http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (4): 648-655, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.648.655

Efficient Data Replication and Consistency Maintenance Scheme for MANETs

[1]N. Sabiyath Fatima and [2]P. Sheik Abdul Khader
[1]Department of Computer Science and Engineering, B.S.A. Crescent Engineering College, Chennai, India
[2]Department of Computer Applications, B.S.A. Crescent Engineering College, Chennai, India

Abstract: Now-a-days many challenges are encountered in the deployment of caching in an ad-hoc network. The traditional data caching and path caching techniques do not produce good results because of node mobility. If a new route is found and if its cache is full, then it must look for another route even if it is valid, hence a powerful replacement technique is necessary. Even if a node has utmost memory to cache information about all other nodes, it may become invalid in a short period of time due to its mobility. To eliminate the above problems, effective replacement schemes are provided along with an efficient caching technique. The main objective of this study was to upgrade the caching efficiency in a mobile network environment. The cache has been made consistent and efficient. Cache memory must contain reliable data. This study proposed a Complete Cache Management (CCM) scheme which includes enhancing the cache efficiency, data replication and cache replacement. The contribution of the work is twofold. First, it provides efficient path caching by making use of Path Prefetching (PPF) algorithm. Secondly, it provides better cache consistency by Hit Count (HC) based cache replacement scheme. Simulations are conducted using ns-2 to study the performance of the system in terms of average delay and hit ratio and finally it is compared with the performance of other caching schemes for MANETs.

Key words: Path Prefetching, mobile ad hoc network, complete cache management, hit count, data replication, cache replacement

INTRODUCTION

Nodes of an ad-hoc network are mobile with similar transmission power, bandwidth and computation capabilities. In general, the source node caches routes, hence a route is made available when an application within the same node, demands for it. This is called source route caching (Liang and Haas, 2003). Also discussed the intermediate route caching. On demand routing protocols like Ad hoc On Demand Routing Protocol (AODV) and Dynamic Source Routing Protocol (DSR), allow an intermediate node which has a cached route to the destination and sends reply to the source with the cached route.

This study stated two different algorithms namely Path Prefetching (PPF) and Hit Count (HC) replacement scheme. The Path Prefetching Algorithm prefetches same path from its neighbor for future use. In order to prefetch the path the neighbor must assemble in one hop distance from the source and it should store the path up to the TTL (time to live) value expires. This method of storing the path in the nearby node will increase the cache hit ratio, because mobile nodes will move far away from the actual source node after T sec which results in maximum utilization of the cache spaces. In Hit Count based cache replacement scheme, the data with the maximum Hit Count value should be retained in the cache in order to replace the unpopular data (Artail *et al*., 2008).

Hara and Madria (2006) states that access frequency of each data should be taken into account for making data replication as a effective method for efficient data accessibility. Casetti *et al*. (2002) proposed a Random Walk Dissemination (RWD) strategy where each and every copy of the information roams around the network by moving from a node to a one-hop neighbor that is selected with equal probability. This article considers Cache and forward mechanism as one of the solutions. Papapetrou *et al*. (2005) stated that the routing protocols accesses caches by means of a search or lookup for a specific node. The bloom filters provide summaries of cache content for testing cache membership. Cao (2003) described that the effective method to enhance the performance of data access in MANETs is to make use of Caching techniques. Nuggehalli *et al*. (2003) introduced two caching techniques. The first method makes use of a centralized server to deliver information to

Corresponding Author: N. Sabiyath Fatima, Department of Computer Science and Engineering, B.S.A. Crescent Engineering College, Chennai, India

the network nodes. The second method places a copy of the server content into all network nodes.

Panchal and Abu-Ghazaleh (2004) showed that caching have undesirable side effects like inefficiencies due to stale paths and use of low quality paths even when shorter paths are available. The purpose of Neighbor Caching (NC) (Shobha and Rajanikanth, 2009) is for utilization of the cache space of inactive neighbors for caching tasks. Cao (2002) described Invalidation Report (IR) based cache management where the server broadcasts an invalidation report in a periodic way. Here the changed data items are indicated. Valera *et al.* (2003) follows cooperative packet caching and shortest multi-path routing in order to reduce packet loss. Artail *et al.* (2008) described a caching system that minimizes delay and maximizes the likelihood of finding data that is cached in the ad hoc network, This system relies on the indexing of cached queries. Panchal and Abu-Ghazaleh (2004) showed that caching have undesirable side effects like inefficiencies due to stale paths and use of low quality paths even when shorter paths are available.

Joyce *et al.* (2009) proposed that the validity of a cached item could be verified by a mobile client by comparing the last update time and its Absolute Validity Interval. Jung *et al.* (2005) have stated that the idea behind node caching is that the nodes that are involved in recent data packet forwarding have got more reliable information about its neighbors and they have better locations than other nodes. Artail *et al.* (2008) describes Cooperative and Adaptive Caching System which is for creating a cooperative caching system that minimizes delay and maximizes the likelihood of finding data that is cached in the ad hoc network.

Yang and Chu (2006) stated that DSR using Reclaim- Based Caching increase the usage of cache and decrease the occurrence of broadcasting because of temporarily broken link during node mobility. The concept behind this approach is that the broken routes are capable of repairing themselves. Hu and Johnson (2002) reduced cache staleness by preventing the re-learning of stale knowledge of a link after hearing that the link has broken. In TTL based consistency maintenance (Tang *et al.*, 2007) the internet helps in the speeding up of content distribution by keeping the replicas consistent and up-to-date. Wang (2009) proposed a method that finds the optimal path which reduces energy consumption of mobile nodes and increases the life-span of network. Hu and Johnson (2000) discussed the analysis of effects of different design choices for caching on demand routing protocols in wireless ad hoc networks.

Cao *et al.* (2004) stated that data-path caching scheme is also a kind of effective scheme to reduce data request delays without causing a large number of data clones. In Cache path, mobile nodes cache the data path and use it to redirect future requests to the nearby node which has the data instead of the faraway data center. In Cache path, a node need not record the path information of all passing data. Rather, it only records the data path when it's closer to the caching node than the data source. By the time a cached path is needed, it might be invalid (due to node movement). Using a stale path can add significant delay until the path is discovered broken. If many paths in the cache are stale, several stale paths may need to be followed before a valid path is found (or a route request is initiated).

Du and Gupta (2005) talk about COOP to improve data availability. Here cache management uses inter-category and intra-category rules to minimize cache duplications The limitation of this scheme is that flooding incurs high discovery overhead and do not take into account size and consistency for replacement.

Ramaswamy and Liu (2004) stated that the Cache replacement algorithms reduce the response time by selecting a subset of documents for caching so that a given performance metric is maximized. At the same time the cache must take extra steps to guarantee some form of consistency in the cached documents. Cache consistency algorithms enforce appropriate guarantees about the staleness of the cached documents. Lin *et al.* (2003) stated that most of today's caching proxies employ the Least Recently Used (LRU) algorithm or the Least Frequently Used (LFU) algorithm or some variant of LRU or LFU as their cache replacement policy. Proxy caches that employ LRU or one of its variants need to maintain the last hit timestamp for each document they cache. Similarly, proxy caches that choose to use LFU or one of its variants need to maintain the total number of hits experienced by every cached document. An expiration-Age based document placement scheme is also introduced in web caching. This scheme reduces the replication of documents across the cache group while ensuring that a copy of the document always resides in a cache where it is likely to stay for a longest time.

Gitzenis and Bombos (2002) and Fan *et al.* (1999) stated that the existing Prefetching approaches make use of uplink and battery energy in order to improve cache hit ratio and access latency. A Prefetching scheme that takes into account power and bandwidth efficiency is considered for data dissemination environments. Due to node mobility the data in caches become stale. Cao (2002) stated that mobile devices calculate the Prefetch Access Ratio (PAR).

Feeney and Nilsson (2001) stated that Proactive Prefetching schemes are not energy efficient. According to Shen *et al.* (2004) a passive prefetching scheme is used for cache management of mobile devices in multicast and broadcast communications. In passive prefetching, a mobile device do not send uplink request which in turn reduces the burden on the server and improves cache performance.

According to Artail and Mershed (2009) message forwarding algorithm is based on the concept of selecting the neighbor node from a set of designated nodes. The algorithm uses routing information to select the node with shortest distance. The goal is to reduce the average number of hops taken to reach the destination node that holds the desired data. Yang *et al.* (2012) addressed the problem of reliable data delivery in MANET. A novel position based opportunistic routing protocol was used to efficiently maintain uninterrupted communication. When a node sends or forwards a packet the neighbor nodes that have overheard the transmission will serve as forwarding candidates and forwards the packet if it is not relayed by the specific best forwarder. To handle the communication voids a virtual destination based void handling scheme is used to enhance robustness of POR. High packet delivery ratio is achieved while delay and duplication are reduced.

MATERIALS AND METHODS

The mobile nodes in the wireless environment suffer from scarce bandwidth, low-quality communication, frequent network disconnections and limited local resources. Hence, the CCM scheme is introduced with two different algorithms to effectively utilize the cache space. PPF algorithm deals with efficient prefetching which fetches some data path for the future requests. The HC based replacement scheme is proposed for cache replacement which is based on hit count value.

Path prefetching algorithm: A prefetch buffer stores data which are prefetched from the main memory. This reduces the interference between the prefetched data and the demand-fetched data in cache. Since, all the prefetched data streamed into a prefetch buffer, the working set of the demand-fetched data in cache will not be disturbed by data prefetching. Prefetch buffer is found in the instruction cache design but not in data caches. Hence, the path caching is performed on the prefetching scenario. PPF integrated with DSR: The simulation model of DSR are according to nodes storing route information in a path cache. DSR is an on demand routing protocol that allows the caching of multiple routes for a single destination with nodes adding a complete source/destination paths. Cache is designed with a limited capacity.

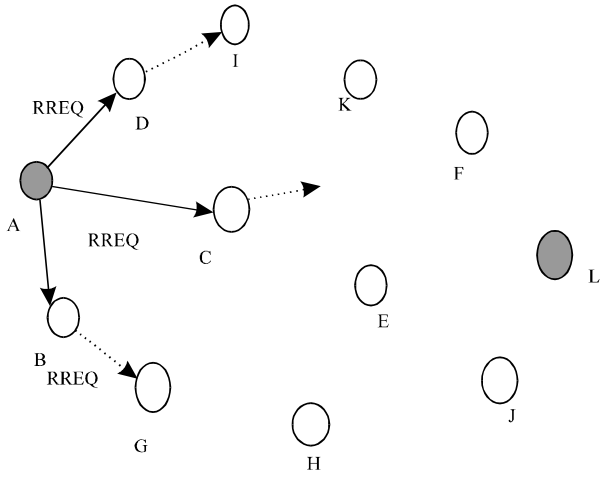

Fig. 1: Source Broadcasting RREQ

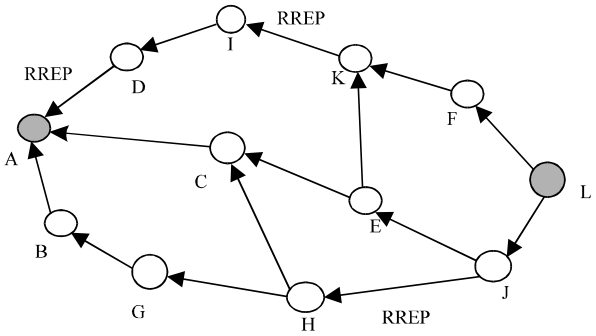

Fig. 2: Source getting RREP from different path

Hence, entries are timed out after 300 sec (TTL value). The shortest path is broadcasted to the one-hop neighbor. This happens only when the destination is so far from the source node (when the number of hops is large). When intermediate nodes receive this request they rebroadcast it if they have not already received it. The path from source to destination node is stored in the route request header with nodes appending their own address before rebroadcasting the packet.

Figure 1 and 2 shows a DSR algorithm with RREQ and RREP packets. The source node A sends the RREQ to the neighbor nodes such as D, C, B to find the destination path. In Fig. 2, the RREP packet is received from different path. DSR algorithm chooses a shortest path among these in order to send data and other discovered paths are saved for future use. The Fig. 3 shows the data transmission in the shortest path selected by the Source node A i.e. A-C-E-J-L.

The Fig. 4 shows that the shortest path is broadcasted to the one-hop neighbors of A i.e. D, C, B to prefetch the path. The number of hop in the discovered

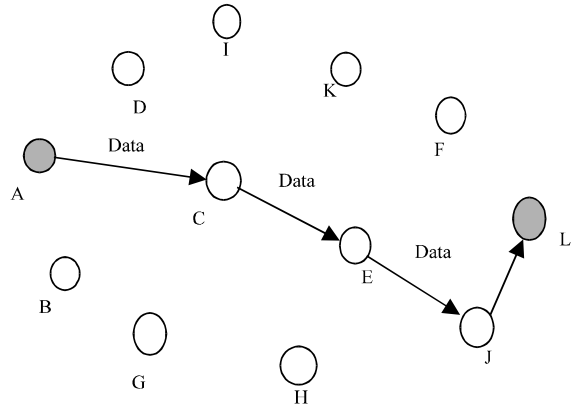

Fig. 3: Sending data in a shortest path

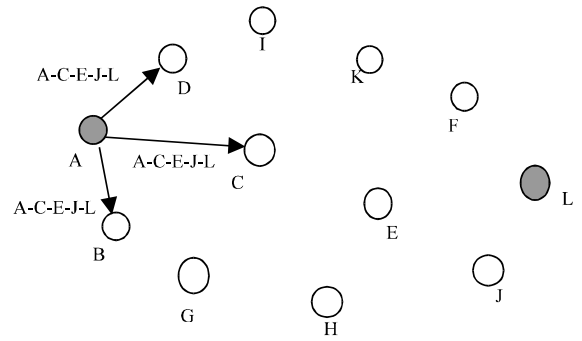

Fig. 4: Broadcasting the current path to one-hop neighbors

path is greater than the threshold value. Due to node mobility, nodes move far away from the current position after a T sec. Hence replicating the path to the nearby node provides a greater chance of increasing the efficiency of caching and cache hit ratio drastically.

Algorithm 1:

N_h : Number of hop in the path;
T_h : Threshold hop count
Step 1: Received a new DSR packet
Step 2: Source send RREQ packet
If node IP address matches
- sent Route Reply;
Else If RREQ table already has an entry with this
Request ID and from the same source;
Fetches the path from PATH CACHE;
ELSE
Rebroadcasts the RREQ packet until reaches destination;
END If;
Step 3: Destination sends RREP message
Step 4: Source saves all the paths for future use
Step 5: If N_h in the current path> T_h
Step 6: Send a copy of current path to the one-hop neighbors;
Step 7: If no path currently available
Send RERR packet to the source;

Algorithm analysis: In PPF, the source node saves all the paths which are discovered in the route discovery phase of DSR in its local cache. Then it sends this current route information to one hop neighbor if the number of hop in the cached path is greater than the threshold value. If the number of hop is very less, it will create unwanted traffic in the network, also instead of getting the replicated data the requester can get the data from the original source. Instead in order to overcome the above problem the source node will broadcast the path to one-hop neighbor only when the destination is far away from the source. The saved path will get deleted if the TTL value expires. This is explained in algorithm 1.

HC based replacement: A cache replacement policy is necessary when a client wants to cache a data item but the cache is full. Hence, a suitable subset of data items to be evicted from the cache. The data item size need not be fixed, the replacement policy handles data items of varying sizes. Cache hit ratio is the key factor for efficient cache replacement and hence it is called as "hit ratio based replacement". The replacement algorithm also considers cache consistency that is, data items that tends to be inconsistent earlier should be replaced earlier.

Algorithm 2:

```
1:  B: Buffer size
2:  HC: Hit Count value
3:  LC: Least hit count valued segment
4:  N: Number of current segments in the cache
5:  Segment (n): segment n arrives at time t
6:  Segment (l) = least HC valued segment at t
7:  Free buffer space (FB) = B-N
8:  If (FB > = Segment (n))
9:          {
10:         Accept (n)
11.         }
12: Else
13:         {
14:         Hit count look-up in cache table
15:                 If (LC = Segment (l))
16:                         {
17:                 Remove Segment (l) from the cache
18:                 }
19: //repeat the steps 8 to 18 until the arrived data stored in the cache
```

Algorithm analysis: The HC replacement algorithm checks for the free buffer space. If the free space is enough to hold the data it accepts otherwise it removes the least hit count valued segment (LC) from the cache in order to save newly arrived segment l. This process is continued until the new segment gets enough space in the cache memory. This is given in algorithm 2.

RESULTS AND DISCUSSIONS

Performance evaluation: The experiment is conducted to evaluate the cache efficiency by taking into account parameters like delay, hit ratio, hop count, traffic rate and link failure rate. The performance of a traditional caching

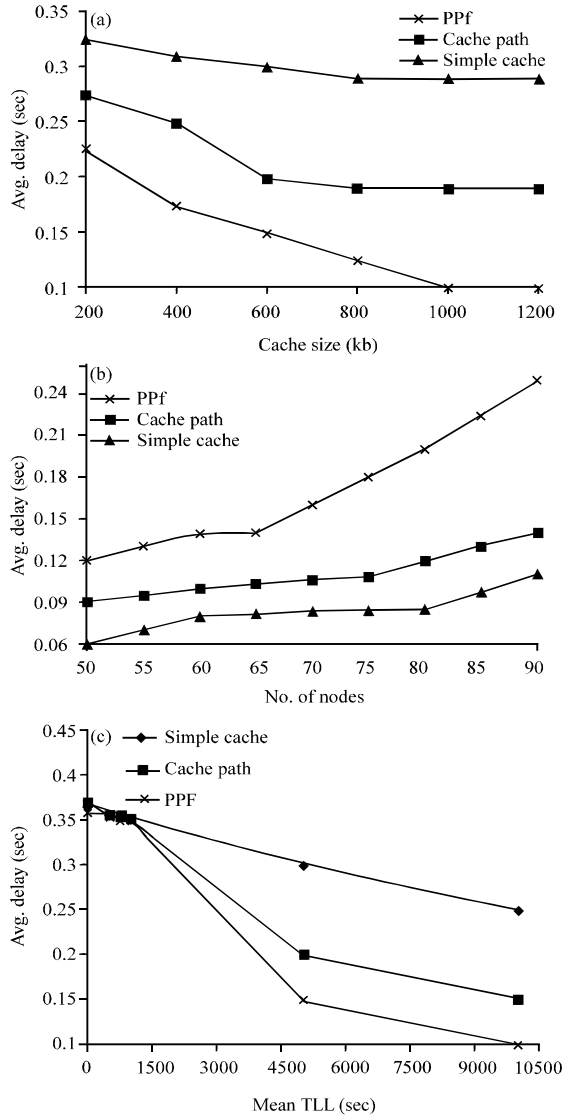

Fig. 5(a-c): Performance of (a) Cache size vs. average delay (b) No. of nodes vs. average delay and (c) Mean TTL vs. average delay

Table 1: The Simulation Parameters

Parameter	Default value
Size	700×400
Query interval	50 sec
Item number	500
Threshold HC	10
Data size	128 bytes
Uplink bandwidth	19.2 kbps
Downlink bandwidth	144 kbps
Speed	(0-5) m sec^{-1}
Data packets	20
Mean interval time	30 sec
Packet size	512 bytes
Wireless bandwidth	2 Mbps
Cache size	75
No of nodes	100
Mobility model	Random way point model

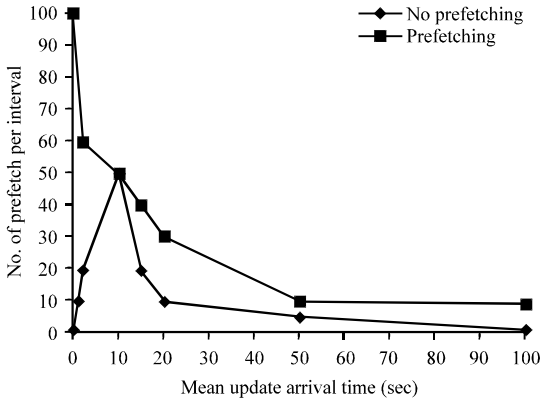

Fig. 6: Number of prefetch vs. update arrival time performance

scheme and proposed CCM schemes such as PPF, HC replacement scheme is evaluated using NS-2 simulator. The Simulation Parameters are given in Table 1.

Average delay: In the earlier path cache algorithms, updating data path result in cache misses. This problem is solved by asking the clients to prefetch the data that may be used in the near future. In this section, the performance of the PPF scheme is compared with the Simple Cache scheme and the Cache Path scheme in terms of the delay in seconds.

In Fig. 5a, when the cache size increases beyond 1000 kb all three algorithms produces less delay and maintains a constant level. In Simple Cache the maximum delay is 0.3 because the request and reply needs to travel a larger number of hops. Cache Path is an efficient method to reduce the number of hops travelled by the request and replies, it generates minimum of 0.2% delay. In case of PPF the delay is reduced to 0.1 when the cache size reaches 1000, the reason for the reduction is it is prefetching cached path. The PPF algorithm performs 10% better than the traditional CachePath algorithm. Figure 5b shows the delay variations with respect to number of nodes, the delay increases with the increase in number of nodes.

Figure 5c calculates the delay with respect to TTL value, when the TTL value is below 1000 sec all three algorithms behave in the same way i.e., initially produces more delay then decreases gradually.

Prefetching ratio: When looking into Fig. 6, the number of prefetches increases as the mean update arrival time decreases in the case of the non-prefetching approach.

In PPF algorithm, when the mean update arrival time decreases, more frequent data updating occurs

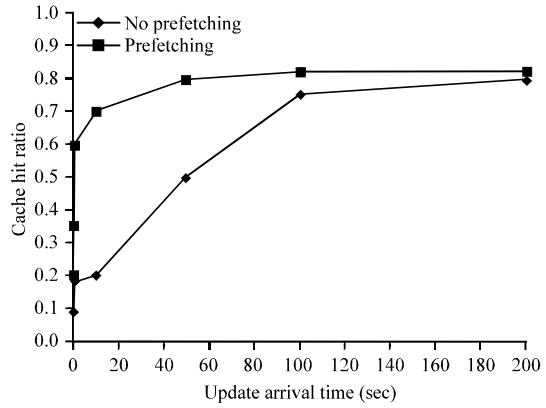

Fig. 7: Update arrival time vs. cache hit ratio graph

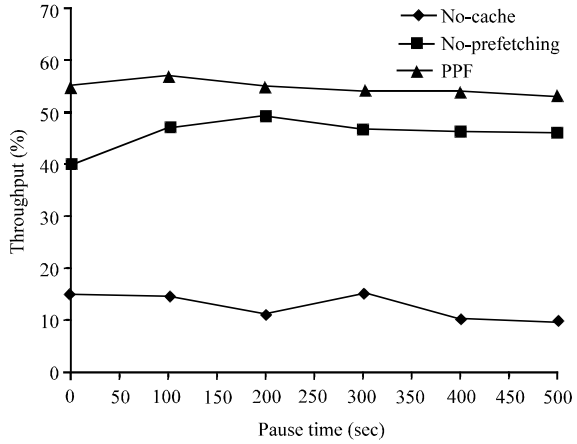

Fig. 8: Throughput vs. pause time graph

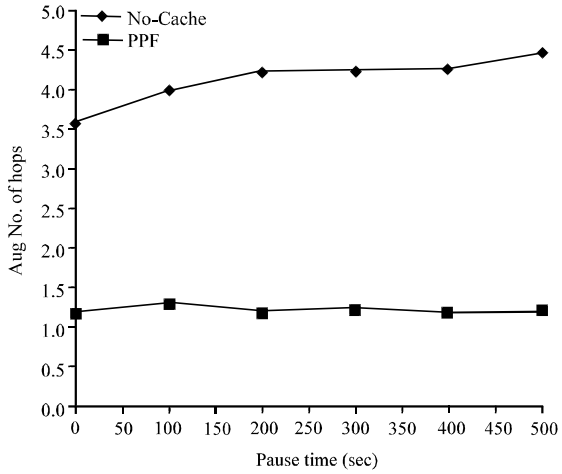

Fig. 9: Average number of hop vs. pause time

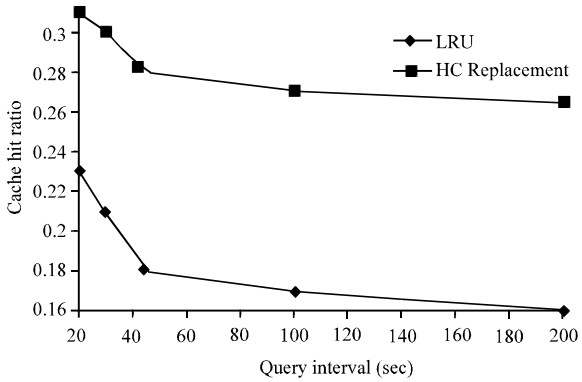

Fig. 10: Query interval vs. cache hit ratio

which causes more cache misses. For example PPF gives 100 prefetches to mean update arrival time of 0 sec but non-prefetching algorithms gives only 50 prefetches. So, PPF produces 50% improvement in the cache hit ratio. When the mean update arrival time is 10 sec algorithm with no prefetching effect produces 50 prefetches and prefetching produces 50 prefetches, the cache miss will be less during this period. It shows an increased cache hit ratio. The significance of this intersection is as mean update arrival time increases the number of prefetch per interval first shows an increased effect due to increased hop count then later due to cache miss the number of prefetches starts to decrease after 10 sec which is the maximum arrival time. Our proposed prefetching algorithm shows utmost greater number of prefetches at arrival time 0 sec then it starts decreasing due to decrease in hit ratio.

Cache hit ratio: Prefetching data in the local cache improves the cache hit ratio. When some data items are marked as non-prefetch, the cache hit ratio gets reduced. The Fig. 7 shows that when the mean update arrival time is high, there are not too many data updates and most of the queried data are served locally. When the mean update arrival time decreases, data is updated more frequently.

With prefetching, the cache hit ratio increases rapidly until it reaches the maximum threshold value. When the update arrival rate is 10, non prefetching algorithms provide 0.2% of cache hit ratio but PPF produces 2%.

Through put: The Fig. 8 compares the throughput of three algorithms such as, no-caching, no-prefetching and PPF algorithm. With caching, there is a high probability of the requested data being cached in the nodes local cache or at other nodes. In the no-caching scenario very low throughput is produced. i.e., maximum of 15%, No prefetching gives maximum of 50% throughput but PPF produces nearly 60% for 50 m sec pause time. Compared to no-prefetching algorithm PPF algorithm generates 10% higher throughput.

Figure 9 depicts the number of hops required to transfer data with respect to pause time. Throughput

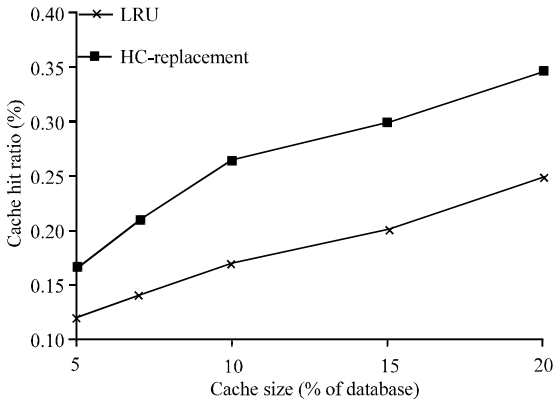

Fig. 11: Cache size vs. cache hit ratio graph

decreases when number of hop count increases. Because of prefetching, the nodes in the PPF environment gets the required data with minimum number of hop count. Compared to no-caching algorithm PPF algorithm produces 30% better performance.

Cache hit ratio: In the Fig. 10 shows the comparison between LRU and HC based replacement algorithm. When the query interval is increased, almost both the scheme gives a worse performance. The reason is for a longer query interval, the client makes more movements between two successive queries hence, the client has a lower probability of residing in one of the valid scopes of the previously queried data items when a new query is issued.

For the query interval 20 sec, LRU gives 0.23% hit ratio but HC based scheme gives 0.31%. When the query interval increases to 200 then also LRU gives only 0.16% but the proposed scheme provides 0.26%.

Effect of cache size: When cache size is varied from 5-20% the performance of the replacement schemes improves with increasing cache sizes. From the Fig. 11, when the cache size is 5% of the database LRU provides 0.11% of hit ratio but HC policy gives 0.16%. When the cache size increases to 20, LRU gives 0.23% and HC based replacement gives 0.35%. This result shows that, the proposed scheme outperforms the existing policies even for the different cache sizes.

CONCLUSION

In this study, a Complete Cache Management scheme is designed and evaluated to provide efficient data access in ad hoc networks. The existing methods cache data, cache path and hybrid caching has some drawbacks like low hit ratio and high delay. Our solution PPF is designed to provide prefetching facility to the mobile node which increases the throughput by storing one-hop neighbor information. Hit count is one of the important parameter to find out the popular data in the network, this idea have been utilized in the HC based replacement policy to provide efficient replacement. The simulation results showed that the proposed algorithm outperforms the existing systems in terms of delay, hit ratio, network traffic, link failure rate and throughput. Simulation results evolved that the proposed algorithm reduces the query delay by a factor of 2 and doubles the throughput when compared to the traditional algorithms. Though, most of the previous research in MANET focuses on routing the upcoming work are concentrating on data access. This work will further stimulate and pave way for future research on prefetching and data replication to maintain cache consistency. The future practitioners are expected to compare cache based data access with data replication techniques.

REFERENCES

Artail, H. and K. Mershed, 2009. MDPF: Minimum distance packet forwarding for search applications in mobile ad hoc networks. IEEE Trans. Mobile Comput., 8: 1412-1426.

Artail, H., H. Safa, K. Mershad, Z. Abou-Atme and N. Sulieman, 2008. COACS: A cooperative and adaptive caching system for MANETs. IEEE Trans. Mobile Comput., 7: 961-977.

Cao, G., 2002. Adaptive power aware cache management for mobile computing systems. IEEE Trans. Comput., 51: 608-621.

Cao, G., 2003. A scalable low-latency cache invalidation strategy for mobile environments. IEEE Trans. Knowledge Data Eng., 15: 1251-1265.

Cao, G., L. Yin and C. R. Das, 2004. Cooperative cache based data access in ad hoc networks. IEEE Comput. Soc., 37: 32-39.

Casetti, C., C.F. Chiasserini, M. Fiore, C.A. La and P. Michiardi, 2002. P2P cache and forward mechanisms for mobile ad hoc networks. Proceedings of the IEEE Symposium on Computers and Communications, July 5-8, 2009, Sousse, Tunisia, pp: 386-392.

Du, Y. and S. Gupta, 2005. COOP-A cooperative caching service in MANETS. Proceedings of the Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services, October 23-28, 2005, Papeete, Tahiti, pp: 58-63.

Fan, L., P. Cao, W. Lin and Q. Jacobson, 1999. Web prefetching between low-bandwidth clients and proxies: Potential and performance. Proceedings of the International Conference on Measurement and Modeling of Computer Systems, May 1-4, 1999, Atlanta, GA, USA, pp: 178-187.

Feeney, L.M. and M. Nilsson, 2001. Investigating the energy consumption of a wireless network interface in an Ad Hoc networking environment Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies, April 22-26, 2001, Anchorage, AK., USA., pp: 1548-1557.

Gitzenis, S. and N. Bambos, 2002. Power-controlled data prefetching/caching in wireless packet networks. Proceedings of the IEEE 21st Annual Joint Conference of the IEEE Computer and Communications Societies, June 23-27, 2002, New York, USA., pp: 1405-1414.

Hara, T. and S.K. Madria, 2006. Data replication for improving data accessibility in ad hoc networks. IEEE Trans. Mobile Comput., 5: 1515-1532.

Hu, Y.C. and D.B. Johnson, 2002. Ensuring cache freshness in on--demand ad hoc network routing protocols. Proceedings of the 2nd ACM International Workshop on Principles of Mobile Computing, October 30-31, 2002, Toulouse, France, pp: 25-30.

Hu, Y.C. and D.B. Johnson, 2000. Caching strategies in on-demand routing protocols for wireless ad hoc networks. Proceedings of the 6th Annual International Conference on Mobile Computing And Networking, Mobicom, August 6-11, 2000, Boston, MA., USA., pp: 1-12.

Joyce, M., J.C. Pamila and K, Thanushkodi, 2009. Performance analysis of improved cache invalidation scheme in mobile computing environment. Int. J. Comput. Sci. Network Security, 9: 153-160.

Jung, S., N. Hundewale and A. Zelikovsky, 2005. Node caching enhancement of reactive ad hoc routing protocols. Proceedings of the IEEE Conference on Wireless Communications and Networking, Volume 4, March 13-17, 2005, New Orleans, LA., USA., pp: 1970-1975.

Liang, B. and Z.J. Haas, 2003. Optimizing route -cache lifetime in ad hoc networks. Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications, Volume 1 (AJCCC-2003), San Franciso, California, USA., pp: 281-291.

Lin, Y.B., W.R. Lai and J.J. Chen, 2003. Effects of cache mechanism on wireless data access. IEEE Trans. Wireless Commun., 2: 1247-1258.

Nuggehalli, P., V. Srinivasan and C.F. Chiasserini, 2003. Energy-efficient caching strategies in ad hoc wireless networks. Proceedings of the 14th ACM International Symposium on Mobile Ad-Hoc Network and Computing, June 1-3, 2003, Annapolis, MD., USA., pp: 25-34.

Panchal, N.I. and N.B. Abu-Ghazaleh, 2004. Active route cache optimization for *ad hoc* networks. Proceedings of 3rd International Conference on Networking, Gosier, Volume 1, December 2004, Guadeloupe, French Carribbean, pp: 604-611.

Papapetrou, E., E. Pitoura and K. Lillis, 2005. Speeding-up cache lookups in wireless ad-hoc routing using bloom filters. Proceedings of the IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications, Volume 3, September 11-14, 2005, Berlin, pp: 1419-1423.

Ramaswamy, L. and L. Liu, 2004. An expiration age-based document placement scheme for cooperative web caching. IEEE Trans. Knowledge Data Eng., 16: 585-600.

Shen, H., M. Kumar, S.K. Das and Z. Wang, 2004. Energy-efficient caching and prefetching with data consistency in mobile distributed systems. Proceedings of the 18th International Parallel and Distributed Processing Symposium, Volume 1, April 26-30, 2004, Santa Fe, New Mexico, USA., pp: 67-67.

Shobha, K.R. and K. Rajanikanth, 2009. Intelligent caching in on-demand routing protocol for mobile ad hoc networks. World Acad. Sci. Eng. Technol., 32: 413-420.

Tang, X., H. Chi and S.T. Chanson, 2007. Optimal replica placement under TTL-based consistency. IEEE Trans. Parallel Distribut. Syst., 18: 351-363.

Valera, A., K.G.W. Seah and S.V. Rao, 2003. Cooperative packet caching and shortest multi-path routing in mobile ad hoc networks. Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications, Volume 1, March 30-April 3, 2003, San Franciso, CA., USA., pp: 260-269.

Wang, Y., 2009. Energy control based routing for mobile ad hoc network. Proceedings of the International Symposium on Intelligent Information Systems and Applications, October 28-30, 2009, Qingdao, China, pp: 59-62.

Yang, S., C.K. Yeo and B.S. Lee, 2012. Towards reliable data delivery for highly dynamic mobile ad hoc networks. IEEE Trans. Mobile Comput., 11: 111-124.

Yang, S.J. and S.C. Chu, 2006. Performance analysis of DSR usingreclaim-based caching policy on the MANET. Proceedings of the 14th IEEE International Conference on Networks, ICON, Volume 2, September 2006, Singapore, pp: 1-6.