Enhanced POC Tree-Based Algorithm for Data Item Correlation and Cache Effective Replacement in Mobile Ad Hoc Network

[1] Y. J. Sudha Rani
JNTUH
[1,2] SMICH, Hyderabad, Telangana, India
{su.joyfull@gmail.com}

[2]M. Seetha
GNITS, Hyderabad, Telangana
India
{smaddala2000@yahoo.com}

ABSTRACT: *Minimizing delay and improving efficiency in content delivery is essential in resource hungry networks. Caching enables such networks like Mobile Ad Hoc Network (MANET) to have improved performance as cached data is shared with coordination. It also helps nodes to operate offline with the help of cached content besides promoting availability and accessibility. Caching is the process of storing repeatedly used data in a temporary memory thus avoiding unnecessary processing which leads to waste of time and resources. However, it is crucial to know how data is to be cached, how long it has to be retained and when it has to be replaced with new content as it has finite size. Cache replacement is an important activity which needs intelligence for efficiency. In this paper we proposed an enhanced POC tree based algorithm for association rule mining. The association rules of the proposed algorithm provide significant heuristics to make cache replacement decisions besides achieving correlation of data items. This algorithm employs light-weight and scalable data structure known as Nodesets. We compared the proposed approach with existing algorithms. Our simulation results found the proposed method outperforms existing ones.*

Keywords: Mobile Ad Hoc Network (MANET), Caching, Data Item Correlation, Cache Replacement

Received: 19 December 2019, Revised 16 April 2020, Accepted 6 May 2020

DOI: 10.6025/jnt/2020/11/3/81-92

1. Introduction

Mobile Ad Hoc Networks (MANETs) are networks with nodes containing no access points as they are made up of self-organizing devices. These networks are widely used in real world applications like contingencies, emergency rescue operations and so on. Due to technology innovations, the devices in MANET can also be used to transfer data. When data is to be transferred, each node in the network can act as both transmitter and receiver. When nodes have information, that information

needs to be sent to other nodes based on the request. In such cases, transferring data to other nodes in a multi-hop network causes delay, consumes more resources and even results in redundant data transfer. To overcome these drawbacks many content delivery acceleration schemes came into existence as explored in [5]. Caching is one of the strategies that can leverage content delivery acceleration besides reducing utilization of resources. Apart from coding, the other techniques for content delivery acceleration include data redundancy elimination, pre-fetching, data compression, handover optimization, queue management, network coding, TCP optimization, session layer optimization, content adaptation, network aggregation and traffic data offloading. In this paper, we studied correlation of data items for effective cache replacement in MANET.

Cache replacement is the policy in which cache memory is replaced with new data. This is an important decision since cache has finite size. When the cache is full, it is not immediately replaced. However, it is done based on certain factors. In fact, the cache replacement approach needs heuristics to understand the need for replacement. When to replace is an important decision. Many cache replacement schemes came into existence. They include Regionally Maintained Cooperative Caching (RMCC) [3], A-CACHE [4], Tightly Cooperative Caching Approach (TCCA) [6], Group-based Caching Scheme (GCC) [10], Adaptive Cooperative Caching Strategy (ACCS) [12], Least Utility Value (LUV) [16], push-pull cache consistency method [17], Mometic Algorithm (MA) [18] and association rule mining (ARM) [19].

Almost all schemes found in the literature are focusing on cache replacement and cache management techniques. Little information was found on the exploration of data mining techniques like association rule mining for extracting heuristics to make well informed decisions on replacement. In this paper, we proposed a data mining based mechanism which is meant for leveraging cache replacement for efficient communications in MANET. We proposed an enhanced POC-based algorithm for ARM which can reduce computational cost and generate association rules faster. Due to innovative technologies and increased capacities of devices in MANET, it is an important research directive to explore the suitability of data mining techniques for correlation of data items and effective cache replacement.

Our contributions in this paper are as follows. We proposed an algorithm based on POC-tree for faster generation of association rules. The algorithm uses an underlying data structure which makes it cost effective. The algorithm is meant for extracting knowledge from data in order to correlate data items and take expert decisions on cache replacement. The remainder of the paper is structured as follows. Section 2 threw light into the literature on cache replacement techniques. Section 3 provides problem definition. Section 4 presents the proposed cache replacement approach with an algorithm. Section 5 provides results of experiments. Section 6 gives conclusions and recommendations for future enhancements.

2. Related Works

This section reviews literature on caching in MANET. Sheeja*et al*. [1] proposed a protocol for congestion avoidance. The node in the network used route cache for performance improvement. They achieved congestion less routing mechanism with NS2 simulations. Park and Jeong [2] proposed a caching strategy for spatial queries. It was done in a mobile network with location based cache maintenance strategies. With regard to privacy, they proposed an approach based on hierarchical tree. Caching and its maintenance procedure are defined to have effective spatial query processing. Chuang *et al*. [3] introduced a new caching scheme known as Regionally Maintained Cooperative Caching (RMCC). Due to high node mobility, the deterioration of cache hit ratio is handled with their proposed algorithm. They achieved lower latency and higher cache hit ratio. Yao *et al*. [4] proposed an anchor-based caching scheme named A-CACHE. However, it is meant for public key caching in large wireless networks. It makes use of anchor nodes with high cache memory and regular nodes also support caching of public keys for effective communications.

Han *et al*. [5] explored content delivery acceleration in wireless networks. Win *et al*. [6] proposed a cooperative caching approach known as Tightly Cooperative Caching Approach (TCCA). This scheme encourages all nodes to cooperate in caching mechanism well as they get credits by doing so. Priyadarshini*et al*. [7] proposed a cache consistency scheme for mobile networks. It is server-based consistency scheme. Here the cache management activities are under control of server. It has cache nodes and query directory nodes to facilitate caching and cache consistency. It focused on the data consistency between server and client cache. Rane and Dhore [8] proposed data replication strategies for efficient communications in wireless networks. They found that data replication provides advantages like making data available and reduce communication cost. Kohila*et al*. [9] explored data structures for frequent patterns mining for understanding behaviour of mobile users with respect to their movements. Thus it helped in searching for requested services efficiently.

Ting and Chang [10] proposed a cooperative caching scheme known as Group-based Caching Scheme (GCC). It was meant for improving latency and cache hit ratio performance in MANET. It was able to optimize and reduce communication cost and computational cost. Mandhare and Thool [11] employed distributed route cache update algorithm for improving Quality of Service (QoS) in MANET. Their cache update scheme works as part of Dynamic Source Routing (DSR) protocol. Routes are updated in cache from time to time for efficient routing of data. Saleh [12] proposed yet another cooperative caching scheme known as Adaptive Cooperative Caching Strategy (ACCS). It employs policies for pre-fetching and cache replacement. It divides MANET into number of clusters that are not overlapping. It collects routing information while forming clusters. They also studied stateless and tasteful behaviour of nodes and pre-fetching and cache replacement strategies.

Leu and Lin [13] proposed a cooperative caching scheme with energy efficiency. Their scheme considered residual energy of mobile nodes to make caching decisions. Based on the energy, the scheme determines which node participates in caching without compromising data availability. The scheme has improved performance in terms of data availability, response time, and lifespan of wireless network. Hallous*het al*. [14] used network coding approach for content distribution in multi-hop wireless networks. Islam and Shaikh [15] proposed a cache replacement scheme with correlation of data items. They used data mining approach with FP-growth algorithm for association rule mining to make heuristics. The heuristics are used for making cache replacement decisions.

Chand *et al*. [16] proposed a utility based cache management policy known as Least Utility Value (LUV) to improve data availability and promote cache hit ratio. It considers many factors such as data size, coherency, and the distance between the cache and the requester. Selvin and Palanichamy [17] also studied cooperative caching and proposed push-pull cache consistency method in MANET. The purpose of their method was to improve rate of serving to questions in the network. It is a hybrid scheme in which both client and server nodes are involved. It improves latency performance and reduces communication overhead. Kuppusamy*et al*. [18] proposed a cache replacement strategy in MANET with cooperative caching. In order to locate replaceable data, they employed Mometic Algorithm (MA) which is an evolutionary algorithm. It improved packet delivery ratio (PDR), reduced latency and control overhead. Jabas [19] studied association rule mining (ARM) of data generated in MANETs. As MANETs are growing in size and capabilities in information dissemination, ARM can help in finding trends in the data for making effective decisions.

Lipasti [20] studied various cache replacement policies and the memory used for caching mechanisms. Since cache memory has finite size, they studied the means of replacing it with new data and when to replace it. They also explored metrics like cache miss and cache hit ratios for understanding the performance of networks with caching enabled. Their work also includes optimal replacement policies and various techniques like Least Recently Used (LRU), Most Recently Used (MRU) and other approaches. Rani and Seetha [21] made a review of various cache replacement methods and performance dynamics related to cache management in MANET. They illustrated mobility and topology changes and the need for effective cache management.

It is understood from the literature that man cache replacement techniques came into existence. The size of MANET is also growing from time to time. As the MANET devices are deployed in massive scale, there is need for processing huge amount of data. This can be done with data mining techniques. Especially association rule mining can help in understanding trends in the data to make well informed decisions. This is the motivation behind this paper which explores a novel data mining approach that uses a dynamic and cost-effective data structure to achieve heuristics to determine cache replacement decisions.

3. Cache Replacement Algorithms

There are many cache replacement techniques that are available. This section provides different mechanisms, their merits and demerits.

3.1 Least Recently Used (LRU)

This mechanism is based on the notion that the data that is used more number of times is likely to be used more in future also. In the same fashion, the data that is not used for long time is likely not used in future also. When cache is full in LRU approach, the data which is not used for long time is removed from cache. It is the algorithm that is widely in use with regard to cache replacement. Cache maintains the data which is most recently used. When data is used, that is moved to the recently use list. An important drawback of the LRU approach is that it does not consider the size of data. Since size is an important factor in wireless networks, it is considered a limitation of LRU. However, LRU is simple to be adapted and suitable for many workloads. It exploits

recency features but cannot exploit frequency features of workload. Even when there are pages that are frequently requests, more gaps in temporal distance also cannot allow LRU to take advantage of those pages.

3.2 Least Frequently Used (LFU)
Unlike LRU, it is frequency based approach where an object or data is replaced based on the number of references that object has. Therefore, every object in cache needs to maintain a count of references. It makes use of frequency property and assumes that frequently requested data may be requested again in future. This feature is known as temporal locality. The LFU approach cannot adapt when there are different access patterns. Another problem with it is that it may maintain stale state reflecting high frequency counts. Therefore this mechanism is not much useful.

3.3 Last Recently Frequently Used (LRFU)
This mechanism maintains a value associated with every block of data. This value is known as Combined Recency and Frequency. This count can tell the likelihood of the block to be referenced in near future also. The computation is done with the combination of frequency and recency in order to generate a single parameter for decision making. There is correlation between the two ingredients. However, the correlation cannot be generalized as it shows differently for different workloads.

3.4 FUZZY Algorithm
Fuzzy logic is also useful to deal with cache replacement problem. It promotes the concept of probability or partial truth which specifies to which extent a given proposition is true. It supports expression of everything in the form of 0 and 1 in general. However, fuzzy logic can get a value between true and false as well. Therefore the degree of truth can replace Boolean truth. Thus fuzzy inference systems (FIS) came into existence. FIS can have input output and processing stages. The recency of reference and frequency of reference are mapped in the input stage with true values. The processing is used for generating result and the output stage generates an output value.

3.5 Adaptive Replacement Cache (ARC)
It is self-tuning cache which exhibits low overhead and it can respond to access patterns that are dynamic in nature. Moreover, it is able to balance between the features of workload such as recency and frequency. It also gets rid of pre-tuning pertaining to workloads. It can be used online for real time workloads. This method can adapt to the dynamics and variability of workloads over time. The workloads associated with this can have I/Os for long sequences with different frequencies and temporal locality. It is simple to implement and independent of size of cache.

3.6 First In First Out (FIFO)
It is the algorithm that mimics FIFO queue. The cache is maintained and evicted as per the FIFO policy. It evicts the blocks accessed in the first in first out fashion. It does not consider how many times or how often the data or blocks are accessed.

3.7 Time Aware Least Recently Used (TLRU)
This is a variant of LRU where cached data items do have specific life time. It is suitable for applications related to network caching. Applications like Content Delivery Network (CDN) and Information-Centric Network (ICN) are the examples where TLRU is best used. TLRU mechanism is associated with new terms such as Time to Use (TTU). This is the timestamp which helps in having more control for regulating cache memory. It is associated with a cache node where TTU value is computed and updated. Locally defined function is used to compute TTU. The algorithm sees that content with small life and less popular is replaced with new content.

3.8 Most Recently Used (MRU)
It is one of the replacement algorithms widely used. It results in more hits when compared with LRU as it has tendency to retain old data as well. This algorithm is very useful in situations that contain older items that are likely to get accessed again.

3.9 Random Replacement (RR)
This algorithm randomly selects a data item known as candidate item and gets rid of it to make room whenever required. It does not maintain any access history. It is simple and used processors where ARM is involved.

4. Problem Definition

MANETs are widely used in different real world applications. They came into existence as emergency ad hoc networks but later

on their utility is increased and they became ubiquitous. This is the reason there is large scale deployment of MANET devices in real world applications. In such cases, the network traffic is more. Due to the resource constrained nature of the network, there is need for minimizing the usage of resources associated with the network. Towards this end, the workloads that need to be transferred are to be done efficiently. When data is transferred from source to destination, there are possibilities to reuse the data items that already exist in nodes. Such data items need not be transferred so as to realize the resource consumption. Towards this end caching concept came into existence. Caching is the phenomenon which has high speed memory to store frequently used data items. When caching is enabled in MANET, requests from nodes can be served easily with cached data. If caching is snot there, it needs to reach a remote node and there is more resource consumption. This is the motivation behind this research. Many existing systems are studied and understood that the large scale deployment of MANET devices need more efficient algorithms. Thus data mining algorithms can help in processing huge amount of data. In this paper, we proposed an algorithm to generate association rules from the data to have data item correlation quickly and make well informed cache replacement decisions.

5. Proposed Methodology

This section presents our methodology used for cache replacement. It is based on data mining. The overview of the methodology is shown in Figure 1. As the workloads are processed in MANET, the transaction database contains historical data that is mined by the proposed association rule mining algorithm. There is query manager module that contains query processing capabilities. Query processor can hit the cache for faster data transfer in MANET. It takes care of queries and processes them by preferring cache usage. When data is not present in cache, it needs to follow the conventional approach that takes more time and resources. This causes overhead. The high speed cache memory is meant for reducing latency time. Knowledge database is the repository of data that is updated with historical processing results. This DB can help in data mining process. The proposed association rule mining has underlying data structure which is light weight and causes less overhead. The cache module in the system takes care of caching and providing data which is required by nodes while transmission of data. The cache replacement is made based on the result of data mining algorithm. The algorithm takes transactional database and support value as input and generates association rules.

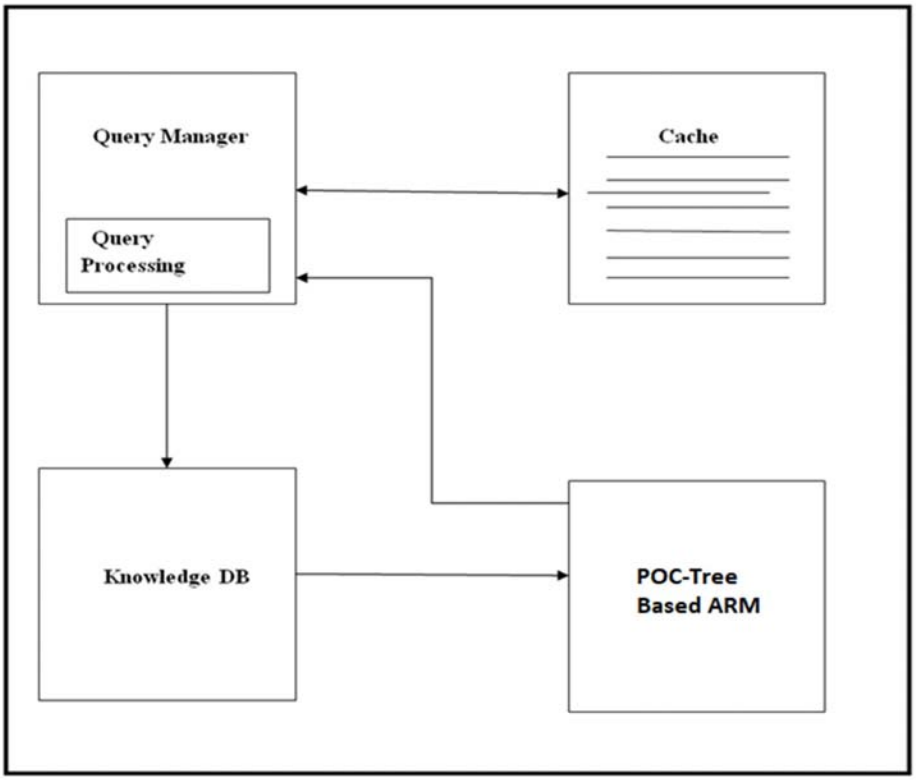

Figure 1. Methodology for cache replacement

The proposed algorithm can help in reducing latency and increasing various performance indicators. POC tree based data mining algorithm for association rule mining is used to have higher efficiency. The algorithm reduces traditional steps and makes it a two process procedure. In the first process, it generates frequent item sets and in the second process it writes association rules. The overview of sample data is presented in Table 1.

ID	Items	Ordered Frequent Items
1	a, c, g, f	c, f, a
2	e, a, c, b	b, c, e, a
3	e, c, b, i	b, c, e
4	b, f, h	b, f
5	b, f, e, c, d	b, c, e, f

Table 1. Sample data

The data present in Table 1 is used to generate POC tree which appears as in Figure 2. The POC tree is essentially a tree which supports pre-order traversing and efficient in mining data. The tree has 0 values in its root. In each node of the array the data item and its support value is associated. This makes the traversal and the process of retrieving frequent item sets easier.

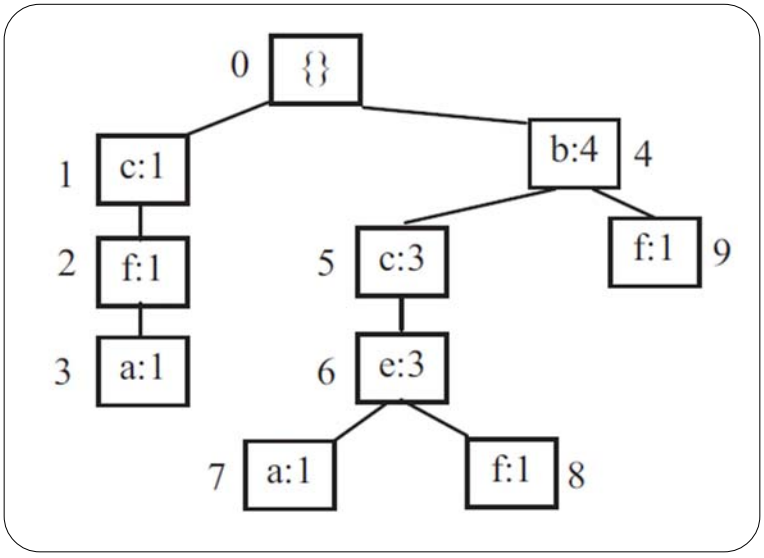

Figure 2. Generated POC-tree based on the data of Table 1

As presented in Figure 2, it is evident that the given data is represented in POC tree. This tree data structure supports faster generation of frequent item sets and thus association rules are mined efficiently. When compared with conventional tree approaches, enhanced POC based approach is much faster. The POC-tree based ARM algorithm exploits the POC tree for quick generation of association rules.

5.1 POC-Tree Based ARM

The proposed algorithm works on the data presented in POC tree. Tree is constructed based on given dataset dynamically. Once data is loaded into POC tree, it supports pre-order traversing to reach different nodes in the tree and generate frequent item sets. Once frequent item sets are generated, they can be used to filter them and write association rules to a text file. The association rules help in making well informed decisions in the proposed cache replacement strategy. The algorithm is based on association rule mining as presented in section 5.

Algorithm: POC-Tree Based Association Rule Mining Algorithm

```
Input    :  Transactional DB, minimum support
Output   :  Association rules
1     Construct POC tree based on DB
2     Obtain candidate frequent item sets from POC tree
3     Generate final frequent item sets
4     Filter item sets using support
5     Generate association rules
Return rules
```

5.2 Association Rule Mining

Let $I = \{i_1, i_2, ..., i_n\}$ are binary attributes and they are known as items. Let $D = \{t_1, t_2, ..., t_n\}$ are a set of tansactions present in database. Every transaction is identified with unique ID. Then an implication rule can be formed among the items as follow.

$X => Y$, where $X, Y \subseteq I$.

According to Agrawal, a rule is defined as $X => i_j$ for i_j □ I.

A rule is made up of two set of items known as X and Y where X is known as antecedent and Y is known as consequent. Two statistical measures are used to improve quality of association rules. They are known as support and confidence.

Support

This measure refers to the frequency of iteam set that appears in given dataset. With respect to T, the support of X is defined as follows.

$$supp\ (x) = \frac{|\{t \in T; X \subseteq t\}|}{|T|} \quad (1)$$

Confidence

Confidence on the other hand refers to how often the association is found in the database is true. The confidence of a rule $X => Y$ with regard to set of transactions is computed as follows.

$$Conf\ (X => Y) = supp\ (X\ U\ Y) / supp\ (X) \quad (2)$$

6. Results

Experiments are made using NS2 simulations. The proposed algorithm is evaluated and compared against other existing data mining algorithms used in the research of caching in MANET. The performance of the proposed algorithm is compared with PrePost and FP-Growth algorithms.

As shown in Figure 3, it is evident that the number of nodes in the network and query delay time in seconds. There are two trends found in the results. There is gradual increase in the query delay when number of nodes is increased. Another trend is that query delay of the proposed method is less than other two algorithms. FP-Growth has shown better performance than PrePost. The results revealed that the proposed approach is able to reduce delay due to its efficient cache replacement ability and data item correlation.

As shown in Figure 4, it is evident that the number of nodes in the network and update delay time measured in seconds. There are two trends found in the results. There is gradual increase in the update delay when number of nodes is increased. Another

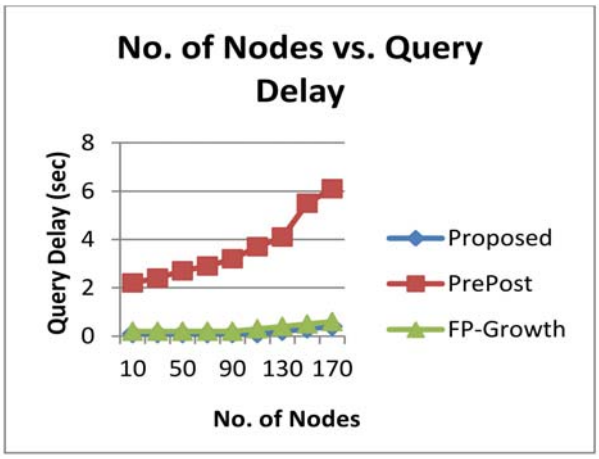

Figure 3. Query delay performance comparison

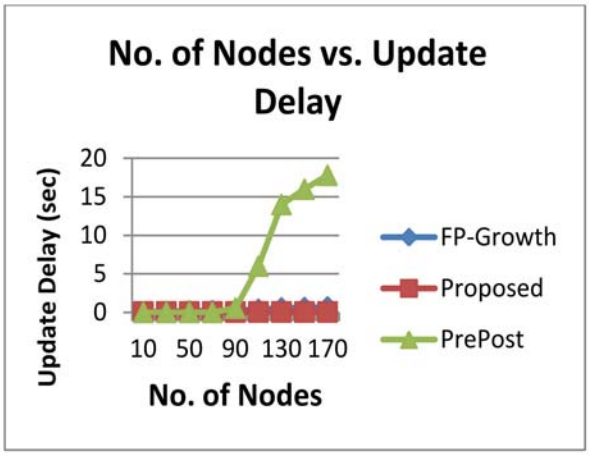

Figure 4. Query delay performance comparison

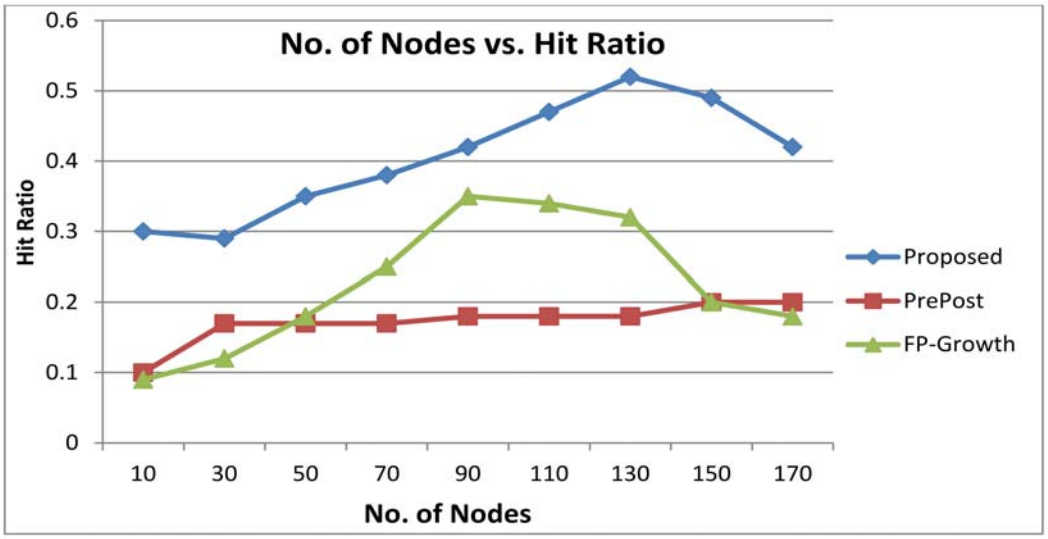

Figure 5. Hit ratio performance comparison

trend is that update delay of the proposed method is less than other two algorithms. FP-Growth has shown better performance than PrePost.

As presented in Figure 5, it is evident that observations are made with different number of nodes such as 10, 30, 50, 70, 90, 110, 130, 150, and 170. There are some important observations in the results presented. The Hit ratio is increased when number of nodes is increased. However, beyond 90 nodes showed FP-Growth to decline in hit ratio. In case of the proposed method, the drop in hit ratio occurred at 150 and 170 nodes. In case of PrePost, there is gradual increase in the hit ratio. Nevertheless, the proposed method showed highest hit rate for all number of nodes used in experiments.

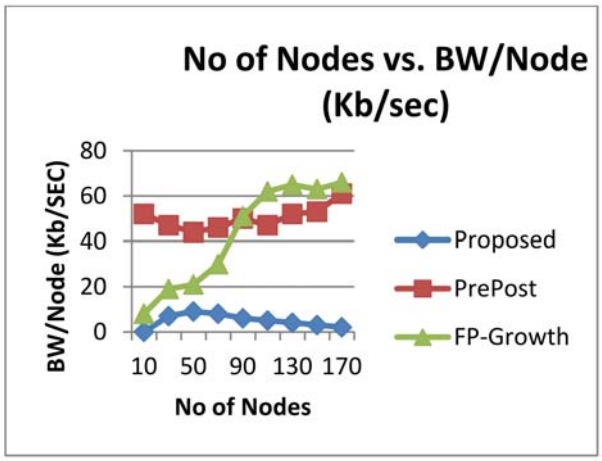

Figure 6. Bandwidth usage comparison

As shown in Figure 6, the result of the three algorithms is presented to have bandwidth comparison. Number of nodes is gradually increased and observations are recorded. Bandwidth usage of proposed method is low with all number of nodes in the experiments when compared with other methods. In other words, the proposed method outperforms the other two methods. When number of nodes is increased, the bandwidth usage is gradually increased in case of FP-Growth while it is almost the same with PrePost as well. In case of the proposed method the bandwidth usage is even decreased from 70 nodes through 170 nodes.

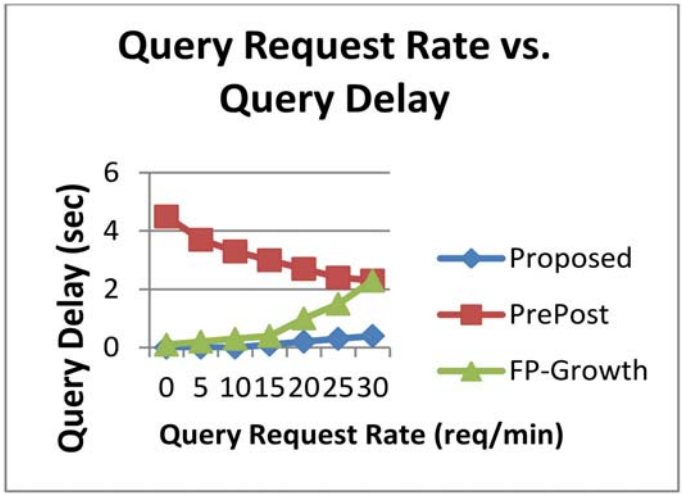

Figure 7. Query delay comparison against query request rate

As presented in Figure 7, it is evident that there is query delay observed against query request rate. Two algorithms have shown similar trends while the other algorithm named PrePost showed different trend. Query delay in case of proposed method is gradually increasing when number of requests per minute is increased. FP-Growth also showed increase gradually when query

request rate is increased. In case of PrePost, the query delay time is more initially and reduced when number of queries per minute is increased. Nevertheless, the proposed method outperforms the other two approaches.

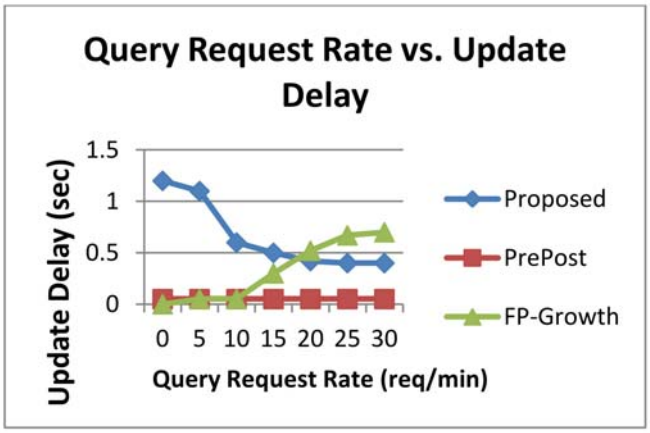

Figure 8. Update delay against query request rate

As shown in Figure 8, it is evident that the update delay is presented against different query request rate. It is observed that PrePost showed better performance in this case. The proposed approach showed more update delay initially and reduced as number of requests per minute is reduced. In case of FP-Growth, the update delay is less initially and it is increased as number of queries per request is increased. With respect to the proposed system the query request rate has its influence on the update delay.

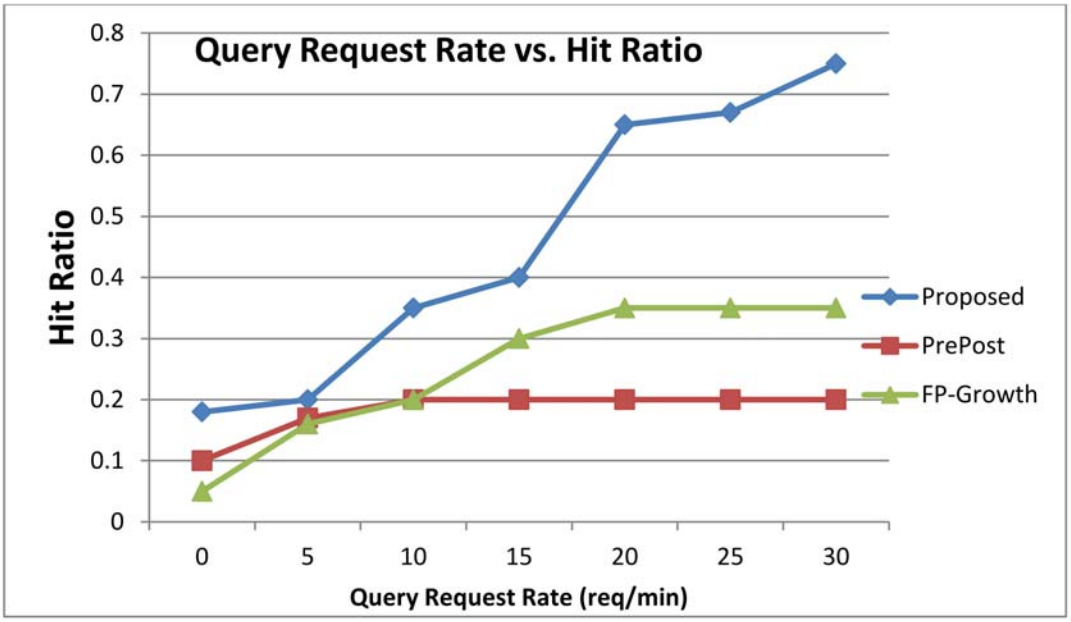

Figure 9. Hit ratio against query request rate

As shown in Figure 9, it is evident that hit ratio of all algorithms is observed with the three algorithms. As the number of requests per minute is increased from 0 through 30, it observed that the hit ratio is increased. The hit ratio is therefore influenced by the query request rate. The proposed approach showed highest hit ratio for all query request rates. PrePost algorithm showed least hit ratio performance while FP-Growth is better than that. Nevertheless, the proposed algorithm outperforms other algorithms.

As presented in Figure 10, it is evident that bandwidth usage dynamics of different algorithms are compared. Bandwidth usage

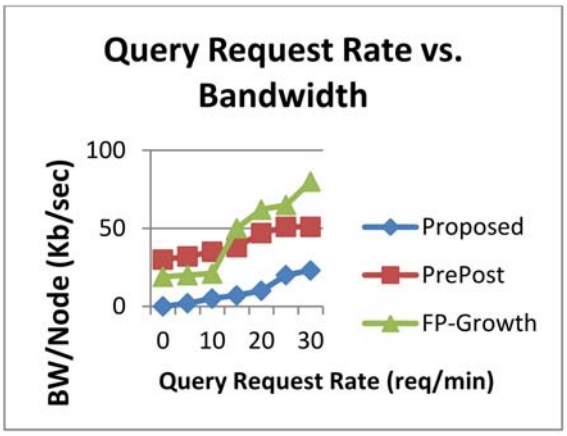

Figure 10. Bandwidth comparison against query request rate

is gradually increased when query request rate is increased. However, the performance of the proposed system is high as it consumes least bandwidth for all query request rates used in the experiments. FP-Growth utilized less bandwidth up to 10 query request rates but showed increased bandwidth usage from query request rate 15 onwards. Nevertheless, the proposed system outperforms the other two algorithms.

7. Conclusions and Future Work

In this paper, we proposed a methodology for improving efficiency in data transfer in MANET by using optimized cache replacement approach. Since caching plays vital role in reducing bandwidth usage and increase in speed of communications in wireless networks, we focused on the cache replacement technique. Cache replacement technique is meant for finding ideal factors that can help in making cache replacement decision. The bottom line of the research is to aim at achieving higher throughput and less resource utilization. The main resource considered is bandwidth usage. A data mining approach is followed for cache replacement. We proposed an enhanced POC based data structure and algorithm for association rule mining in order to have better intelligence on the cache replacement decisions. Nodesets is the data structure used in the proposed method for discovery association rules. The proposed method is capable of generating association rules that are used in discovering heuristics for helping ideal cache replacement. We made NS2 simulations to demonstrate proof of the concept. We compared the proposed approach with existing algorithms. Our simulation results found the proposed method outperforms existing ones. In future we intend to improve cache replacement by investigating possibility of a hybrid approach that makes use of ARM and other techniques towards increasing cash hit ratio further.

References

[1] Sheeja, S., Ramachandra., Pujeri, V. (2013). Effective Congestion Avoidance Scheme for Mobile Ad Hoc Networks. *I. J. Computer Network and Information Security*. 1, 33-40.

[2] Park, Kwangjin., Jeong, Young-sik. (2014). A Caching Strategy for Spatial Queries in Mobile Networks. *Journal of Information Science and Engineering*. 30, P1187-1207.

[3] Chuang, Po-Jen., Chen, Yu-Yiu., Chen, Hang-Li. (2013). RMCC: An Efficient Cooperative Caching Scheme for Mobile Ad-hoc Networks. *International Journal of Future Generation Communication and Networking*. 6 (3) 1-13.

[4] LinYaoa, JingDengb, JieWanga, GuoweiWu. (2015). A-CACHE:Ananchor-based public key caching scheme in large wireless networks. *Computer Networks*. 87, 78–88.

[5] Han, Tao., NirwanAnsari., Wu, Mingquan., Yu, Heather. (2013). On Accelerating Content Delivery in Mobile Networks. *IEEE Communications Surveys & Tutorials*. 15 (3) 1-20.

[6] Nwe Nwe Htay Win, Bao Jianmin, Cui Gang, Dalaijargal Purevsuren. (2014). Tightly Cooperative Caching Approach in Mobile Ad Hoc Network. *WSEAS TRANSACTIONS on COMPUTERS*. 13, 1-11.

[7] Chindrella Priyadharshini, T., Neelakandan, S., Swarnalatha, M. (2014). An Efficient Cache Consistency Scheme in Mobile Networks. *International Journal of Scientific & Engineering Research*. 5 (5) 1-6.

[8] DhirajRane, M. P. Dhore. (2016). Overview of Data Replication Strategies in Various Mobile Environment. *IOSR Journal of Computer Engineering*, 1-7.

[9] Kohila, M. J., Ganeshkumar, V. (2015). Data Structures for Frequent Patterns Mining in Mobile Service Environment. *NNGT Int. J. on Networking and Computing*. 2, 1-6.

[10] I-Wei Ting, Chang, Yeim-Kuan. (2013). Improved group-based cooperative caching scheme for mobile ad hoc networks. *J. Parallel Distrib. Comput*. 73, 595–607.

[11] Mandhare, V. V., Thool, R. C. (2016). Improving QoS of Mobile Ad-hoc Network using Cache Update Scheme in Dynamic Source Routing Protocol. *Procedia Computer Science*. 79, 692 – 699.

[12] Ahmed, I., Saleh. (2017). An Adaptive Cooperative Caching Strategy (ACCS) for Mobile Ad hoc Networks. *ACM*, P1-51.

[13] YunghoLeu, Hsin-Chang Lin., Lee, Chi-Chung. (2013). An Energy Conserving Cooperative Caching Policy for Ad Hoc Networks. *Academy Publisher*, 1-6.

[14] Halloush, Rami., Liu, Hang., Dong, Lijun., Wu, Mingquan., HayderRadha. (2017). Hop-by-hop Content Distribution with Network Coding in Multihop Wireless Networks. *Digital Communications and Networks*. 3, 47–54.

[15] Islam, Noman., Zubair., Shaikh, A. (2016). Exploiting Correlation Among Data Items for Cache Replacement in Ad-hoc Networks. *ACM*, 1-6.

[16] NarottamChanda, R. C., Joshib., ManojMisrab. (2007). Cooperative caching in mobile ad hoc networks based on data utility. *Mobile Information Systems*. 3, 19–37.

[17] Lilly SheebaSELVIN., Yogesh PALANICHAMY. (2016). Push-pull cache consistency mechanism for cooperative caching in mobile ad hoc environments. *Turkish Journal of Electrical Engineering & Computer Sciences*. 24, 1-13.

[18] P. Kuppusamy, B. Kalaavathi and S. Chandra. (2014). Optimal Data Replacement Technique for Cooperative Caching in Manet. *ICTACT Journal on Communication Technology*. 5 (3) 1-6.

[19] Ahmad Jabas. (2016). MANET Mining: Mining Association Rules. *ACM*, 1-27.

[20] Mikko., Lipasti, H. (2016). Cache Replacement Policies. *University of Wisconsin-Madison*. 1-25.

[21] Rani, Sudha., Y. J., Seetha, M. (2014). Caching Mechanisms in MANET for Data Availability and Performance Improvement: A Review. *International Journal of Electrical Electronics & Computer Science Engineering*. 1 (6) 1-7.