A Literature Review on Collaborative Caching Techniques in MANETs:

Issues and Methods used in Serving Queries

Mohamed Ahmed Elfaki
Department of Computer Science,
College of Computing & Information Technology, Shaqra University, Riyadh, Saudi Arabia
malfaki@su.edu.sa

Majdi Abdellatief
Department of Health of Informatics,
Inaya Medical College,
Riyadh, Saudi Arabia
khwaja24@yahoo.com

Ali A. Alwan
Department of Computer Science,
International Islamic University Malaysia,
Kuala Lumpur, Malaysia
aliamer@iium.edu.my

Abubaker Wahaballa
Department of Computer Science,
Arab East Colleges,
Riyadh, Saudi Arabia
wahaballah@arabeast.edu.sa

Abstract—**Collaborative cache management in Mobile Ad Hoc Networks (MANETs) environment is considered as an efficient technique to increase data accessibility and availability, by sharing and coordination among mobile nodes. Due to nodes' mobility, limited battery power and insufficient bandwidth, researchers addressed these challenges by developing many different collaborative caching schemes. The objective of this paper is to review various collaborative caching techniques in MANETs. Collaborative caching techniques are classified by methods used in serving queries, such as: hop-by-hop discovering, broadcasting messages, flooding, and query service differentiation. This review reveals that techniques utilizing hop-by-hop methods have better performance compared to others, especially techniques using additional strategies.**

Keywords-mobile ad hoc network; cooperative caching; caching methods

I. INTRODUCTION

Mobile computing is a new paradigm emerged from the exponential development of wireless networking. Advances in wireless communication engage focus in Mobile Ad-hoc Networks (MANETs). This allows a portable device to access required information services and data via a shared infrastructure, without considering physical location and movement behavior. Dynamic mobility, disconnections, limited bandwidth, variable capacity links, limited resources and energy are the main characteristics of MANETs [1-4]. Therefore, accessing distant data in such a situation reduces system's performance in terms of average delay, hit ratio, and bandwidth. The typical wireless mobile network architecture for MANETs, consists of a wired network, mobile nodes (MNs), base stations (BSs) and a database server (DB) connected to the wired network. BSs communicate with mobile nodes located within the coverage range, known as wireless cell [5, 6]. There are two types of wireless channels, defined uplink and downlink. Uplink channel is utilized for sending requests from MNs to the BS, while downlink is used by the BS to send information to the MNs within the cell range. One of the functions of the BS is to ensure MNs are able to communicate with other networks by sending their queries. The BS also manages handoff and mobility functions [7].

Caching frequently accessed data in mobile devices is an effective method to enhance performance in a MANET. Hence, while requests are satisfied locally, data retrieval response time is reduced [8]. However, a reliable cache management in MANETs is difficult to be achieved due to dynamic movement, disconnection, limited bandwidth, and limited energy consumption [9-11]. Therefore, accessing distant data in MANETs is costly in terms of delay, energy consumption, and bandwidth usage [12, 13]. Collaborative caching is developed in order to share the cache content among a number of participating nodes. Figure 1 shows the four fundamental schemes for sharing cache content among participating nodes. In IndexCache each node caches its own query result and a distributed index is built on top of these local caches to facilitate sharing. In CacheData each node is assigned a particular part of the cache data space and the results of each query are not stored in the node who requested it, but instead at the node that is responsible for that part of data space [14]. Furthermore, intermediate nodes cache data to serve future queries. A limitation of CacheData is that cached data items in intermediate nodes could take a lot of space. CachePath provides the hop count information between source and destination. By caching the data path for each data item, bandwidth and query delay can be reduced since data can be obtained in even less hops [15]. CachePath is used to redirect future queries along the cached path. In MANETs, the node caching the data items might leave the transmission range or it

Corresponding author: Mohamed A. Elfaki

might replace or remove cached data items due to cache size limitations. HybridCache is developed to further improve performance by considering the advantages of both schemes CacheData and CachePath, while avoiding their weaknesses.

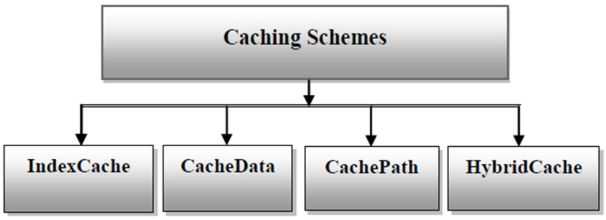

Fig. 1. Caching Schemes

Collaborative caching is particularly attractive in environments where the network is constrained in terms of bandwidth and disconnections. Collaborative caching also offers several benefits, since it enables efficient utilization of available resources by storing different data items and sharing them among neighbor nodes [16, 17]. Furthermore, by introducing peer-to-peer device partnership, it can reduce query latency, provide a higher number of hits and increase the accessibility of requested data items [18]. Collaborative caching improves the performance of mobile query service compared to simple caching in MANETs for two reasons. At first, in collaborative caching, requested data items can be retrieved from neighboring nodes instead of the remote database server, reducing the risk of dropped packets caused by network congestions and link failures. Additionally, nodes can still receive requested data items from neighboring nodes while network partitioning occurs [19].

II. RELATED WORKS

The survey in [20] outlines different studies on cooperative caching in mobile P2P networks. This survey described cooperative caching based on cache placement, replacement, consistency and security. Authors concluded that most studies on cooperative caching in mobile P2P focus on resource searching instead of data access. Authors in [12] classified and compared existing cooperative caching strategies based on performance metrics, such as query latency and cache hit ratio. Authors in [12] found that having a strong consistency in caching schemes, lead to efficient information retrieval architecture. A hypothetical analysis of various cooperative caching schemes is proposed in [10, 21], where authors presented the advantages and limitations of previous schemes in a MANET, in finding a data item with less resources (e.g. network bandwidth, energy etc.) and improving the performance interim of data availability and latency time. This review not only extends and updates previous reviews [10, 12, 20, 21] but also proposes collaborative caching for future researches in MANETs and has the following differences:

- Different goal: Our main purpose is to understand, classify, and analyze the existing methods used to serve queries in collaborative caching techniques in MANETs, in order to direct future research, while other reviews [10, 12, 20, 21] mainly aimed to provide an overview of quality cache management and performance metrics analysis.
- Comprehensive studies: This paper reviews 35 studies. Authors in [10, 21] examined only 14 articles, while authors in [12] and [20] examined 7 and 30 respectively. Comprehensive study has advantages, since it reduces the bias and allows for a general conclusion.

III. ANALYSIS AND DISCUSSION

This section classifies previous studies based on methods utilized to access and discover data items in collaborative caching in MANETs. These methods are: hop-by-hop discovering, broadcasting messages, flooding, and query service differentiation.

A. Hop-by-Hop Discovering

Hop-by hop discovering is one of the methods utilized by collaborative caching approaches. Works presented in [14, 15, 22] contain collaborative caching approaches to enhance data access in ad hoc networks. These approaches have similar schemes like IndexCache, CacheData, CachePath and HybridCache, while their cache discovery method is hop-by-hop forwarding. In other words, each node along the way to the DB is queried for the data. There are several routing algorithms that have been designed to route messages in MANETs. Some of them are proposed to reduce bandwidth consumption and query delay, where the amount of hops between data center and the requesting node should be as few as possible. However, a drawback of these schemes is that caching information of a node cannot be shared if the node does not lie on the path between the requesting node and the DB [9, 29, 23]. Besides that, in case a failure on the path, the request packet might be lost, or delayed until the node reestablishes the connection. Furthermore, they do not provide different services for different classifications of queries. Another example of collaborative caching using clusters is proposed in [24], which is called the Cluster Cooperative (CC) Caching. The main purpose of this approach is to reduce the cache discovery overhead and enhance collaborative caching performance. CC divides the MANET on geographical networks (clusters) which are equal in size. In each cluster, CC dynamically elects a node as Cache State Node (CSN) to maintain the Cluster Cache State (CCS) information of different nodes within its cluster domain. The CCS for a client is the list of cached items along with their TTL field. To improve system's performance, all mobile clients within the same cluster form a collaborative cache system among them. In CC, if a query is missed locally, the client requests the data item from the cluster members by forwarding a request to the CSN. If the required data item is missed within the cluster members' cache, the query is forwarded to clients along the path towards the DB. If the required data item is found along the path towards the DB (remote cache hit), the request can be served, without further redirecting towards the DB. However, if the query is missed locally, within the cluster's members and remote cache, the query is forwarded to the DB. One of this approach drawbacks is that caching information of a client cannot be shared if does not lie on the path between the requesting node and the DB [9, 29]. Furthermore, the query in CC is only served through node by node forwarding along the path towards the DB and not guided and coordinated to a particular data source.

Zone Cooperative (ZC) caching approach is proposed in [16] in order to minimize caching overhead by providing cache replacement policy. In ZC, each data item is stored in a particular data source, which acts as server, where each mobile host (MH) caches frequently accessed data items. The advantage of the ZC caching design is that it enables nodes to share their cached data items with neighboring nodes located in the same zone. Thus, cached data items satisfy not only the node's own needs but also the needs of other nodes that are passing through it towards the DB. In ZC, each MH first searches the data item in its local cache, its zone cache after miss occurred at local cache, before forwarding the request to the next node that lies on a path towards the DB. In ZC, MH caches the required data items based on server's location. Therefore, a data item is not cached if the server of the requested data item resides in the same zone of the requested MH. If there is not sufficient cache space for the data item, ZC utilizes a least utility value (LUV) based replacement policy to enhance the efficiency of caching. However, query latency might increase if intermediate nodes lying between the requested node and the DB do not have the required data item [10]. Furthermore, there is no query service differentiation in ZC, where in real applications some queries should be served urgently. Group Caching (GC) is proposed in [11], which uses one-hop neighbors to maintain a localized caching status for performing the tasks of data item discovery and caching replacement, when a query is initiated by a mobile host. In GC, each MH maintains only one-hop neighbors in a group. Each MH sends its caching status to its group members. Each MH sends a "hello" message within transmission range to form a group of one-hop neighbors. A query is checked in the local cache once an MH receives it. If the local cache is missed, group table is checked to redirect the request to group members in order to serve the request, which is known as remote cache hit. If the remote cache is missed the request is forwarded to the next group along the path towards the DB. Once the intermediate group along the path receives the request, it checks the group table for the requested data; if the requested data item is found, it sends the request's response directly to the MH requester. If a miss occurred at the intermediate groups, the query is forwarded to the DB. However, GC consumes more energy to check the requested data item within the group members along the path towards the DB. Besides, no service differentiation is utilized in serving queries. Thus, GC consumes energy in MH since each group members towards the DB have to be checked [10].

Another collaborative caching design is proposed in [19], based on cluster collaborative caching scheme for mobile ad hoc networks. This scheme improves caching performance in terms of data accessibility, query delay, and distance compared to caching schemes without collaborative caching policy. Information searching is based on clustering architecture, combining different methods, ranging from local to global search. In this scheme, MHs search for the required data items within neighbor nodes after a local miss. Initially, an MH checks whether the data server is its neighbor. If it is, the query is sent to the data server. In case the data server is not a neighbor of the MH, two cases are possible: a) MH is a cluster header, or b) MH is a cluster member. If MH is a cluster header, the ID list is checked for the required data item. If the ID is found, the query is forwarded to the corresponding cluster member. If the ID is not found, the cluster header forwards the query to the data server. On the other hand, if MH is a cluster member, it unicasts the query to its cluster header. Although the broadcasts based data search locates the nearest required data item, the energy and bandwidth cost of the flooding search is significantly high for a MANET. This flooding search can be problematic for network communication [17, 23]. Furthermore, queries are served without considering service differentiation. This causes query delay in fetching the requested data items. Authors in [9] introduced a cooperation and adaptive database caching system for MANETs known as COACS. Its main idea is that node caches submitted queries by indexing them to facilitate the process of locating the required data item in less delay. Nodes in COACS can play one of two possible roles, which are cache node (CN) and query directory (QD). QD is responsible for caching previously queries submitted by the requesting nodes, while CN is responsible for caching data items that are responses to queries. If a node requires a data item not cached in the system, the requesting node (RN) is served by retrieving the data item from the database server. RN could be any node including a QD or a CN. Once RN received the response, it will act as a CN by caching the data item. The nearest QD caches the query and makes an entry in its hash table to link the query to its corresponding data item. The query is checked at the nearest QD, and if is not indexed, the request is forwarded to another QD. To minimize the response time of the system, COACS forms several QDs that are used as a distributed indexing system, whereby if one QD receives a query that is not indexed, it is forwarded to the nearest QD. This is continued until the query is satisfied by one of the QDs or the database server. However, even though COACS decreases the query delay to some extent, it is inefficient as requests are served through multiple QDs in order to locate the required data item. A query may be forwarded from one QD to another even though some of the QDs did not index it . This causes delay and lowers the hit ratio in serving queries. Besides, COACS serves queries without considering service differentiation, where in real applications some queries need to be served urgently.

The approach proposed in [25] considers the progress of data discovery. In this approach, nodes within a zone form a cooperative cache, since a zone transmission range is formed based on a set of one-hop neighbors. Each node has a local cache to store frequently accessed data, not only for its requests, but also for other nodes passing requests through it. Once the data are updated at the server, nodes' cached copies become invalid. If a data miss happens in the local cache, the node first checks the requested data items in its zone before forwarding the request to the next node lying on the path towards the server. However, latency may increase if the requested data item is not cached in intermediate neighbor's nodes. In [26] the Group-based Cooperative Caching scheme (GCC) is described. This scheme is based on the concept of group caching, where each mobile host and its k-hop neighbors are allowed to form a group. Each mobile host maintained a directory of cached data items. Each MN broadcasts a message to the group in order to obtain the directories of its k-hop group members. However, overhead might be generated by mobile host's requests [27].

B. Broadcasting Messages

The approach proposed in [28], introduced a new type of caching, known as distributed group based cooperative caching in a mobile broadcast environment, called Tightly-Coupled Group (TCG). The main idea is to capture data affinity of individual peers and their mobility patterns in mobile broadcast environments where all TCGs are dynamic. It consists of two types of cooperative cache management protocols: cooperative cache replacement and cooperative cache management admission control. The latter is applied when MH experiences a local cache miss and sends a request to its peers. If some peers turn in the required data item to the MH and the local cache is not full, MH caches the data item, no matter whether it is returned by a peer in the same TCG or not. However, after a peer sends the required data item to the requesting MH, if they belong to the same TCG it must update the last access timestamp of the data item in order to raise its TTL in the global cache. In cooperative cache replacement, MH replaces the least valuable data item related to the MH itself and other members in the same TCG. This replacement satisfies three useful properties. First, the most valuable data item is always retained or kept in the local cache. Second, data items in local cache not accessed for a long period will be replaced eventually. Third, a data item in global cache which "spawns" replicas is first replaced in order to achieve an effective cache size. This approach focuses more on the availability of data items in a mobile broadcasting environment through scalable broadcast channels. However this approach relies on broadcasting messages to discover the required data items. The broadcasting message is received by all network's nodes, including those who have not cached the required data items. In addition, queries in TCG are served without considering service differentiation. Thus, serving queries based on broadcasting messages in TCG caused network congestion and delay [21]. Besides, the TCG approach does not consider the mobile limited resources as it applied push and pull based data dissemination. Authors in [23] proposed a collaborative caching technique to enhance data access in ad hoc networks. There are two protocols based on the notion of a zone. The first protocol is Index Push (IXP) Protocol, where an MN broadcasts an index message to regional nodes in order to promote a caching event. Basically, IXP is used to fetch the requested data item from a nearby neighboring node, if a node knows that its neighbor has cached it. The second protocol is called Data Pull/Index Push (DPIP) Protocol, where a requesting node may broadcast a special request message to nodes in its zone, asking them to serve its demand. This type of process works as pull-based with implicit index pushing property. DPIP allows a wider scope of local resources caching cooperation without involving any additional communication overhead. However, the protocol works well in environments where the zone contains a small number of nodes. Thus, this can be a drawback in IXP and DPIP protocols, as when a new node joins a zone, its nodes are not aware about the join node updates [29]. Therefore, less collaboration among nodes in the same zone consequently affects the cost in terms of delay when locating a data item. Furthermore, it lacks a query directory at the global level to guide queries to data sources, especially when queries cannot be served locally. Hence, lack of coordination and service differentiation in serving queries leads to inefficient bandwidth usage and server overload, since each node forwards queries directly to the DB. Authors in [30] proposed a scheme called Cooperative Caching (COCA) for conventional mobile database systems. COCA introduces a new logical layer between the mobile client cache and MSS layer, which is known as peer cache layer. This layer is invoked into two situations: firstly, when a miss occurs in local cache and secondly, when the query needs to be served by neighboring nodes. The request in COCA is served in the following stages: a) the required data item is checked in MH's local cache. b) If a local miss occurs, MH broadcasts the request to its neighboring nodes. If any of its neighboring nodes has cached the required data item, it responds to the MH. c) If none of the neighboring nodes has the required data item, MH sends the request to the MSS. When the MH is residing outside of the service range and encounters a global cache miss, it suffers from an access miss. Furthermore, COCA can reduce communication attempts with the server by allowing MHs to share their cache among them. However, COCA has longer access latency, especially when MHs encounter a global cache miss, while the request is not served based on its classification, not coordinated and not guided to a particular data source as it broadcasts the request to all neighbor nodes Furthermore, COCA is inefficient in terms of bandwidth usage due to broadcasting, which lowers system's throughput because of network flooding with request packets [21].

C. Flooding

Authors in [31, 32] proposed a new type of caching to improve data availability and access efficiency, by collaborating local resources of mobile nodes and utilizing cooperation zones and hop-by-hop resolution. There are two basic fundamentals of cooperative caching that are tackled in [31, 32], which are cache resolution and cache management. For cache management, COOP maximizes the capacity of cooperative caching by reducing caching duplications within a short distance cooperative neighborhood. On the other hand, cache resolution addresses how to serve a query with less delay, bandwidth, and energy. Furthermore, COOP resolution is a cocktail scheme that develops based on three basic schemes. In the first scheme, adaptive flooding is used to estimate a proper flooding range based on the cost to retrieve the required data items. This limited flooding is not only used for the purpose of discovering a cached node around the requesting node, but also as an announcement method to share and manage cached contents among the neighborhood. The second scheme, profile-based resolution, utilizes a historical profile of previously resolved queries to determine the nearest data item source to a requesting node. If the required data item is not fetched using those two schemes, the query is redirected to the original data item source. In the third scheme, roadside resolution resolves the query via hop-by-hop along the forwarding path towards the database server. To improve data availability and access efficiency, authors in [31, 32] utilized flooding to discover data items instead of coordinating the query to particular data sources that cached it. Serving queries using flooding and hop-by-hop resolution caused network traffic since all queries are served without considering service differentiation and are not guided to a particular data source. Hence, using flooding can be problematic for network communication [23], and consequently leads to query latency, especially when the number of queries or the number of forwarding nodes to the remote data source is increased [29].

D. Query Service Differentiation

Due to limited mobile resources, such as insufficient wireless bandwidth, disconnections and client's mobility, collaborative cache management faces vast challenges. Query latency is one of the issues that are tackled in collaborative caching management. A priority-based caching service in web caching and content distribution is proposed in [33], describing a service that differentiates the caching mechanism based on priority queues. It provides the ability to differentiate services in terms of hit rate and object lifetime, with minimal efficiency penalties compared to conventional caching techniques. Therefore, service differentiation can be beneficially applied to proxy caches and Content Distribution Network (CDN) server. However, most of the existing collaborative caching techniques in MANETs environments have been developed without considering service differentiation. Serving requests by checking each node towards the database server without considering the urgent needs for required data items causes delay and increases the number of hops. Authors in [34] review the current Quality of Service (QoS) routing mechanisms and frameworks for multimedia traffic in MANETs. They review two QoS categories: QoS solutions that depend on routing protocols for selecting a new hop based on QoS parameters and QoS solutions based on some specific design frameworks along with integration of various modules that are applicable with any routing protocol. However, there is still a wide area of research before proposing a QoS framework for MANETs, particularly for multimedia traffic, to overcome limitations such as resource reservation, connection admission control, etc. which will help in reducing the delay of query responses. The approach presented in [35] can be categorized under hop-by-hop discovery with some enhancements that apply query service differentiation. This approach attempts to enhance the data access efficiency and cache discovery performance in hop-by-hop, by adding some new features, such as node cached data item information indexing that facilitates in guiding the queries to a particular data source. Hence, this approach attempts to increase the collaborative caching among neighbor nodes, since each node is provided a Recent Priority Table (RPT) to index its neighbor nodes' cached data item information. Query classification either as priority or normal, which is specified by the requesting node, is utilized to determine to which data source the query should be forwarded. This enhances the hop-by-hop method by coordinating and serving queries with less forwarding hops. This helps serving queries in MANETs with minimum cache discovery overhead and delay. Table I presents a summary of the existing collaborative caching works, proposed to improve the efficiency of data item access and cache discovery in MANETs. Two points can be highlighted: At first, most approaches utilize hop by hop while others utilize flooding and broadcasting messages to access the data items. However, these flooding and broadcasting methods can be problematic for network communication and consequently cause query latency, if the neighbors of intermediate nodes do not have the requested data items to serve the request. Moreover, most of the presented approaches, served queries without considering service differentiation.

TABLE I. SUMMARY OF COLLABORATIVE CACHING APPROACHES

	Methods			
Reference	**BC**	**FL**	**HbH**	**QSD**
[9]	×	×	✓	×
[11]	×	×	✓	×
[14]	×	×	✓	×
[15]	×	×	✓	×
[16]	×	×	✓	×
[19]	✓	×	✓	×
[22]	×	×	✓	×
[23]	✓	×	✓	×
[24]	×	×	✓	×
[25]	×	×	✓	×
[26]	✓	×	✓	×
[28]	✓	×	×	×
[30]	✓	×	×	×
[31]	×	✓	✓	×
[32]	×	✓	✓	×
[33]	×	×	×	✓
[35]	×	×	✓	✓

BC: Broadcast, FL: Flooding, HbH: Hop-by-Hop, QSD: Query Service Differentiation

IV. CONCLUSION

This paper presented a review of various collaborative caching techniques in MANETs. The collaborative caching techniques increase the efficiency of data accessibility and availability in MANETs environments by sharing and coordination among mobile nodes. We discussed the advantages of methods utilized in each technique, in order to improve the performance of fetching a data item in MANETs by using less bandwidth, and power. Moreover, we addressed the limitations faced by the mobile database environments in MANETs. The analysis reveals that techniques utilizing hop-by-hop methods have better performance compared to those utilizing flooding or broadcasting methods This review will be helpful to future research in terms of improving data accessibility and data availability in MANETs.

REFERENCES

[1] H. Safa, H. Artail, M. Nahhas, "Enhancing cache invalidation in mobile environments", Mobility '08 - International Conference on Mobile Technology, Applications, and Systems, Yilan, Taiwan, September 10-12, 2008

[2] T. Lin, Mobile Ad-Hoc Network Routing Protocols: Methodologies and Applications, PhD Thesis, Virginia Polytechnic Institute and State University, 2004

[3] D. D. Perkins, H. D. Hughes, C. B. Owen, "Factors affecting the performance of ad hoc networks", 2002 IEEE International Conference on Communications, New York, USA, April 28–May 2, 2002

[4] J. Yoo, S. Choi, C. Kim, "Control overhead reduction for neighbour knowledge acquisition in mobile ad hoc networks", Electronics Letters, Vol. 39, No. 9, pp. 740-741, 2003

[5] A. Madhukar, T. Ozyer, R. Alhajj, "Dynamic cache invalidation scheme for wireless mobile environments", Wireless Networks, Vol. 15, No. 6, pp. 727-740, 2009

[6] Z. Wang, M. Kumar, S. Das, H. Shen, "Dynamic cache consistency schemes for wireless cellular networks", IEEE Transactions on Wireless Communications, Vol. 5, No 2, pp. 366-376, 2006

[7] L. Ni, Y. Zhu, B. Li, Q. Deng, "Optimal mobility-aware handoff in mobile environments", 2011 IEEE 17th International Conference on Parallel and Distributed Systems, Tainan, Taiwan, December 7-9, 2011

[8] G. Anandharaj, R. Anitha, "An improved architecture for complete cache management in mobile computing environments", International Journal of Soft Computing, Vol. 4, No. 3, pp. 142-147, 2009

[9] H. Artail, H. Safa, K. Mershad, Z. Abou-Atme, N. Sulieman, "COACS: a cooperative and adaptive caching system for MANETs", IEEE Transactions on Mobile Computing, Vol. 7, No. 8, pp. 961–977, 2008

[10] P. Kuppusamy, K. Thirunavukkarasu, B. Kalaavathi, "A review of cooperative caching strategies in mobile ad hoc networks", International Journal of Computer Applications, Vol. 29, No. 11, pp. 22-26, 2011

[11] Y. Ting, Y. Chang, "A novel cooperative caching scheme for wireless ad hoc networks: Groupcaching", 2007 International Conference on Networking, Architecture, and Storage, Guilin, China, July 29-31, 2007 „

[12] G. Radhamani, S. Umamaheswari, "Comparison of cooperative caching strategies in mobile ad-hoc network (MANET)", International Conferences NeCoM 2010 - WiMoN 2010 - WeST 2010, Chennai, India, July 23-25, 2010

[13] H. Safa, F. Deriane, H. Artail, "A replication based caching strategy for MANETs", 2011 International Conference on Selected Topics in Mobile and Wireless Networking (iCOST), Shanghai, China, October 10-12, 2011

[14] K. Lillis, E. Pitoura, "Cooperative xpath caching", 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, Canada, June 9-12, 2008

[15] G. Cao, L. Yin, C. R. Das, "Cooperative cache-based data access in ad hoc networks", Computer, Vol. 37, No. 2, pp. 32-39, 2004

[16] N. Chand, R. C. Joshi, M. Misra, "Cooperative caching in mobile ad hoc networks based on data utility", Mobile Information Systems, Vol. 3, No. 1, pp. 19-37, 2007

[17] M. Denko, "Cooperative data caching and prefetching in wireless ad hoc networks", International Journal of Business Data Communications and Networking, Vol. 3, No. 1, pp. 1-15, 2007

[18] X. Shen, M. M. Akon, A. Singh, "Low cost peer-to-peer collaborative caching for clusters", 2007 4th IEEE Consumer Communications and Networking Conference, Las Vegas, USA, January 11-13, 2007

[19] J. Tian, M. K. Denko, "Exploiting clustering and cross-layer design approaches for data caching in MANETs", Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMOB 2007), White Plains, USA, October 8-10, 2007

[20] X. Zhou, Z. Zou, R. Song, Y. Wang, Z. Yu, "Cooperative caching strategies for mobile peer-to-peer networks: a survey", in: Information Science and Applications (ICISA) 2016 – Lecture Notes in Electrical Engineering, Vol. 376 Springer, 2016

[21] P. Kumar, N. Chauhan, "A review of cooperative cache management techniques in MANETs", International Journal of IT & Knowledge Management, Vol. 3, No. 1, 2010

[22] L. Yin, G. Cao, "Supporting cooperative caching in ad hoc networks", IEEE INFOCOM 2004, Hong Kong, China, March 7-11, 2004

[23] C. R. Young, G. M. Chiu, F. L. Wu, "Efficient cooperative caching schemes for data access in mobile ad hoc networks", International conference on Embedded and Ubiquitous Computing, Nagasaki, Japan, December 6-9, 2005

[24] N. Chand, R. C. Joshi, M. Misra, "An efficient caching strategy in mobile ad hoc networks based on clusters", 2006 IFIP International Conference on Wireless and Optical Communications Networks, Bangalore, India, April 11-13, 2006

[25] S. Umamaheswari, G. Radhamani, "Enhanced antsec framework with cluster based cooperative caching in mobile ad hoc networks", Journal of Communications and Networks, Vol. 17, No. 1, pp. 40–46, 2015

[26] I. W. Ting, Y. K. Chang, "Improved group-based cooperative caching scheme for mobile ad hoc networks", Journal of Parallel and Distributed Computing, Vol. 73, No. 5, pp. 595–607, 2013

[27] A. Larbi, L. Bouallouche-Medjkoune, D. Aissani, "Improving cache effectiveness based on cooperative cache management in MANETs", Wireless Personal Communications, Vol. 98, No. 3, pp. 2497–2519, 2018

[28] C. Y. Chow, H. V. Leong, A. T. S. Chan, "Distributed group-based cooperative caching in a mobile broadcast environment", 6th International Conference on Mobile Data Management, Ayia Napa, Cyprus, May 09-13, 2005

[29] P. Kumar, N. Chauhan, L. K. Awasthi, N. Chand, "Proactive approach for cooperative caching in mobile adhoc networks", International Journal of Computer Science Issues, Vol. 7, No. 3, pp. 21-27, 2010

[30] C. Y. Chow, H. V. Leong, A. Chan, "Peer-to-peer cooperative caching in mobile environments", 24th International Conference on Distributed Computing Systems Workshops, Hachioji - Tokyo, Japan, March 23-24, 2004

[31] Y. Du, S. K. S. Gupta, "COOP - A cooperative caching service in MANETs", Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services (ICAS-ICNS '05), Papeete–Tahiti, French Polynesia, October 23-28, 2005

[32] Y. Du, S. K. S. Gupta, G. Varsamopoulos, "Improving on-demand data access efficiency in MANETs with cooperative caching", Ad Hoc Networks, Vol. 7, No. 3, pp. 579-598, 2009

[33] M. Feldman, J. Chuang, "Service differentiation in web caching and content distribution", The IASTED International Conference on Communications and Computer Networks, Cambridge, USA, November 4-6, 2002

[34] S. Soni, J. S. Shah, "QoS frameworks for Multimedia Traffic in Mobile Adhoc Networks: A Comparative Review", Engineering, Technology & Applied Science Research, Vol. 7, No. 3, pp. 1708-1712, 2017

[35] M. A. Elfaki, H. Ibrahim, A. Mamat, M. Othman, H. Safa, "Collaborative caching priority for processing requests in MANETs", Journal of Network and Computer Applications, Vol. 40, pp. 85-96, 2014