Performance Analysis of Cache Resolution Policies for Data Discovery

Devendra Kumar Jain
Research Scholar
Department of Computer Applications
MANIT Bhopal, India

Sanjay Sharma
Professor
Department of Computer Applications
MANIT Bhopal, India

ABSTRACT
Most of previous research in Mobile Ad Hoc Networks (MANETs) concentrated on the development and enhancement of neighbor discovery protocols and routing protocols, which could efficiently discover reliable route between two communication mobile nodes. Data caching addressed two basic issues: cache resolution and cache management. In this paper, we give a review of cache resolution scheme and performance analysis of these schemes using ns2.32.

Keywords
Mobile Ad Hoc Network, Caching, Cooperative Caching, Data Caching, Cross-layer

1. INTRODUCTION
Mobile Ad Hoc Networks (MANETs) is special type of wireless network in which collection of mobile network interfaces may forms a temporary network with-out the aid of any established infrastructure or centralized administration. MANETs have dynamic topologies; mobile nodes can change position, join and leave the entirely. In this situation, gives a number of problem likes message loss, unpredictable disconnection of mobile devices and network partitioning [1] [2]. The absence of any fixed infrastructure makes MANETs very attractive for military operation, rescue operation, sensor network, and time-critical applications [2] [3].

Most of the research effort focuses on neighbor discovery protocol and network layer solution (routing protocol) to increases reliable connectivity among mobile nodes in a constantly varying a topology. Although routing is an important issue in mobile ad hoc networks, Routing protocols for MANETs have classified in to three categories: first one is proactive, second one is reactive and third one is combination of both (Hybrid). Different routing protocols try to solve a different problem in different cases. Proactive that means route is always available for data item or predefine such as OLSR [4] and DSDV [5], reactive that means a route is discover when mobile node need a data item to other node such as AODV [6] and DSR [7], and hybrid that means a combination of both routing protocol such as ZRP [8] and OPHMR [9].

The ultimate goal of these networks is to data availability among mobile nodes. In mobile ad hoc networks due to frequent network division or network partition, data availability is lower compared to traditional wired networks. Cooperative caching and data replication offer a solution to this problem. The main difference between caching and replication can be stated as the caching occurs after the retrieval of data item, while replication of a data item can occur before a request received for that item [10] [11]. Various issues and a survey solution associated with the data replication and cooperative caching for mobile ad hoc networks are given in [12] [13] [11].

In addition, caching techniques are developed for improve data accessibility. With caching, the data access delay is reduces since data requests can be served from the cached client before reaching the server. Caching has been successfully applied in design of CPUs, multi-processors, and routers. It has also been employed in internet based technologies, such as the cache design of the World Wide Web (WWW) [14], proxy caches [15], Internet Cache Protocols [16] for proxy servers, cache digests [17] and summary caches [18], various distributed system, with the purpose of reducing data access delays. Furthermore, several caching clients can also work together to form a cooperative caching environment, further improving the overall caching performance.

In this paper, review the cache resolution schemes for resolve a data item request with minimal cost of time, energy, and bandwidth and simulation analysis of these scheme

The rest of this paper is organized as follow. In section 2, we present cooperative caching in MANETs. In section 3, survey existing cache resolution approach. Section 4, will give an overview of our implementation and environment used to simulate and detail of simulation result. Finally in section 5 concludes the paper.

2. COOPERATIVE CACHING IN MANETs
Caching is copy a portion of data that is called data item from the data provider or server to a smaller and faster storage device (Cache) interposed between the data consumer or client or mobile nodes and the data provider and server, so that in future, data access can resolved from the cache with low cost. Caching can decrease the access latency of such request and also decrease the resources needed to access data which is important in mobile device that have greatly reduce resources than traditional network devices. Cooperating caching based on the idea of sharing and coordination of cached data among multiple mobile nodes or client can be particularly effective for data access in mobile ad hoc networks. However, limited cache space, movement of nodes and frequent disconnection limit availability of data. By cooperative caching frequently accessed data item in the MANETs, we can improve data accessing rate and data availability.

In cooperative caching based data access architecture, mobile nodes cache the data or cache the path or hybrid to reduce latency and to increase data availability. A good cooperating cache management technique should have proper cache discovery, cache admission control, cache replacement, and cache consistency scheme.

3. CACHE RESOLUTION SCHEMES

Cache resolution addresses how to resolve a data item request with minimal cost of time, energy, and bandwidth. In cooperative caching, the emphasis of cache resolution is to answer how nodes can help each other in resolving a data request to improve the success ratio.

In No Cache resolution approach, that is, requested get only server.

In SimpleCache resolution approach [19] [20], is similar to the client cache of traditional client server model in wireless network. When data request is generated, the requested client first searches its local cache to check if the requested data item is cached (cache hit) or not (cache miss). If there is cache hit, the requested data item can be served locally. Otherwise, the data request sent to the server and wait for a response. The clients in SimpleCache do not use the cache space available in their neighbors or any other clients.

In Hop-by-Hop resolution approach [19] [20], a data request is checked at every hop in local cache along the forwarding path to server, if forwarding node has the requested data the request get resolved before reaching the server. In MANETs, Throughput and reliability heavily depend on the number of hops (Distance) covered by the connection. As the number of hop increases, the throughput and overhead decreases.

4. SIMULATION EVALUATION

This section presents the simulation results of the proposed scheme that is compared to No Cache, SimpleCache, and Hop-by-Hop caching scheme, which is similar to data cache scheme proposed in [19] [20]. Specifically, the comparisons are No Cache, SimpleCache, and Hop-by-Hop caching.

The following metrics are used to evaluate the proposed scheme from the perspectives of data availability, energy efficiency, and time efficiency.

Success Ratio: The percentage of successfully resolved requests before reaching the server. This is used to reflect the resulted data availability.

Miss Ratio: The percentage of requests that have to be forwarded to the server for resolution.

Average Messages sent per successful request: The average number of sent messages for resolving a data request. This is used to reflect the message overhead and energy efficiency of cooperative caching scheme.

Average Travel Distance: The average travel distance in hops for a request to get reply. This is used to reflect the delay for data retrieval from cached client to requested client.

4.1.Simulation Model and Evaluation

The experiments are conducted on Network Simulator (NS-2.32 [21] on a fedora platform) with the CMU wireless and mobility extension. IEEE 802.11 protocol is used as the basic in the MAC layer. The well-known Ad hoc On-demand Distance Vector (AODV) [22] routing protocol is used as the underlying routing algorithm. In the simulation environment, we assume that the server has fixed at position 200*800 in the network area. In the network, there are 1000 original data items of fixed size of 20 KB that are held by server. When the server is receives a data request from requested client, the first-come-first-served policy is applied.

The cache size of each client is set to 200-600 KB. In other words, a client can cache 1.0-3.0% of the data items. In the simulation, each pair of requested client and server is selected randomly. A client is randomly selected as the query client whose query rate is set to 0.2 to 1 queries/second. The query pattern of the data object is based on a zipf-like distribution [23]. In the zipf-like distribution, which has been frequently used to model a non-uniform distribution, the access probability of the i^{th} ($0 \le i \le n$) popular data item represented as follow. The normalized mathematical representation is given as:

$$P_i = \frac{1}{i^\theta \sum_{k=1}^{n} \frac{1}{K^\theta}}, \qquad \text{where } 0 \le \Theta \le 1$$

In this formula, the parameter n is the total number of data items, and the parameter Θ indicate how skew is distribution. The larger the value of Θ, the more concentration of requests put on popular data items. The value of Θ is varies for different applications. For web page access, it is in the range of [0.64 to 0.83] according to studies in [24]. The setting of important simulation parameters is present in table1.

Table 1: Simulation setup parameters

Parameters	Default Value	Range
Simulation Area	1000*1000	
Server Position	200*800	
Number of clients	50	
Speed (m/s)	4	
Pause Time (s)	40	
Transmission Range (Meter)	250	
Zipf-Distribution (Θ)	0.4	0.5-1.0
Total Number of data item	1000	
TTL	10	
Client Cache Size	200KB	200-600KB
Query Rate	0.6	0.2-1.0
Hello Interval	1s	
Allowed hello loss	3s	
Server	1	
Routing	AODV	
Replacement	LRU	

4.2.Result

4.2.1Effect of Cache Size

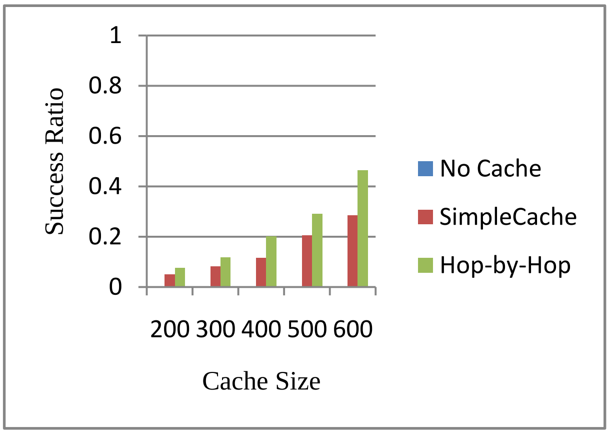

Figure 1: Plot of success ratio with different cache size.

Figures 1, 2, 3, and 4 shows the effects of the cache size on success ratio, miss ratio, average number of messages per request, and average travel distance in hops by varying the client cache size from 100KB to 500KB. From these figures, we can see that as the client cache size increases, the miss ratio, average number of messages per request, and average travel distance decrease for all the schemes and increase success ratio for all the schemes. The reason is that the increases the client cache size makes it possible to accommodate more data items in local cache. For different client cache sizes we have tried, the Hop-by-Hop strategy performs better than No Cache and SimpleCache.

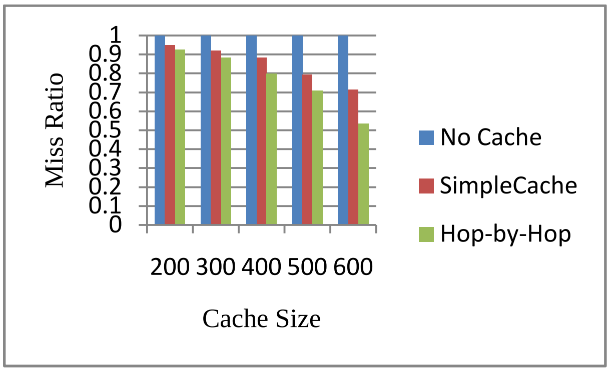

Figure 2: Plot of miss ratio with different cache size.

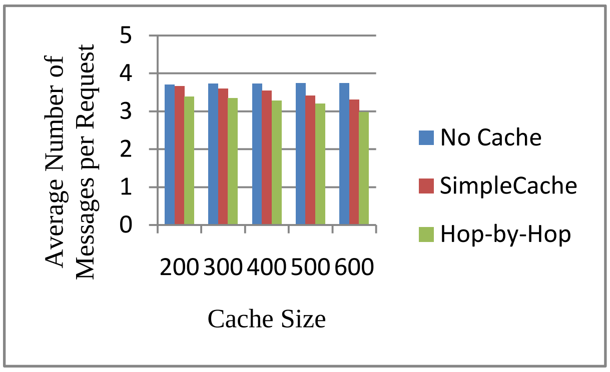

Figure 3: Plot of average number of messages per request with different cache size.

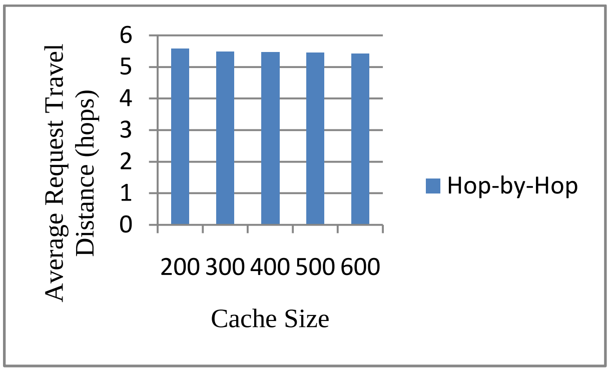

Figure 4: Plot of average request travel distance with different cache size.

4.2.2 Effect of Alpha

Figures 5, 6, 7, and 8 shows the effects of the alpha on success ratio, miss ratio, average number of messages per request, and average travel distance in hops by varying the Θ from 0.5 to 1.0. An increasing of Θ makes the requests more concentrated on a smaller percentage of data items thus improving the miss ratio and success ratio, and reducing the average travel distance. The Hop-by-Hop strategy performs better than No Cache and Simple Cache with respect to the miss ratio, success ratio, and average number of messages per request for all the Θ values simulated.

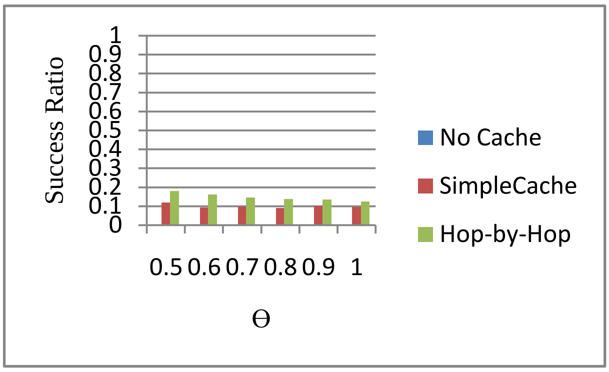

Figure 5: Plot of success ratio with different Θ.

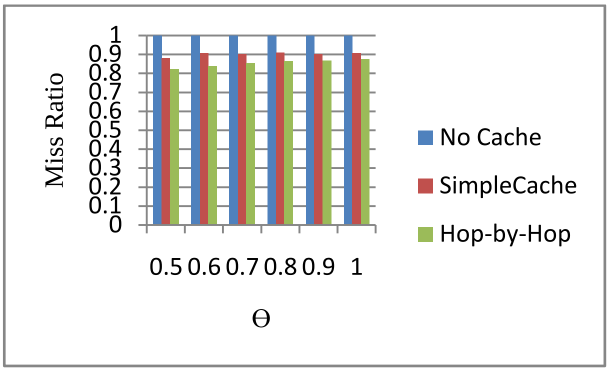

Figure 6: Plot of miss ratio with different Θ.

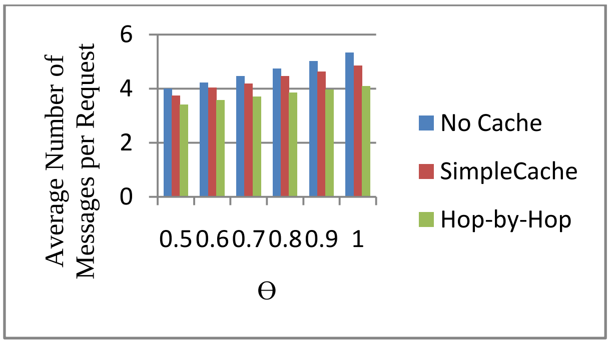

Figure 7: Plot of average number of messages per request with different Θ.

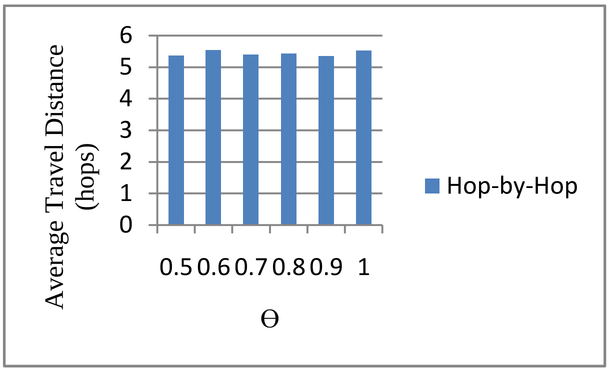

Figure 8: Plot of average request travel distance with different Θ.

4.2.3 Effect of Query Rate

Figures 9, 10, 11, and 12 shows the effects of the query rate on success ratio, miss ratio, average number of messages per request, and average travel distance in hops by varying query rate from 0.2 to 1.0. An increasing in the query rate implies a significant decrease the success ratio, increase the average travel distance, and increases the average number of messages per request for all the strategy. The reason is that the increase the query rate makes it possible to data traffic is high. When the data traffic is high, the success ratio is decrease, the average travel distance is longer, and increases the average number of messages. From the below simulation result, the conclusion is that the Hop-by-Hop strategy outperform the other strategy from the perspectives of the success ratio, miss ratio, average number of messages and average travel distance.

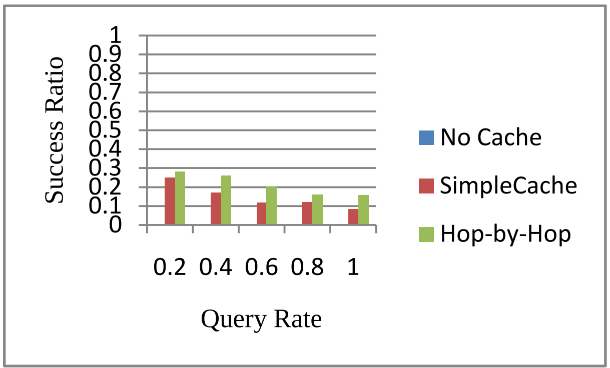

Figure 9: Plot of success ratio with different query rate.

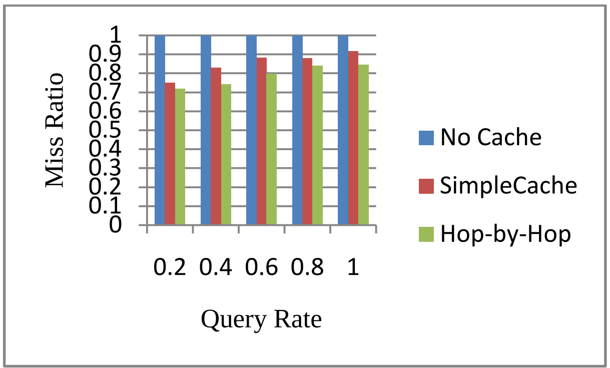

Figure 10: Plot of miss ratio with different query rate.

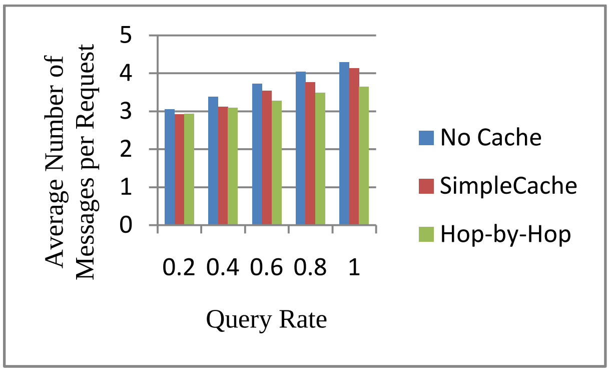

Figure 11: Plot of average number of messages per request with different query rate.

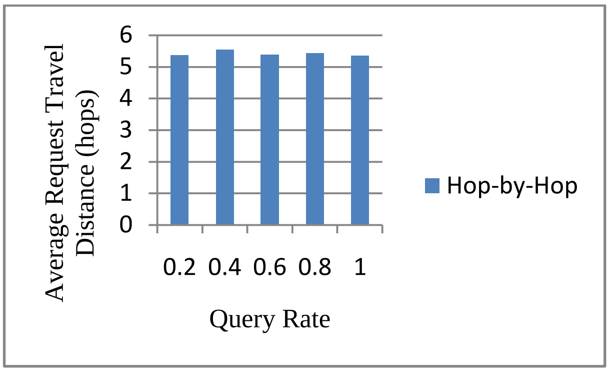

Figure 12: Plot of average request travel distance with different query rate.

5. CONCLUSIONS

This work is an attempt towards the comprehensive performance evaluation of No Cache, SimpleCache, and Hop-by-Hop Cache resolution strategy. In short, Hop-by-Hop cache resolution is better approach for data discovery, because a data request checked at every hop in local cache along the routing path to server.

6. REFERENCES

[1] M. Frodigh, P. Johansson, and L. Larsson, "Wireless Ad Hoc Networking – The Art Networking without a Network," Ericsson Review, No. 4, 2000.

[2] L. Fife and L. Gruenwald, "Research Issues for Data Communication in Mobile Ad Hoc Networks Data Base System," Proc. in International Conference on Management of Data, pp. 42-47, ACM SIGMOD 2003.

[3] X Hong, M. Gerla, G Pei, and C. Chiang, "A Group Mobility Model for Ad Hoc Wireless Networks," ACM/IEEE MSWiM, pp. 53-60, 1999.

[4] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum and L. Viennot, "Optimized Link State Routing Protocol for Ad Hoc Network," IEEE INMIC, pp. 62-68, 2001.

[5] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computer," ACM SIGCOMM, pp 234-244, 1994.

[6] C. Perkins and E. M. Royer, "Ad Hoc On-Demand Distance Vector Routing," IEEE Workshop on Mobile Computing Systems and Applications, pp. 90-100, 1999.

[7] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Network," Mobile Computing, pp. 158-181, 1996.

[8] Z. J. Haas and M. R. Pearlman, "The Zone Routing Protocol for Ad Hoc Networks," Internet Draft, draft-ietf-manet-zone-zrp-04.txt, 2002.

[9] B. Manouer, L. Chen, C. H. Foh and J. W. Tantra, "OPHMR: An Optimized Polymorphic Hybrid Multicast Routing for MANETs," IEEE Transaction on Mobile Computing, vol 6 issue 5, pp. 551-562, 2007.

[10] T. Hara, "Replica Allocation in Ad Hoc Networks with Periodic Data Update, "Int. Conf. on Mobile Data Management, 2002.

[11] Preetha Theresa joy and K. Poulose Jacob, "Cooperative Caching Techniques for Mobile Ad hoc Networks,"

Proc. in international conference on data science & engineering, ICDSE-2012.

[12] Prasanna Padmanabhan, Le Gruenwald, Anita Vallur, and Mohammed Atiquzzaman, "A Survey of Data Replication Techniques for Mobile Ad Hoc Network DataBases," The VLDB Journal, pp. 1143-1164, 2008.

[13] Abdelouahid Derhab and Nadjib Badache, "Data Replication Protocols for Mobile Ad-Hoc Networks: A Survey and Texonomy," IEEE Communication Surveys & Tutorials, vol. 11, No. 9, pp. 33-51, 2009.

[14] C. Aggarwal, J. Wolf, and P. Yu, "Caching on the World Wide Web," IEEE Transaction on Knowledge and Data Engineering, vol. 11, No. 1, .pp. 94-107, 1999.

[15] J. Shim, P. Scheuermann and R. Vingralek, "Proxy cache algorithms: design, implementation, and performance," IEEE Transaction on Knowledge and Data Engineering, vol. 11, No. 4, pp. 549-562, 1999.

[16] D. Wessels, and K. Claffy, "ICP and the Squid web cache," IEEE Journal on Selected Area of Communications, vol. 16, No. 3, pp. 345-357,1998.

[17] A. Rousskov and D. Wessels "Cache Digests. Computer Network and ISDN Systems," 30, pp. 2155-2168, 1998

[18] L. Fan, P. Cao, J. Almeida, and A. Z. Broder, "Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol," IEEE/ACM Transaction on Networking, vol. 8, No. 3, pp. 281-293, 2000.

[19] L. Yin and G. Cao, "Supporting Cooperative Caching in Ad Hoc Networks," IEEE Transaction On Mobile Computing, pp. 77-89, 2006.

[20] Yu Du, S. K. S. Gupta, and Georgios Varsamopoulos, "Improving On-Demand Data Access Efficiency in MANETs with Cooperative Caching," ELSEVIER Ad Hoc Networks, pp. 579-598, 2009.

[21] K. Fall and K. Varadhan (Eds.), "The NS Mannual", URL: http://www.isi.edu/nsnam/ns/doc/index.html

[22] C. Perkins and E.M. Royer, "Ad Hoc On-Demand Distance Vector Routing,", IEEE Workshop on Mobile Computing Systems and Applications, pp. 90-100, 1999.

[23] G. Zipf, "Human Behavior and the Principle of Least Effort," Addison Wesley, 1949.

[24] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," proc. of ACM/IEEE Mobicom, pp. 85-97, 1998.