# An Efficient Adaptive Cache Update Protocol for Mobile Adhoc Networks

**C. Vijayakumaran[1], T. Adiline Macriga[2]**
[1]Assistant Professor, Valliammai Engineering College,
Chennai, Tamilnadu, India.
[2]Professor, Information Technology, Sri Sairam Engineering College,
Chennnai, Tamilnadu, India.

**ABSTRACT**
*The cooperative caching is one of the most attractive schemes used in the area of Mobile Adhoc Networks (MANETs) to improve the data access performance. The cached data is shared between multiple mobile nodes which cooperatively manages the cached contents. The node mobility and unpredictable network partitioning creates additional overhead, latency and reduces data access ratios in the highly dynamic network environment. An efficient adaptive cache update protocol (ACUP) is proposed in this paper, which address the above problems effectively. A Rendezvous Region (RR) is adopted in the proposed work to exploit the cluster based approach that satisfies the data consistency requirements among partitioned network. A content server transmits data through cluster head nodes (H-Nodes) to local mobile nodes (L-Nodes) in a rendezvous region. The ACUP provides the consistency of cached data in L nodes by utilizing a Time-To-Live (TTL) threshold value. The simulation results show that the proposed scheme outperforms the existing scheme called Flexible Cache Consistency Maintenance (FCCM) in terms of reduced delay, control overheads and increased packet delivery ratio.*

***Keywords:*** Mobile Adhoc Networks (MANETs), Cooperative caching, Adaptive Cache Update Protocol (ACUP), Rendezvous Region (RR), Time-to-Live (TTL), Flexible Cache Consistency Maintenance (FCCM).

## I INTRODUCTION

Mobile Ad-hoc Network (MANET) is a category of wireless networks which is capable of operating without the support of any fixed infrastructure. It is peer-to-peer, self-configured network in which mobile terminals (such as PDAs, cell phones, tablets and laptops) are connected by wireless links. The application areas of MANETs include disaster relief, emergency response, etc. [1] and become the essential ingredient of the next generation networks [2]. Due to mobility of individual nodes, the network topology becomes highly dynamic which enables frequent and unpredictable changes in the nodes connectivity. Hence, some pairs of nodes may not be able to exchange their information directly with each other and should have to depend on some other nodes so that the information is being correctly delivered to the destination node. This scenario can be referred to as multi-hops networks or store-and-forward networks [3] and is shown in Figure 1. One of the most common data access applications of Mobile Ad-hoc networks is content sharing in which a user can share some information, such as music, video, and document files, with other user(s) on the network. The user may also search for required information and find it from other nodes and downloads them from First, long delay in accessing remote station via multi


**Fig 1 Multi-Hops / Store-and-Forward Networks**

hop-communication links leads to high energy consumption. Second, frequent accesses to the database server by many nodes cause a high load on the server and reduce the server response time. Third, the network capacity degrades when network partition occurs due to node's mobility. To overcome the above limitations, an efficient data caching method is introduced in this paper to reduce access delay and bandwidth. To enhance the performance of proposed system, a cluster based approach is used to explore the idea of cluster head and the capability of in-network data caching. Specifically, a Rendezvous Region (RR) – is a cluster with subset of mobile nodes in the network. Within a RR, nodes are one-hop away from each other and it may access a head node (it may be either static node or node with limited mobility) acts as a cluster head that serves as a gateway to the other side networks (Internet or web servers). When a mobile node (MN) visits a RR, it receives data from the AP (i.e., Cluster Head) and stores it in its local cache and disseminates the collected information to its neighbor node when it is requested by the neighbor.

In the data caching approach, the data is widely partitioned and cached across many nodes of the network. The dynamic cached content is periodically modified and updated which requires consistent method to ensure that all cached data are consistent with respect to the source. The mobile node may change its location during the data transmission process. A node in the network cannot retrieve the required data from a remote terminal due to the mobility and network partitioning. Hence, it may cache the data retrieved from remote source to share with its neighbors which improves data availability in the network. However, the overhead in the query latency and response time drastically decreases the performances of the network. The remote source node must ensure the consistency of cached copies of the data which makes the data consistency becomes a challenging problem in the mobile environment.

In this paper, cluster based data caching algorithm is proposed to improve the overall performance of the network by reducing the delay and response time. The rest of the paper is organized as follows: The related works are described in section II. The network model and the proposed algorithm are presented in section III. Section IV describes the performance evaluation of the proposed algorithm and section V concludes the paper.

## II RELATED WORKS

There are many data caching and consistency maintenance schemes for distributed environments have been proposed by G. Cao et al [5]. However, they cannot be suitable for MANET due to node mobility, dynamic topology and energy constraints. The data prefetching techniques presented by Grassi, V [6] provides good trade-off between delay and the cost of power in broadcasting environments. H. Jin et al [7] have proposed a novel scheme based on a selective push algorithm for cooperative cache consistency maintenance over MANETs. However, they fail to reduce control overheads involved in data transmission. Narottam Chand et al [8] have presented an approach, called zone cooperative (ZC) for caching in MANETs. In this scheme, one-hop neighbors of a mobile node establish a cooperative cache zone which reduces the cost for energy consumption and message exchange but increases control overheads due to more control packets exchange. Yu Huang et al [9] have proposed an algorithm to provide cached data consistency maintenance by using a hybrid approach where server utilizes self-learning technique based on history of past queries initiated by mobile clients. In [10-13], an extensive number of schemes have been proposed which give higher priority to data accessibility than access latency. However, the traditional cache replacement policies like LRU, LFU, etc. cannot be directly applied to the MANET environments. The proposed ACUP scheme provides the consistency of cached data by utilizing a Time-To-Live (TTL) threshold value and outperforms in terms of reduced delay, control overheads and increased packet delivery ratio.

## III NETWORK MODEL AND ALGORITHM

In the proposed network model, two types of nodes are used: Local mobile nodes (L) with low caching capacity, low power and lower transmission range and Head nodes (H) with high power, high caching capacity and have a long transmission range. It is assumed that at least one H node is present in every RR and number of H node << number of L node in a RR. It is possible that a RR may not cover an L node and is referred to as isolated mobile node. To exploit the benefits of gateway node as cluster head, the initial phase of the proposed algorithm forms a hierarchical structure of the network as shown in figure 2.

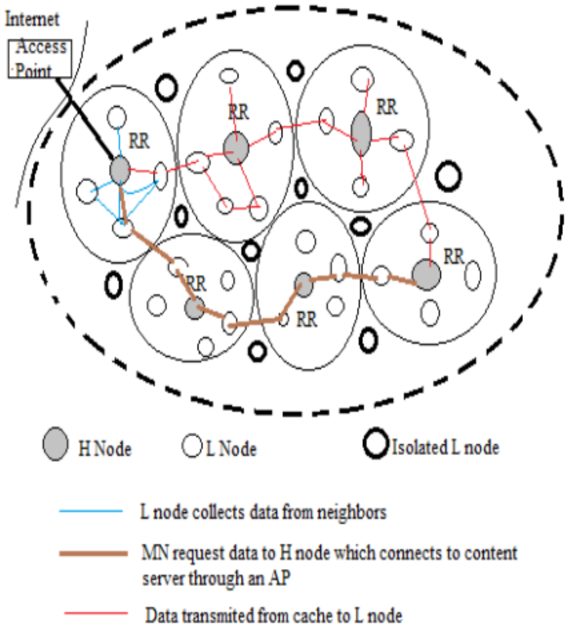


**Fig 2 Proposed Network Model**

The initial phase of the proposed algorithm called Adaptive Cache Update Protocol (ACUP) finds the neighbor nodes with bidirectional links. Each node periodically sends a 'hello' message containing its own ID (with other network related parameters) and its discovered neighbors. All nodes then construct two tables, a neighbor table (NT) and a cache table (CT) using the following procedure.

(a) Each node broadcasts 'hello' packets within one hop nodes and informs about its type and CT status initially has a null value.

(b) Each node waits for a random period of time T and uses the received 'hello' packets to construct its neighbor table and updates its CT if the received message type is a request for data access.

(c) When a node receives the 'hello' packet, it will check whether its own information is available in the hello packet. If yes, the bidirectional link between the current node and the sender of the packet will be determined. Then the sender information will be updated into the CT for future reference.

To exploit the benefits of Rendezvous Region (RR), a novel algorithm is proposed. In this algorithm, the H node is selected as the head node (cluster Head) of the RR. The H node creates a loose coupling relations ship with its one hop neighbors. Only L nodes come under the coverage of H nodes can participate in the communication process involved within the RR. Isolated L nodes may join the nearest RR region at any time. The proposed algorithm has two features. First, the loose coupling between H node and L nodes avoids overheads caused by node mobility and cluster maintenance when the density of H nodes with in a RR is small. Second, the proposed scheme can be adaptive to the density of H nodes.

All nodes update local cache table contents by exchanging control information during second phase (Cache Table Update) of the proposed algorithm. Notice that the initial neighbor table of an L node stores local topology information based on discovered bidirectional links. The detailed procedures for updating local CT are presented below:

- Each L-node broadcasts its CT content information to its neighbors based on the NT constructed in the initialization phase. The CT information is updated at every node when a new control packet is exchanged between H node and L node. Notice that control information has been exchanged with in the RR and it will be controlled by time-to-live (TTL). Because TTL is very small, exchanging control packets will not incur much overhead to the network. The control packet contains the details of most accessed data items available in the respective RR.
- After constructing local cache table at every L nodes as in step 1, the H nodes wait for any new request from the L node. For every new request, it updates the local CT of each L node by broadcasting the control packets with in the RR. Every L node in the RR must send its request to H node of the requested data is not cached in its local CT.
- H node then computes the access rate from the frequency of new request made with in pre-determined threshold value $T_H$. For any new data request from an L node in the RR, H node compares it with $T_H$. If the new access rate is > $T_{H,}$ the H node set a flag in its CT to reflect the changes in the next control packets exchanges. H node also sends a feedback message to the content server to update its own cache table. Note that if the local CT is full and a new data request exceeds the threshold value, then H will replace with the leased accessed item in the table.
- If the H node fails in a RR and wants to elect a new H node, the next H node will be the one with high capacity and power than the old H node. The mirror image of CT of the old node is stored in the new H node.
- Content Server determines the TTL value for every pre fetched data item adaptively using the equation (1)

  $$TTL = F * IURI(t) \qquad \text{Eq. (1)}$$

  where F is a system determined factor and IURI is the inter-update request interval.

In heterogeneous network environment, some nodes may have more cache capacity and power than other nodes. For instance, a laptop can have more processing capacity, power and cache capacity than a cell phone. In such situation, these nodes will be more active and give instant response to other nodes' requests, as they can cache more data items in their local cache and remain active for a longer time. Hence, such a node can be chosen as H node in the RR. This paper does not focus on the selection of H node and how to form the rendezvous regions with in a network topology. Any clustering algorithm can be utilized for this purpose. Network area can be virtually extended such that any RRs have the same size in the network. The size of any RR in the network topology is an important parameter in the network partitioning process. All L nodes in a RR can reach to every other RR in the network at point in time. The value of RR size '$R_z$' is derived from the equation 2

$$R_z = (r_L + r_H) / \sqrt{2} \qquad \text{Eq. (2)}$$

where, $r_L$ is the transmission range of an L node and $r_H$ is the transmission range of H node ($r_L >> r_H$).

## IV PERFORMANCE EVALUATION

The proposed approach was simulated using NS2 discrete event simulator with channel capacity 3 Mbps. The IEEE 802.11 DCF (Distributed Coordination Function) is used as MAC Protocol for Wireless LAN. The link layer notifies the network layer about the link break occurred in the network. The random way point mobility model is used for node mobility in an area of 500 m x 500 m for 100 seconds simulation time. The transmission range of L node is fixed as 100 m and for H node is 250 m, respectively. The variable cluster size is used and the speed of the L nodes is also varied. However, for H nodes, it is assumed to be static. The simulated traffic is assumed to be in Constant Bit Rate (CBR) and the data query and cache updates processes are based on Poisson process. The simulation settings and parameters are summarized in table 1.

**Table 1: Simulation settings**

| Parameters | Values |
|---|---|
| No. of L nodes | Variable (Between 10 – 100) |
| No. of H nodes | Fixed |
| Tx Range for L nodes | 100 meters |
| H nodes | 250 meters |
| Simulation period | 100 seconds |
| Cache Size – L Nodes | 1 MB |
| H Nodes | 3 MB |
| Mobility model | Random way point |
| Speed | Variable |
| Inter update request Interval (IURI) | 20 seconds |
| Bandwidth | 3 Mbps |
| TTL Threshold | 5 Seconds |

The first scenario was simulated by varying the number of nodes with speed of the L nodes. When the density of the nodes increased in a given RR, the query latency and control overheads also increased due to additional caching overheads which causes deficiency in overall packet delivery ratio.

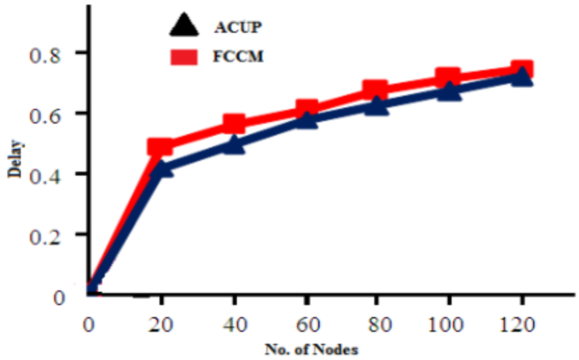


Fig 3 No. of Node Vs Delay

The proposed Adaptive Cache Update Protocol (ACUP) is compared with Flexible Cache Consistency Maintenance (FCCM) [14]. The ACUP shows lower delay than FCCM.

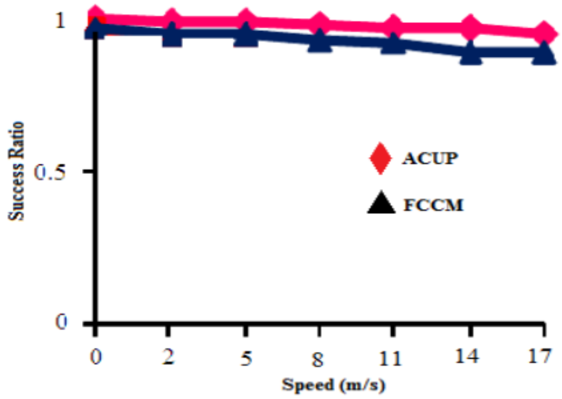


Fig. 4 Speed Vs Success Ratio

The ACUP scheme achieves better success ratio than the FCCM and is illustrated in Fig. 4. The overhead incurred in control message transmission is reduced in ACUP by utilizing cluster based approach in rendezvous region. Finally, the mobile node speed is plotted against latency incurred in query response is illustrated in the figure 5. From the figure, it is evident that the proposed ACUP has less query latency.

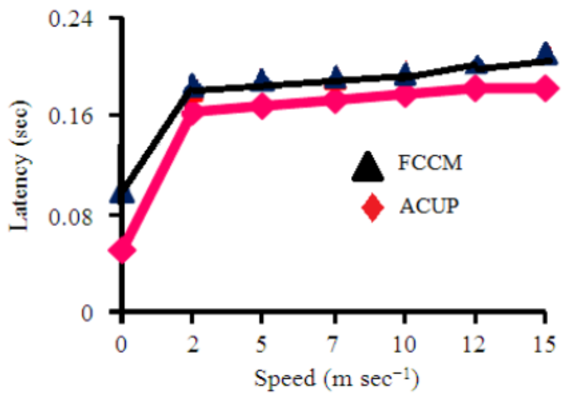


Fig. 5 Speed Vs Latency

From the simulation, it is clear that the proposed APUC scheme reduces latency, control overheads and packet success ratio and out performs than the FCCM scheme.

## V CONCLUSION

This paper has been designed to provide the adaptive cache update protocol for Mobile Adhoc Network (MANET) environments. The proposed adaptive approach utilizes a Rendezvous Region (RR) to emulate the cluster based approach. Each RR consists of two types nodes L node and H nodes both contains cache table (CT) to record each data item requested from content server. In heterogeneous MANETs, some nodes may be more efficient and powerful than other nodes which are responsible for maintaining the information in the Cache Table of different nodes belong to the RR boundary. The ACUP provides the consistency of cached data by utilizing a Time-To-Live (TTL) threshold value. The simulation results exhibits that the proposed scheme outperforms well in terms of reduced delay, control overheads and increased packet delivery ratio.

## REFERENCES

[1] M. Frodigh, P. Johansson, and P. Larsson. "Wireless ad hoc networking: the art of networking without a network," *Ericsson Review*, No.4, 2000, pp. 248-263.

[2] A. Goldsmith, M. Effros, R. Koetter, M. Medard, A. Ozdaglar, and L. Zheng, "Beyond Shannon: the quest for fundamental performance limits of wireless ad hoc networks," *IEEE Commun. Mag.*, vol. 49, no. 5, pp. 195–205, May 2011.

[3] John Tengviel, K. A. Dotche and K. Diawuo, "The Impact Of Mobile Nodes Arrival Patterns In Manets Using Poisson Models", International Journal of Managing Information Technology (IJMIT) Vol.4, No.3, August 2012, Page 55 – 71.

[4] Sharan, "A. Exploiting Semantic Locality to Improve Peer-to-Peer Search Mechanism". Master thesis in computer science, Rochester Institute of Technology, 2006.

[5] Cao, J., Y. Zhang, L. Xie and G. Cao, "Consistency of cooperative caching in mobile peer to peer systems over MANET" International Conference in Distributed Computer System, IEEE Xplore Press, pp: 573-579. 2005.

[6] Grassi, V. "Prefetching policies for energy saving and latency reduction in a wireless broadcast data delivery system,." Proceedings of the 3rd ACM International Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems. 2000.

[7] H. Jin, J. Cao and S. Feng. A Selective Push Algorithm for Cooperative Cache Consistency Maintenance over MANETs. In Proc.Third IFIP Int'l Conf, Embedded and Ubiquitous Computing, Dec. 2007.

[8] Narottam Chand, R.C. Joshi and Manoj Misra,"Efficient Cooperative Caching in Ad Hoc Networks", IEEE Transactions, pp.1-8, 2006.

[9] Yu Huang . "Cooperative cache consistency maintenance for pervasive internet access". In State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China, 2009.

[10] L. Yin and G. Cao, "Supporting cooperative caching in adhoc networks," IEEE Trans. Mobile Comput., vol. 5, no. 1, pp. 77-89, Jan. 2006.

[11] H.Artail, H.Safa, K.Mershad, Z.Abou-Atme, and N.Sulieman, "COACS: A cooperative and adaptive caching system for MANETs,"IEEE Trans. Mobile Comput., vol. 7, no. 8, pp. 961-977, Aug. 2008.

[12] N.Chand, R.C.Joshi,and M.Misra,"Cooperative caching strategy in mobile ad hoc networks based on clusters, Wireless Person.Commun" pp. 41-63, Dec. 2006.

[13] J. Tian and M. K. Denko, "Exploiting clustering and cross layer design approaches for data caching in MANETs," in *Proc. 3rd IEEE Int. Conf. Wireless and Mobile Computing, Networking and Communications, (WiMob)*, 2007.

[14] Huang, Y., J. Cao, B. Jin, X. Tao and J. Lu, "Cooperative cache consistency maintenance for pervasive internet access". Wireless. Commun. Mobile Comput., 2010: 436-450.