, aims at creating data diversity within the neighbor nodes so that users likely find a copy of the different data items nearby and avoid flooding the network with enquiry messages. A similar concept has presented in [3]–[6], the novelty in our proposal resides in the probabilistic estimate, run by each node, of the information presence in the node proximity. The estimate is evaluated in a cross-layer fashion by overhearing query and data reply messages due to the broadcast nature in the wireless channel. By leveraging such a local estimate, nodes autonomously decide what information to keep and for how long, resulting in a distributed scheme that does not require additional control messages.

We proposed a solution based on the size of the cache in nodes. For Large-sized caches, nodes cam store a large amount of the available data items. Reduced memory usage is a desirable condition, because the same memory may be shared by different services and applications which may run at various nodes. In such scenario, a caching decision consists of computing for how long a data should be stored in a node that has requested previously to minimize the memory usage without affecting the overall performance. In Small-sized caches, each node has a dedicated but limited amount of memory to store a small amount of the data that they retrieve. In this case, caching decision translates into a cache replacement strategy that selects the data items to be replaced among the items just received and the items that already available in the memory.

The remainder of this paper is organized as follows. In section 2 discuss about related work, section 3 describes about the proposed solution, section 4 explains the experimental setup and section 5 concludes the paper.

## 2. Related Work

Several algorithms have been proposed by various authors to address the data caching and replacement mechanism in adhoc networks. In this section, we discuss about those solutions which are related to this paper and highlighting the differences with respect to proposed solution.

### *A. Cooperative Caching*

In [7], author has proposed distributed caching strategies for ad hoc networks according to it nodes may cache highly popular data that passes by or record the data path and use it to redirect future requests. Among the schemes presented in [7], the approach called HybridCache best matches the operation and system assumptions that we consider; we thus employ it as a benchmark for evaluating the proposed solution. In [8], a cooperative caching technique is presented it provides better performance than HybridCache. This method is based on the formation of an overlay network composed of "mediator" nodes, so it is applicable only to fixed topology networks. In [9] Y. Du proposed a complete framework for data retrieval and caching in mobile ad hoc networks. It is built on an underlying routing protocol and requires networkwide "cooperation zone" parameter. In adhoc networks, maintaining network connectivity is either impossible or more communication expensive than the querying/caching process. The "cooperation zone" parameter makes the method hard to configure in adhoc networks. Y. Zhang [10] proposed the solution in vehicular ad hoc networks, this method aims to find the most popular and relevant data matching a user query and a popularity-aware data replacement scheme.

### *B. Content Diversity*

G. Cao [4] proposed a method where mobile nodes cache data items which are not there in its neighbors to improve data accessibility. This method aims at caching copies of the same content farther than a given number of hops and it requires the maintenance of a consistent state among nodes and is not efficient in adhoc networks. The same concept is also exploited in [5], where nodes with similar interests and mobility patterns are formed into a cluster to improve the cache hit ratio and in [6] T. Hara presented a solution, where neighboring nodes implement a cooperative cache replacement strategy. In both methods, the caching management is based on instantaneous feedback from the neighbor nodes, which requires lot of hello messages and leads to overhead and waste of bandwidth in the network.

### *C. Caching With Limited Storage Capability*

In [6], author proposed a centralized and distributed solutions to the cache replacement problem, which aim to minimize data access costs when network nodes have limited storage capacity. The centralized solutions are not feasible in ad hoc environments and the distributed scheme in [11] makes use of cache tables which is similar to routing tables in mobile adhoc networks. In [12] W. Li proposed a content replacement strategy that aims to minimize energy consumption. To find which packets should be discarded, the solution exploits the knowledge of data access probabilities and distance from the closest provider. M. K. Denko [13] employs a variant of the last recently used (LRU) technique, which favors the storage of the most popular items instead of the uniform data distribution. In [14], the popularity of data is taken into account, along with its update rate, so that items that are more frequently updated are more likely to be discarded. Similarly, in [15], cache replacement is driven by several factors, including access probability, update frequency and retrieval delay. These algorithms address both cache replacement and consistency.

### *D. Data Replication*

Many approaches have been proposed by various authors for data replication which is relevant to the data caching solution. T. Hara [16] proposed a solution for removing data replicas among neighbor nodes. D. Rubenstein [3] proposed a replication method which aims that every node close to a copy of the information and analyzes its convergence time. Finally, in [17] K. Chen presented a cross-layer approach to data replication in mobile ad hoc networks where node mobility information in the network layer helps to trigger the replication before network partitioning occurs.

## 3. Proposed solution

Proposed solution is a fully distributed caching strategy for ad hoc networks where nodes transfer the data items in a peer-to-peer fashion. In adhoc network, nodes may be resource-constrained devices, pedestrian users or vehicles on city roads. Each node runs an application to request, possibly, cache desired information items. Nodes in the network retrieve data items from other nodes that temporarily cache the requested items from one or more gateway nodes. Proposed network model allows user applications to request a data item $i$ ($1 \leq i \leq I$) that is not in their cache. Upon a request generation, the node broadcasts a “request message” for the *C* chunks of the data item. Queries for still missing chunks are periodically issued until either the data item is fully retrieved or timeout occurs.

If a node receives a new query that contains a request for data i’s chunks and it caches a copy of one or more of the requested chunks, it sends them back to the requesting node through “data messages”. If the node does not cache all the requested chunks, it can rebroadcast a query for the missing chunks acting as a forwarder which is described in the below algorithm.

**Algorithm for operation of a query packet at nodes**

```
Step 1: reception of a query
Step 2: search the query list of the request query
Step 3: if the entry is found then discard the duplicate query else add this query in the list with status pending
Step 4: search Cache for chunk if the chunk is found send the chunk to list and keep the status as solved
Step 5: repeat step 4 for all chunks
Step 6: if the chunk is still pending then check its status if it is forward then forward query with list set to this node address and chunk containing only pending status  else discard solved query
```

To overcome proliferation of data copies along the path, the only node that is entitled to cache a new copy of the information is the node that issued the query. Data messages are transmitted back to the source node in a unicast manner in the same path from which the request has been transmitted. Backtracking data is carried and updated in request messages. Intermediate nodes either act as relays for transferring data or simply overhear their transmission without relaying them. Below algorithm describes the operations at a node which receives a message containing a data chunk.

**Algorithm for operation of a data packet at nodes**

```
Step 1: reception of the data
Step 2: search data in the query list
Step 3: if the entry is found then check its status else discard data with unknown source
Step 4: if the status is solved discard duplicate data, if it is pending update status to solved
Step 4: check for the matches in the data if it is no discard data not to be relayed else check the source matches.
Step 5: if it yes use the chunk else forward data to the next node specified in the list.
Step 6: check the cache chunk status if it is yes cache data chunk for a time interval else discard used data chunk.
```

Proposed solution focus on time taken for query process but not on the query propagation, because it directly determines the network load associated with the content retrieval operation. We consider the following two methods for query propagation.

*Mitigated flooding*: This method limits the propagation range of a request by keeping a time to live (*TTL*) parameter in the query messages and every node has to wait for a query lag time before rebroadcasting a query to avoid the transmitting the already solved requests.

*Eureka* [18]: This method extends the mitigated flooding by steering queries toward areas of the network where data required is less.

Proposed model allows wireless nodes to take caching decisions based on data they have retrieved from the network. This method allows users to take such decisions by leveraging a node’s local observation i.e., the node’s ability to overhear queries and data messages on the wireless channel. In each information item, a node records the distance from the node that issues the query, where a copy of the content is likely to be stored, and the distance of the node that provides the information. Based on these observations, the node computes an index of the data presence in its proximity for each of the I items.

### *Information Presence Estimation*

The “reach range” of a node n is calculated as its distance from the farthest node that can receive a query generated by node n itself [19]. As an example, in an ideal setting all nodes have same radio range, the reach range is obtained by the product of the TTL and the node radio range, frequency f at each node estimates the presence of every data item within its reach range and 1/f is the taken for each estimation step. A node uses the information that was captured within its reach range during time interval i to calculate a provider counter by using application-layer data and a transit counter by using data that were collected through channel overhearing in a cross-layer fashion.

### *Large-Sized Caches: Computation of the Content Drop Time*

In this case nodes have a large-sized cache can store a large portion of the data. But node should not cache same data for long time because the same memory is used to store data for various services that run in it. In the proposed model it is very important to calculate a cache drop time for the retrieved data items to improve the cache utilization without affecting the performance of data distribution system.

Cache drop time is denoted as that node n computes at the end of time interval i for data item k. the drop time applies to all chunks, belongs to data item k, that will be received during time step (i + 1). To compute $A_k(n,i)$ node n estimates an overall probability of data presence, by calculating the presence indices p(n, i) of all chunks of information k. Because p(n, i) is the sample of the chunk presence, node n finds the amount of time taken for these samples. If all nodes run in the algorithm, the best guess that node n can take to determine the presence index is to use its local estimate of the cache drop time $A_k(n, i-1)$, assuming that it is not very different from its neighbors'. Consistent with this reasoning, the contribution of a presence index computed at step j should only be considered for a time $A_k(n, j-1)$. However, discarding contributions exactly after a time $A_k(n, j-1)$ leads to an ON/OFF behavior and yields discontinuities in the caching decision process. Moreover, a hard contribution removal threshold is inconsistent with the uncertainty in the knowledge of the neighbors' caching times, the value $A_k(n, j-1)$ used by node n may differ from the cache drop time computed by the nodes within n's reach range.

$$A_k(n, i) = (1 - p(n, i))MC \qquad (1)$$

where MC denotes maximum cache drop time

***Small-Sized Caches: Content Replacement***

Nodes with small-sized cache, cannot store all data but they should decide which items to store and discard every time when memory is full. So it is very essential to find a good content replacement policy that balances distribution of data in the network so that all content is as "close" as possible to the requesting node.

$$\hat{w}_i(n,k,j) = \begin{cases} 1, & \text{if } j-k \leq \Gamma(n,k) \\ 0, & \text{otherwise} \end{cases}$$

$$\text{with} \quad \Gamma(n,k) = \lfloor f\hat{\chi}_i(n,k-1) \rfloor . \qquad (2)$$

The factor $w_i(n, k, j)$, weighting the contribution of samples $p(n, k)\forall c$ and at time step j, should be different from time step j-1 to avoid the sudden removal of chunks from the memory upon the arrival of new content.

The completeness of item *i*, estimated by node *n* from samples observed at time step *k*, as

$$\hat{\phi}_i(n,k,j) = \hat{w}_i(n,k,j)\frac{1}{C}\sum_{c=1}^{C} p_{ic}(n,k) \qquad (3)$$

and the overall presence index as

$$\hat{p}_i(n,j) = \sum_{k=j-\tau}^{j} \hat{\phi}_i(n,k,j) \qquad (4)$$

## 4. Experimental Setup:

We have implemented our proposed algorithm in NS2, which has been highly validated by the networking research community.

TABLE I: Simulation Parameters

| **Parameters** | **Value** |
|---|---|
| MAC Layer | IEEE 802.11 |
| Number of nodes | 100,200,300 |
| Node mobility speed | 0 – 50 m/sec |
| Simulation Area | 600*600 |
| Simulation Duration | 200 sec |
| Traffic Flow | TCP ,CBR |
| Mobility Pattern | Random wave point |
| Packet Size | 512 |
| Transmission range | 100m |

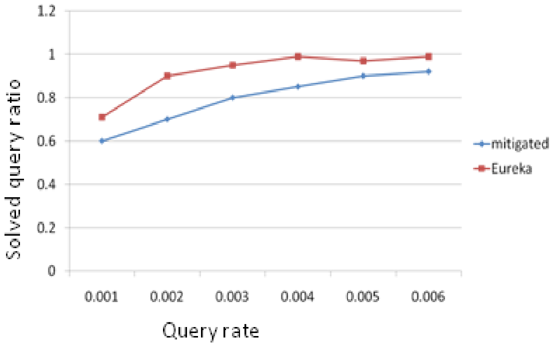

**Figure 1: solved query ratio**

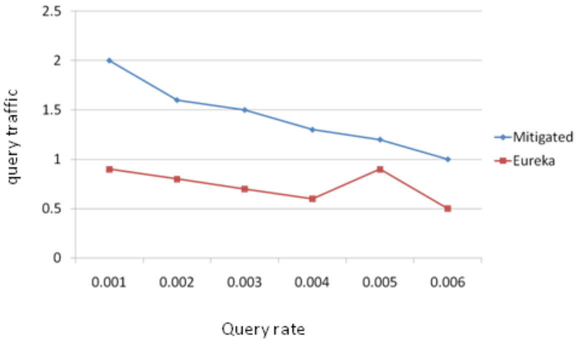

**Figure 2: query traffic**

Fig. 1and 2 shows the solved-queries ratio and the amount of query traffic as $\lambda$ varies linearly. Whenever DetCache is used it provides higher the query rate and more number of nodes are caching data item. Eureka is reducing the propagation delay of unnecessary queries and producing higher solved-queries ratio.

## 5. Conclusion

In ad hoc networks due to high mobility and lack of fixed infrastructure network disconnections occur frequently. Hence data accessibility in ad hoc networks is lower than in the fixed infrastructure networks. We address cooperative caching problem to improve data accessibility in adhoc networks. Proposed solution is based on the size of the cache in nodes. For large-sized caches, nodes take decision independent of each other whether to cache some data and how long. In small-sized caches, a data replacement strategy allows nodes to store newly received data while maintaining the better performance in the data distribution system. Performance of the proposed caching algorithms is evaluated by simulating in NS2 under various ad hoc network scenarios. Simulation results reveal that, proposed solution successfully created data diversity in the adhoc networks which leads to an efficient data access.

## REFERENCES

[1] 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, Aug. 2008.

[2] 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, Second Quarter, 2009.

[3] B.-J. Ko and D. Rubenstein, “Distributed self-stabilizing placement of replicated resources in emerging networks,” IEEE/ACM Trans. Netw.,vol. 13, no. 3, pp. 476–487, Jun. 2005.

[4] G. Cao, L. Yin, and C. R. Das, “Cooperative cache-based data access in ad hoc networks,” Computer, vol. 37, no. 2, pp. 32–39, Feb. 2004.

[5] C.-Y. Chow, H. V. Leong, and A. T. S. Chan, “GroCoca: Group-based peer-to-peer cooperative caching in mobile environment,” IEEE J. Sel.Areas Commun., vol. 25, Jan. 2007.

[6] T. Hara, “Cooperative caching by mobile clients in push-based information systems,” in Proc. CIKM, 2002, pp. 186–193.

[7] L. Yin and G. Cao, “Supporting cooperative caching in ad hoc networks,” IEEE Trans. Mobile Comput., vol. 5, no. 1, pp. 77–89, Jan. 2006.

[8] N. Dimokas, D. Katsaros, and Y. Manolopoulos, “Cooperative caching in wireless multimedia sensor networks,” ACM Mobile Netw. Appl., vol. 13, Aug. 2008.

[9] Y. Du, S. K. S. Gupta, and G. Varsamopoulos, “Improving on-demand data access efficiency in MANETs with cooperative caching,” Ad Hoc Netw., vol. 7, May 2009.

[10] Y. Zhang, J. Zhao, and G. Cao, “Roadcast: A popularity-aware content sharing scheme in VANETs,” in Proc. IEEE Int. Conf. Distrib. Comput. Syst., Los Alamitos, CA, 2009.

[11] B. Tang, H. Gupta, and S. Das, “Benefit-based data caching in ad hoc networks,” IEEE Trans. Mobile Comput., vol. 7, no. 3, pp. 289–304, Mar. 2008.

[12] W. Li, E. Chan, and D. Chen, “Energy-efficient cache replacement policies- for cooperative caching in mobile ad hoc network,” in Proc. IEEE WCNC, Kowloon, Hong Kong, Mar. 2007.

[13] M. K. Denko and J. Tian, “Cross-layer design for cooperative caching in mobile ad hoc networks,” in Proc. IEEE CCNC, Las Vegas, NV, Jan. 2008, pp. 375–380.

[14] H. Chen, Y. Xiao, and X. Shen, “Update-based cache replacement policies in wireless data access,” in Proc. BroadNets, Boston, MA, Oct. 2005, pp. 797–804.

[15] J. Xu, Q. Hu, W.-C. Lee, and D. L. Lee, “Performance evaluation of an optimal cache replacement policy for wireless data dissemination,” IEEE Trans. Knowl. Data Eng., vol. 16, no. 1, pp. 125–139, Jan. 2004.

[16] T. Hara, “Effective replica allocation in ad hoc networks for improving data accessibility,” in Proc. IEEE INFOCOM, Anchorage, AK, Apr. 2001, pp. 1568–1576.

[17] K. Chen, S. H. Shah, and K. Nahrstedt, “Cross-layer design for data accessibility in mobile ad hoc networks,” Wireless Pers. Commun., vol. 21, no. 1, pp. 49–76, Apr. 2002.

[18] M. Fiore, C. Casetti, and C.-F. Chiasserini, “Information density estimation for content retrieval in MANETs,” IEEE Trans. Mobile Comput., vol. 8, no. 3, pp. 289–303, Mar. 2009.

[19] Marco Fiore, Claudio Casetti and Carla-Fabiana Chiasserini, “Caching Strategies Based on Information Density Estimation in Wireless Ad Hoc Networks”, IEEE transactions on vehicular technology, vol. 60, No. 5, June 2011