]ANET 환경에서의 Low-overhead 를 위한 Lookup 기법

Khac Tiep Mai*, 추현승**
성균관대학교 정보통신공학부
e-mail : maikhctiep@skku.edu

Low-overhead Information Lookup in MANET

Khac Tiep Mai* and Hyunseung Choo*
School of Information and Communication Engineering Sungkyunkwan University

Abstract

This paper proposes a novel cluster-based lookup service in mobile ad hoc networks. By applying a self-adaptive multi-hop clustering, we avoid the empty region problem of grid system and omit the need of GPS device. The novelty of the proposed scheme is in its exploitation of hashing function for location querying to achieve low communication overhead. By varying different simulation parameters, we show that the proposed scheme is scalable and adaptive to many networks scenarios. Experiment results demonstrate that our scheme reduces the communication overhead significantly.

1. Introduction

In networks, lookup service is a very general and important problem in which a source node wants to query desired information. A significant research has been conducted in peer-to-peer system. However, there is a limited success of this topic in mobile ad hoc networks (MANET). The challenge is due to the infrastructure-less characteristic and mobility. Unlike normal IP networks which consist of many subnets, MANETs do not impose any hierarchical system by default. Furthermore, all nodes freely move around and constantly change the network topology. These characteristics affect all higher layer protocols including the lookup service.

Lookup service can support any type of information. In routing problem, a source node queries the path to a specific destination. In reactive routing protocols such as AODV and DSR, the source performs the route discovery process which is based on flooding mechanism to find the destination. Another application example of lookup service is the location information querying of mobile hosts. Then, a location-aided protocol is used to improve the performance and scalability of routing. Location information of mobile node may be designed as geographical coordination which is obtained by GPS device. On the other hand, in cluster topology or grid system, it can be the ID of the cluster or grid. Next, mapping between a node ID, or IP address and a hostname is also a specific application of lookup service. In sensor networks, it is also desired to obtain sensing value of a specific node or event by using lookup mechanism.

There are two major types of lookup service, reactive and proactive protocols. Reactive lookup service [1], [2] is one major location service scheme thanks to its simplicity and extremely low maintenance overhead. It is used in the route discovery process of well known routing protocols such as AODV, DSR [3]. Reactive lookup service does not rely on any hierarchical system and hence has no overhead of such system. However, because reactive scheme is based on flooding, the querying overhead is significant especially when the querying rate increases. To overcome such weakness, the performance of flooding mechanism has been improved by many researches.

In this paper, we compare our scheme with a reactive scheme which utilizes the expanding ring search to limit the scope of flooding by the price of increased querying delay time. The principal of expanding ring search lies in the time-to-live, TTL, control of flooding message. When a source floods a query message in the first round, it limits the TTL to the smallest value; Flooding only occurs in a small area. If the destination is within the area, the querying process terminates successfully with low overhead. Otherwise, the source increases the TTL value and repeats the flooding with the bigger area. The process continues until the desired information is found. With this enhanced flooding, communication overhead is reduced while the delay time is increased. Still, later experiments demonstrate that the communication overhead of reactive scheme is high.

In the second category, proactive protocol, the destination information is stored in one or more servers and the source queries the nearest server to obtain the information. KCLS [4], k-hop cluster-based location service, is a proactive scheme which exploits a single level multi-hop cluster structure. k-hop value is a multi-hop clustering parameter which defines the maximum distance between cluster-head and their members. In this work, the location of mobile node, the current cluster ID of that node, is the lookup information. KCLS stores the location information of every node in all cluster-heads. A source only sends the query message to its current cluster-head to obtain the destination location. As a result, both the querying latency and querying overheads are extremely small. On the other hand, the information update overhead is high because each node needs to notify all cluster-heads. This overhead becomes bigger as update frequency of the information increases.

A special case of proactive scheme is home-region location service in which each mobile node is associated with a home region by a hash function. This mechanism satisfies the scalability requirement of MANET and reduces the information update overhead with the small cost of querying delay time. Nevertheless, most of the research [5]-[7] assumes the existence of a global hash function without any specific procedure. Moreover, these home-region protocols rely on the predefined rectangle or cycle system which is calculated from the information of GPS device [6], [7]. This could lead to other issues. Empty home regions may exist due to geographic condition. Nodes which are supposed to store information in these home-regions have to find alternative regions for that purpose. For the consistency of home region mechanism, every other node also needs to know about the empty home regions and the alternative regions of all destinations. These are major disadvantages of predefined geographic system.

Motivated by the challenge of lookup service, we propose a cluster-based hashing lookup service, CHLS, in this paper. Rather than relying on GPS device and the predefined geographic grid or circle system, we exploit the self-adaptive and self-organized multi-hop cluster structure [8], [9]. Then, we present the uniform distributed hash function which maps a node ID to a cluster ID. Based on this hashing function, the anchor system for lookup service is presented. Finally, we compare with other schemes under different system settings to show the scalability and effectiveness of our proposed scheme.

The rest of this paper is organized as follows. In the next section, we propose anchor system for cluster-based lookup service. The procedure of hashing function which hashes a node ID to cluster ID is also presented. Section 3 presents the performance evaluation of our scheme and others. Finally, in section 6, we conclude the paper.

2. Proposed Scheme

We utilize the anchor cluster system which is similar to home region system. It is called anchor cluster system because we build that on top of multi-hop cluster topology rather than predefined circle or grid system. Compared with grid system, clustering does not have the empty region problem with lookup service and allows self-organized, self-adaptive topology. Any clustering scheme [8] can be used for our proposed lookup service. In this paper, we use a multi-hop clustering which is similar to 3hBAC scheme [9]. Cluster ID is selected as the node ID of the cluster-head. We assume that the cluster topology information is propagated through inter-cluster routing to every node. The topology information includes IDs of all clusters and the logical cluster-link between adjacent clusters. Each node is not aware about the membership of other clusters except its own current cluster. For the location service, in this paper, we assume that each node has its own information and each query is generated for information of a specific node. Hence, information is associated with a node ID. We use a specific type of information, current cluster location, for performance evaluation. However, in general, lookup information can have its own scope of ID. In that case, we just need to substitute the node ID by the lookup information ID.

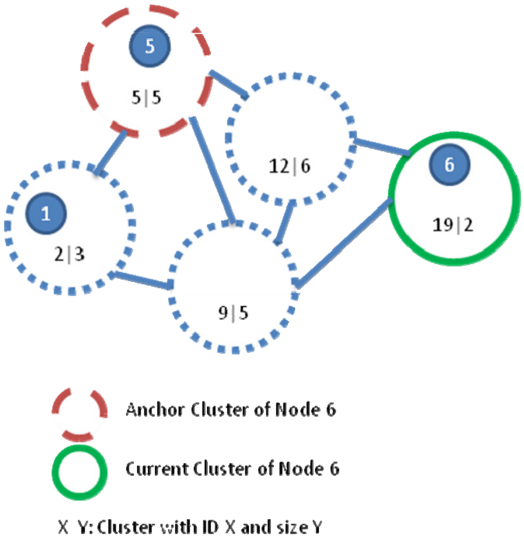

Figure 1: Uniform Dynamic Hashing and Lookup Service.

At the initial stage, the anchor system is constructed as follows. Using the topology information and its own node ID as input on the global hash function, each node obtains its anchor cluster ID. The global hash function must return the same result by the same input even though it may be performed by different node. After getting the hashed cluster, the node sends UPDATE message which contains a pair of values, its node ID and the attached information, to its anchor cluster. Let the cluster-head of a specific anchor cluster be anchor cluster-head and make it store all hashed information. In real environment, in order to balance the overhead, other nodes than cluster-head can share the work. With the help of inter-routing, the corresponding anchor cluster-head can receive the UPDATE message and store the pair values of node ID and attached information.

The same process occurs when a source needs to query desired information of a destination. The source performs a hash on destination ID with the cluster topology and obtains the anchor cluster of the destination. The uniqueness characteristic of hash function ensures the correct result of destination anchor cluster. Then, the source sends the QUERY message which contains the destination ID to the corresponding anchor cluster-head. Later, the desired information is returned to the source by the REPLY message.

The hash function takes the cluster topology and node ID x as input and returns the anchor cluster ID of the corresponding node. First, sort the cluster list CL in the order of increasing cluster ID. Let the number of clusters in the list be length(CL). Use the modulo operation and obtain the increasing list index, $ACIx = x \bmod length(CL)$. The anchor cluster ID is then $ACx = CL[ACIx]$. This function is simple and ensures the global uniqueness characteristic. As an illustration, there are five clusters with specific ID and size in Figure.1. After sorting the clusters by ID, list CL is [2;5;9;12;19] with length(CL) = 5. Regarding node 6, its anchor cluster index is then $ACI6 = 6 \bmod length(CL) = 1$ and $AC6 = CL[ACI6] = 5$. It means the cluster 5 is the anchor cluster of node 6. By the basic mechanism of lookup service, node 6 updates its information to cluster-head 5 which then replies the information to other sources.

3. Performance Evaluation

Parameters	Value
Number of nodes, n	400 $\sim$ 1000 nodes
Node Placement Strategy	Random
Network Area	3000x3000 m^2
Transmission Range	250 m
Mobility Model	Random Waypoint
Maximum Velocity, v	5 $\sim$ 20 m/s
Clustering k-hop value	2 $\sim$ 5
Total number of queries, n_q	5 $\sim$ 25 queries per second

Table 1: Simulation Parameters

In this part, we simulate, study the performance result of our proposed scheme, CHLS, and compared our result with reactive lookup service, RLS [2], and KCLS [4]. The simulation time for each test is 10 minutes and the mean result is obtained by 500 running times. The common parameters used in the simulation are listed in Table.1. The lookup service in this simulation is designed for location information of mobile nodes. In RLS, location information of one destination is the route to that node, starting from the querying source. In cluster-based protocols, the location information is the current cluster of the destination. We choose this specific information for cluster-based schemes to measure the information update overhead. When one mobile node leaves its current cluster and joins another cluster, it needs to update the location information to cluster-heads and consumes communication overhead. To compare cluster-based schemes correctly, we implement KCLS and CHLS on top of the same clustering structure.

In Table.2, we summarize the statistics of the clustering scheme which is used as the base system for both KCLS and CHLS. The leaving rate is count as the number of events in which mobile nodes leaves its current cluster and join another in one minute. This parameter directly affects the information update overhead for both two cluster-based location services.

CHLS is our proposed lookup service which uses uniform dynamic hashing. For the expanding ring search of RLS, the TTL field of flooding scheme is initialized to 2 for the first round of flooding; Then, it increases by 2 per round. We do not use caching mechanism for all schemes. First, we compare the communication overhead of these schemes in Table.3. This overhead consists of 3 components: initialization, maintenance and querying. RLS does not have overhead in initialization and maintenance. In the initialization phase of KCLS, every cluster-head forwards its membership to all other cluster-heads. In the case of CHLS, every mobile node computes its anchor cluster and sends an UPDATE message to the anchor server. When mobile nodes leave and join clusters, they need to update the location information to responsible cluster-heads. The initialization, and maintenance overheads are calculated as the number of messages which are forwarded for these operations in each component. The querying overhead is the number of messages which are used to query desired location information and get the reply. Maintenance and querying overheads are counted for each node per minute.

n	k	v	Number of clusters	Leaving Rate
400	3	10	34.17	91.42
600	3	10	28.00	93.67
800	3	10	24.75	99.00
1000	3	10	23.17	113.00
800	2	10	44.92	126.42
800	3	10	24.75	99.00
800	4	10	15.92	79.17
800	5	10	11.33	67.33
800	3	5	23.25	53.42
800	3	10	24.75	99.00
800	3	15	26.17	149.92
800	3	20	27.42	208.42

Table 2: Statistics of Clustering

In Table.3, we can see that even though RLS does not have initialization and maintenance overheads, the cost of its querying is too high. This is due to the underlying flooding mechanism. As the number of queries n_q increases, this cost becomes more significant. On the other hand, KCLS has small overhead in querying because each source node only needs to send the querying message to its current cluster head. But the downfall of KCLS lies in its cost of initialization and maintenance. Whenever a node changes its current location, the update notification from that node is sent to every cluster-head in the networks. This makes KCLS unsuitable to scenarios with high mobility. In case of CHLS, by the use of dynamic hashing, it can achieve small communication overhead.

4. Conclusions

In this paper, we investigate the combination of global hashing lookup service and cluster structure. Compared with other hashing grid-based lookup service which depends on GPS device, our scheme is built on a multi-hop cluster topology. The clustering allows us to avoid geographic issues such as empty home region, predefined circle or rectangular area. By using the dynamic hashing function, our scheme, CHLS, achieves much smaller communication overhead than RLS and KCLS. Finally, we provide extensive simulation results by comparing with reactive scheme and KCLS under different system settings.

n	k	v	n_q	Initialization			Maintenance			Querying		
				RLS	KCLS	CHLS	RLS	KCLS	CHLS	RLS	KCLS	CHLS
400	3	10	5	0	28.98	10.68	0	126.44	8.44	157.42	2.57	14.99
600	3	10	5	0	18.56	9.79	0	84.40	6.93	157.61	1.92	9.39
800	3	10	5	0	10.79	9.67	0	63.07	6.05	156.72	1.47	6.70
1000	3	10	5	0	8.07	9.21	0	50.72	5.91	158.79	1.21	5.27
800	2	10	5	0	34.74	9.31	0	133.90	6.38	156.72	1.06	6.97
800	3	10	5	0	10.79	9.67	0	63.07	6.05	156.72	1.47	6.70
800	4	10	5	0	4.77	9.85	0	31.20	5.98	156.72	1.85	6.52
800	5	10	5	0	2.35	9.96	0	16.08	5.92	156.72	2.25	6.20
800	3	5	5	0	10.79	9.67	0	32.86	5.57	157.00	1.50	6.68
800	3	10	5	0	10.79	9.67	0	63.07	6.05	156.72	1.47	6.70
800	3	15	5	0	10.79	9.67	0	101.31	6.98	157.73	1.45	6.76
800	3	20	5	0	10.79	9.67	0	147.69	7.60	156.76	1.41	6.88
800	3	10	5	0	10.79	9.67	0	63.07	6.05	156.72	1.47	6.70
800	3	10	10	0	10.79	9.67	0	63.07	6.05	317.59	2.94	13.56
800	3	10	15	0	10.79	9.67	0	63.07	6.05	474.13	4.43	20.36
800	3	10	20	0	10.79	9.67	0	63.07	6.05	633.19	5.90	27.11
800	3	10	25	0	10.79	9.67	0	63.07	6.05	792.98	7.39	33.91

Table 3: Communication Overhead

Acknowledgement

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2008-(C1090-0801-0046)). Prof.Choo is the corresponding author.

References

[1] T. Camp, J. Boleng, and L. Wilcox, “Location Information Services in Mobile Ad hoc Networks,” Proceedings of the IEEE International Conference on Communications, 2002, pp. 3318-3324.

[2] M. Kasemann et al, “A reactive location service for mobile ad hoc networks,” NEC Network Laboratories NL-E Heidelberg, Tech. Rep. TR-02-014, November 2002.

[3] N.D. Pham and H. Choo,, “Energy Efficient Expanding Ring Search for Route Discovery in MANETs,” Proceedings of the IEEE International Conference on Communications, 2008, pp. 3002 - 3006.

[4] S. Leng, L. Zhang, H. Fu and J. Yang, “A Novel Location-Service Protocol Based on k-Hop Clustering for Mobile Ad Hoc Networks,” IEEE Transactions on Vehicular Technology, Volume 56, Issue 2, March 2007, pp. 810 - 817.

[5] J. Erikson, M. Faloutsos and S.V. Krishnamurthy, ``DART: Dynamic Address Routing for Scalable Ad Hoc and Mesh Networks," IEEE/ACM Transactions on Networking, Volume 15, Issue 1, February 2007, pp. 119-132.

[6] S.C. Woo, and S. Singh, ``Scalable routing protocol for ad hoc networks," Wireless Networks, Volume 7, September 2001, pp. 513-529.

[7] H. Cheng, J. Cao, and H.H. Chen, “GRLS: Group-based Location Service in Mobile Ad hoc Networks,” Proceedings of the IEEE International Conference on Communications, 2007, pp. 4734-4740.

[8] J.Y. Yu, and P.H.J. Chong, “A survey of Clustering schemes for Mobile Ad hoc Networks,” IEEE Communication Surveys and Tutorials, Volume 7, Issue 1, 2005, pp. 32-48.

[9] J.Y. Yu, and P.H.J. Chong, “3hBAC(3-hop between adjacent cluster-heads): A novel Non-overlapping Clustering Algorithm for Mobile Ad hoc networks,” IEEE Pacific Rim Conference on Communication, Computers and Signal Processing, August 2003, pp. 318-321.