

An efficient Method for Tracking Nodes (EMTN) in Mobile Ad Hoc Networks

Anuradha Banerjee
Dept. of Computer Applications,
Kalyani Govt. Engg. College
West Bengal, India

Dr. Paramartha Dutta
Dept. of Computer and System
Sciences
Visva-Bharati University, Santiniketan
West Bengal, India

ABSTRACT

In this article, we propose a simple as well as efficient method of tracking nodes (EMTN) in mobile ad hoc networks. During the process of route discovery, source node of a communication session floods ROUTE_REQUEST packets to find a suitable route to the desired destination. EMTN aims at reducing the number of ROUTE_REQUESTS at each hop. This generates great performance enhancement especially in dense networks. Simulation results show that EMTN significantly increases the packet delivery ratio while decreasing the cost of messages and consumption of energy of network nodes at the cost of a bit extra bandwidth.

General Terms

Ad hoc networks, routing protocol.

Keywords

Cache, Directional Antenna, Flooding, HELLO, ROUTE_REQUEST.

1. INTRODUCTION

A mobile ad hoc network is a collection of mobile nodes that communicate with each other without the aid of an established infrastructure or centralized administration. Its applications range from civilian use to emergency rescue sites and battlefields. The communication in ad hoc networks is either single hop or multi-hop. If a node n_j resides within the radio-range or transmission range of another node n_i , then it can directly receive messages from n_i and the communication will be termed as single hop. Otherwise, a chain of intermediate nodes need to establish a multi-hop bridge between the source and destination nodes. Those intermediate nodes are often termed as routers [1-11].

Several proactive and reactive routing protocols have been proposed for ad hoc networks. Among proactive routing protocols destination-sequenced distance vector (DSDV)[1], wireless routing protocol (WRP) [2], global state routing (GSR) [3] and source-tree adaptive routing (STAR) [4] are well known. They suffer from extremely huge storage overhead because they store information both about active and non-active routes. Reactive routing protocols are designed to reduce this overhead. They instruct the nodes to discover routes in a demand-driven manner and maintain information about active routes only. Dynamic

source routing (DSR) [5], ad hoc on-demand distance vector routing (AODV) [6], light-weight mobile routing (LMR) [7] etc. are important reactive routing protocols. Among the state-of-the-art reactive routing protocols, flow-oriented routing protocol (FORP) [8] and associativity based routing (ABR) [9] are mention-worthy. During route discovery in reactive routing protocols, the source node floods ROUTE_REQUEST message to find a suitable route to destination. EMTN aims at reducing the number of route-requests at each hop. Reduction in number of control packets reduce packet collision in the network that results in increase in packet delivery ratio reducing the delay in tracking the destination.

2. DESCRIPTION AND ANALYSIS OF EMTN

2.1. The EMTN Method in Detail

EMTN instructs each node to transmit HELLO message at regular intervals. The components of HELLO message transmitted by n_i at time t include identification number of the node (n_i), its radio-range (R_i), present geographical position of n_i in terms of an ordered pair of latitude and longitude ($latt_i(t)$, $long_i(t)$), interval between consecutive HELLO messages (int_i), current timestamp t and timestamps of ROUTE_REQUEST messages that arrived at n_i between $(t - int_i)$ and t , along with identification numbers of the source nodes generating those ROUTE_REQUESTS. The value of int_i is determined by satisfying the constraint that $int_i < ((m_i - 1)\mu_i / 4)$ where m_i is the number of locations in message queue of n_i and μ_i is the time required to process a message. The nodes residing within the radio-range of n_i at time t are termed as downlink neighbors of n_i at that time and n_i is called their uplink neighbor. A node may have several uplink neighbors. After receiving the HELLO message transmitted by n_i they reply with an acknowledgment message (ACK) containing their identification numbers, latitude and longitude information, radio-range, current timestamp and a RREQ_PROCESS_VECTOR. The field RREQ_PROCESS_VECTOR is a string of 0's and 1's indicating whether the node has already received the ROUTE_REQUESTS arrived at n_i between $(t - int_i)$ and t , or not. If the string is 1001 for a downlink neighbor n_j of n_i , then it denotes that 4 ROUTE_REQUESTS arrived at n_i between $(t - int_i)$ and t among which the first and the fourth one (as per the sequence mentioned in the HELLO message transmitted by n_i at time t) has been received already by n_j from some other uplink neighbor. Hence,

n_i need to forward only the second and third ROUTE_REQUEST to n_j . Before arrival of ACK from all the downlink neighbors, the pending ROUTE_REQUEST messages are entirely stored by n_i in a cache C1. Each node also maintains a cache C2 of ROUTE_REQUESTS it has processed or is processing presently. C2 contains the timestamps of those processed ROUTE_REQUESTS and identification numbers of their sources. C is erased at the end of the day. After receiving a HELLO message containing information about ROUTE_REQUESTS, a node searches in its cache C2 to find out whether the request has already been processed or not. If it has been processed then the corresponding field of RREQ_PROCESS_VECTOR is set to 1. The field is set to 0 if the node has not received that ROUTE_REQUEST or it does not have any downlink neighbor. Depending upon the RREQ_PROCESS_VECTORS and geographical positions of downlink neighbors of n_i , n_i uses directional antenna to forward ROUTE_REQUESTS. This also helps to decrease redundancy during route discovery. If the downlink neighbors of n_i who did not receive any particular ROUTE_REQUEST, are placed far apart so that use of directional antenna will cover more than 70% of the radio-circle of n_i , n_i decides to unicast or broadcast depending upon the comparative energy consumption. Let u_i and b_i denote the cost (in terms of energy consumption) of unicasting and broadcasting of a message by node n_i to a downlink neighbor placed on the periphery of the radio-circle of n_i . If p_i be the number of downlink neighbors of n_i who did not receive one particular ROUTE_REQUEST message, then n_i will unicast the message provided ($p_i u_i < b_i$). The energy saved in the process compared to ordinary broadcasting is $(b_i - p_i u_i)$.

Please note that EMTN instructs nodes to process ROUTE_REQUESTS as soon as they arrive in the nodes. If a node finds that one particular ROUTE_REQUEST needs to be forwarded then it forwards that without delay i.e. the ROUTE_REQUESTS are never kept in message queues in EMTN. Immediately after receiving a HELLO message, the receiver sends the ACK provided some other message (may or may not be HELLO message) has not already arrived. If a message arrives before transmission of ACK, then first that message is received and then the underlying node tries to transmit the ACK.

2.2. Redundant Uplink Neighbors in EMTN

Let a node n_p has three uplink neighbors n_i , n_j and n_k . Assume that, the HELLO message transmitted by n_i at time t consists of the fields (n_i , R_i , $\text{latt}_i(t)$, $\text{long}_i(t)$, 8, t , n_s , t_s , n_a , t_a , n_b , t_b). Similarly the HELLO message transmitted by n_j at time $t+2$ consists of (n_j , R_j , $\text{latt}_j(t+2)$, $\text{long}_j(t+2)$, 5, $t+2$, n_s , t_s , n_m , t_m) and the one transmitted by n_k at time $t+1$ contains the information (n_k , R_k , $\text{latt}_k(t+1)$, $\text{long}_k(t+1)$, 10, $t+1$, n_s , t_s , n_r , t_r) where t_s , t_a , t_b , t_m and t_r all are less than t . The nodes n_s , n_a , n_b , n_m and n_r all have generated ROUTE_REQUESTS before time t . The values of int_i , int_j and int_k are 8, 5 and 10 respectively. n_p could not send ACK to n_i and n_k immediately after receiving their HELLO messages because the HELLO messages continued to come at n_p

and n_p had to receive them all. At time $t+2$, all the three uplink neighbors of n_p are ready to forward it the ROUTE_REQUEST generated by n_s at time t_s . Now n_p will have to specify in its RREQ_PROCESS_VECTORS from whom it wishes to receive the ROUTE_REQUEST with timestamp t_s generated by n_s . This is termed as “Redundant Uplink Neighbor” problem. In order to solve this problem, n_p assigns weights to the redundant uplink neighbors. The weights $w_{ip}(t+2)$, $w_{jp}(t+2)$ and $w_{kp}(t+2)$ assigned by n_p to n_i , n_j and n_k respectively, at time t . Mathematical expression for $w_{ip}(t+2)$ appears in (1).

$$w_{ip}(t+2) = f1_{ip}(t+2) f2_i f3_{ip}(t+2) \quad (1)$$

$$\text{where, } f1_{ip}(t+2) = (1 - \text{dist}_{ip}(t+2)/(R_i+1))$$

$$f2_i = ((R_i - R_{\min})/(R_{\max} - R_{\min}))$$

$$f3_{ip}(t+2) = (1-1/\lambda_{ip}(t+2))$$

$\text{dist}_{ip}(t+2)$ is the distance between n_i and n_p as per the latitude and longitude information available till time $(t+2)$. n_p is continuously residing within the radio-circle of n_i from time $(t + 2 - \lambda_{ip}(t+2))$ till time $(t+2)$. R_{\min} and R_{\max} denote the minimum and maximum possible radio-ranges of the network, respectively. The weight of an uplink neighbor of n_p increases if it is close to n_p (the proximity will increase the signal strength and reduce the time required by a signal to reach from that uplink neighbor of n_p to n_p) and there are chances that the link between them will survive in near future. The chance of link survival depends upon two things – the time duration for which the link has survived and the radio-range of the uplink neighbor within the range from R_{\min} to R_{\max} . If a link has already continuously survived for long, then chances are high that the link will remain alive in near future. Moreover, a node with high radio-range has more capacity to preserve a downlink neighbor within its radio-circle compared to a node with comparatively low radio-range. The issue of survival of link is important in EMTN because if the responsibility of forwarding a ROUTE_REQUEST is assigned to n_i by n_p where the link from n_i to n_p is fragile, then it may happen that before sending the ROUTE_REQUEST n_i loses its link with n_p . If a node has redundant downlink neighbors, then any one of those with highest weight, is picked up as the optimal choice.

2.3. Bandwidth Overhead in EMTN

Let α_i be the time required by n_i to receive a ROUTE_REQUEST. Then the maximum number of ROUTE_REQUESTS that can be received by n_i within time interval int_i , is given by (int_i/α_i) . Assuming L to be the largest identification number of a node in the network and T to be the highest timestamp of a day, the number of bits required by the HELLO message of n_i for representing ROUTE_REQUEST information is given by $(\text{int}_i/\alpha_i)(\log_2 L + \log_2 T)$. The number of bits in the RREQ_PROCESS_VECTOR of all downlink neighbors of n_i is (int_i/α_i) .

2.4. Delay in EMTN

In the dense and congested networks of today, a ROUTE_REQUEST message is not expected to be processed by

a node as soon as it arrives at that node. If there are pending messages to be forwarded, the ROUTE_REQUEST message has to wait in message queue of the underlying node. Please note that the messages to be forwarded are forwarded on a first-come-first-served basis in the unicast protocols proposed for ad hoc networks in references 1-9. Assuming m_i to be the size of message queue of a node n_i , in worst case a message has to wait for processing of (m_i-1) number of messages and in best case a message is processed immediately after arrival. For simplicity in analysis let us assume both the above cases to be equally likely. Then, on an average, the waiting time of a message (including ROUTE_REQUEST) in existing protocols is $(m_i-1)/2$. On the other hand, in EMTN embedded protocols, a ROUTE_REQUEST packet may have to wait in cache C1 of n_i for less than $(2 \times \text{int}_i)$ time interval. Since the value of int_i is determined by satisfying the constraint that $\text{int}_i < ((m_i-1)\mu_i/4)$, $(2 \times \text{int}_i)$ is definitely less than $((m_i-1)\mu_i/2)$.

2.5. Complexity of EMTN

EMTN instructs a node to look into its cache C2 before replying the HELLO message of any uplink neighbor and if redundant uplink neighbors are there, weights are to be assigned to them and one of them with maximum weight will be selected. The maximum possible number of entries in C2 of any node is $N.T$ where N is the total number of nodes in the network and T is all possible timestamps of a day. Applying the method of binary search will reduce the cost of searching an entry in C2. The associated cost is $O(\log_2(N.T))$ i.e. $O(\log_2(N))$ since T is a constant. At any point of time, a node may have at most N redundant uplink neighbors and the complexity of assigning weight to each of them is $O(1)$. Hence, the complexity of resolving the redundant uplink neighbor problem is $O(N)$.

The space complexity of EMTN is due to the caches C1 and C2. Hence, the complexity is $O(N)$.

3. SIMULATION RESULTS

Simulation of the mobile network has been carried out using ns-2 [12] simulator on 800 MHz Pentium IV processor, 40 GB hard disk capacity and Red Hat Linux version 6.2 operating system. Graphs appear in figures 1, 2, 3 and 4 showing emphatic improvements in favor of EMTN embedded protocols. Number of nodes has been taken as 50, 100, 200, 500, 1000 and 2000 in six different independent simulation studies. Speed of a node is chosen randomly between 0 to 45m/sec. Transmission range varied between 5m and 60m. Used network area is $3000m \times 3000m$. Used traffic type is constant bit rate. Mobility models used in various runs are random waypoint, random walk and Gaussian. Performance of some state-of-the-art reactive protocols, for example, DSR, AODV and ABR are compared with their EMTN embedded versions (only the route-discovery part of the protocols follow the logic of EMTN; selection of the optimal route, communication and link breakage recovery – all these parts of the protocols remain unchanged).

In order to maintain uniformity of the implementation platform, we have used ns-2 simulator for all the above-mentioned communication protocols. The simulation matrices are packet delivery ratio (total no. of packets delivered/total no. of packets transmitted $\times 100$), total message overhead, delay in tracking the destination ((for all generated ROUTE_REQUEST message, summation of (smallest timestamp of tracking a destination – timestamp of generation of the ROUTE_REQUEST message))/total no. of ROUTE_REQUESTS generated) and summation of energy consumption of nodes. Simulation time was 500 sec. for each run.

packet delivery ratio vs number of nodes

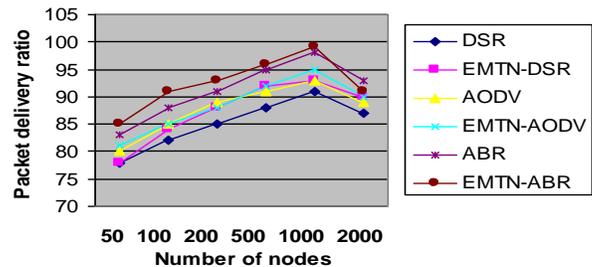


Fig1: Packet delivery ratio vs number of nodes

cost of messages vs number of nodes

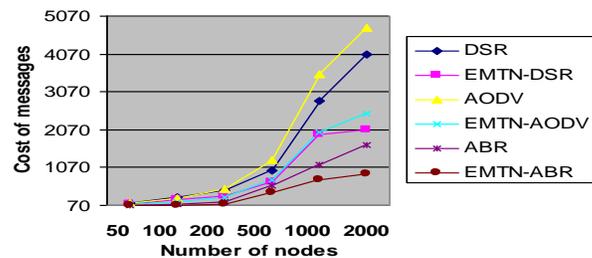


Fig 2: Message cost vs number of nodes

Energy consumption vs number of nodes

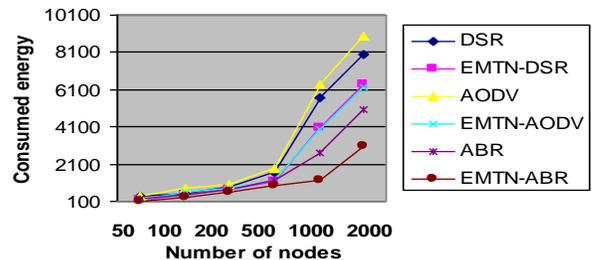


Fig 3: Consumed energy vs number of nodes

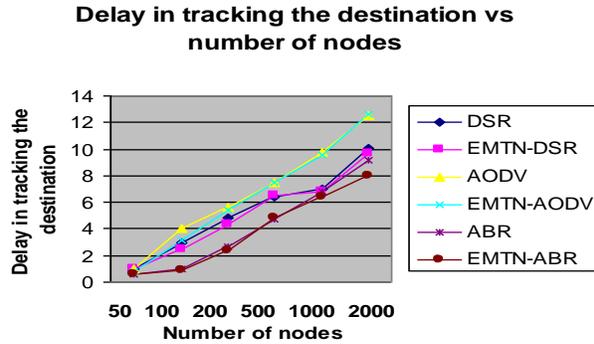


Fig 4: Delay in tracking the destination vs number of nodes

Since redundant ROUTE_REQUESTS are much lesser in EMTN embedded protocols, hence the cost of messages as well as consumption of energy in nodes, are much lesser in those compared to the ordinary version of the protocols. Reduced message cost generates less signal collision and as a result, packet delivery ratio increase with decrease in the delay in tracking the destination. Energy saving is around 50% in EMTN embedded protocols, whereas the improvement in respect of packet delivery ratio is 15.14% and in case of delay in tracking the destination, it is 7.3%.

4. CONCLUSION

EMTN is an efficient method that greatly reduces redundancy during the tracking of destination nodes. It is completely independent of the underlying reactive routing protocol. The logic of EMTN is simple to implement and it produces significant improvement at the cost of some extra bandwidth which is affordable.

5. REFERENCES

[1] C.E. Parkins and P. Bhagwat, Highly dynamic destination-sequenced distance-vector (DSDV) mobile computers, ACM Computer Communications Review, Vol. 24 No. 4, ACM SIGCOMM '94

[2] Murthy, S. and J.J. Garcia-Luna-Aceves, An Efficient Routing Protocol for Wireless Networks, ACM Mobile Networks and App. J., Special Issue on Routing in Mobile Communication Networks, Oct. 1996, pp. 183-197

[3] M. Gerla and T. W. Chen, "Global State Routing : A new routing scheme for Ad-Hoc wireless networks", Proc. IEEE ICC '98

[4] <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.8231>

[5] D. Johnson, D. A. Maltz, Dynamic source routing in ad hoc wireless networks, in Mobile Computing (T. Imielinski and H. Korth, eds.), Kluwer Acad. Publ., 1996)

[6] C.E. Perkins and E.M. Royer. Ad hoc on demand Distance Vector routing, mobile computing systems and applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on, 1999, pp.90 - 100

[7] M. S. Corson and A. Ephremides, A Distributed Algorithm for Mobile Wireless Networks, Wireless Networks, Vol. 1, pp. 61 – 81, 1995

[8] N. Meghanathan, "Energy Consumption Analysis of the Stable Path and Minimum Hop Path Routing Strategies for Mobile Ad hoc Networks," International Journal of Computer Science and Network Security, Vol. 7, No. 10, pp. 30 – 39, October 2007

[9] N. Meghanathan, A Beaconless Node Velocity-based Stable Path Routing Protocol for Mobile Ad hoc Networks, Proceedings of the IEEE Sarnoff Symposium Conference, Princeton, NJ, March 30- April 1, 2009

[10] W. Su, S-J. Lee and M. Gerla, Mobility Prediction and Routing in Ad hoc Wireless Networks, International Journal of Network Management, Vol. 11, No. 1, pp. 3-30, 2001

[11] Pallavi Khatri, Monika Rajput, Alankar Shastri and Keshav Solanki, Performance study of ad hoc reactive routing protocols, Journal of Computer Science, vol. 6, no. 10, pp. 1130 – 1135, 2010.

AUTHOR PROFILE

Anuradha Banerjee is currently working as a lecturer in Kalyani Govt. Engg. College. She completed her B.E. in Computer Science And Technology from Bengal Engineering College (Deemed University), Sibpur, West Bengal, India in the year 2001. She has 12 publications till date, including National and International conferences and journals. Her research area includes Ad Hoc Networks, Neural Networks and Artificial Intelligence.

Dr. Paramartha Dutta is currently working as a Professor in Visva-Bharati University, Santiniketan. He has completed his M. Tech. from Indian Statistical Institute, Calcutta in 1993 and Ph. D. in Computer Science and Engineering from Bengal Engineering And Science University, Sibpur, West Bengal, India in the year 2005. He has numerous publications in National and International conferences and journals. His research interests include Ad Hoc Networks and Image Processing.