PSR Protocol with NN-Query for Mobile Ad Hoc Networks

Nijina A

M.Tech student, Department of Electronics and Communication Engineering, Mount Zion College of Engineering, Kadammanitta, Pathanamthitta, Kerala, India

Abstract: ***A Mobile Ad-hoc Network(MANET) is a wireless communication network where the nodes are not within the direct transmission range. Hence intermediate nodes are required to forward data. Opportunistic data forwarding can't be used in MANETs since the lack of an effective routing protocol. In this paper, a novel routing protocol is designed to use opportunistic data forwarding in MANETs with beacon-less nearest neighbor query processing method for reducing traffic and maintaining high accuracy of the query result in mobile ad hoc networks. In these methods, the query-issuing node first forwards a query using geo-routing to the nearest node from the point specified by the query (query point). Then, the nearest node from the query point forwards the query to other nodes close to the query point, and each node receiving the query replies with the information on itself. The protocol which has superior performance compared to other routing protocol such as link state (LS)-based routing, DV-based protocols and reactive source routing. Such a routing protocol is known as PSR(Proactive Source Routing) protocol.PSR protocol has less overhead than the baseline protocols. The simulation is done by ns2.***

Keywords: Mobile Ad-hoc Network, Opportunistic data forwarding, Streamlined Differential Update, Neighborhood Trimming.

1. Introduction

A MANET (Fig 1) is a wireless communication network comprises of multiple nodes that are able to move in any direction and change network topology at any time. Here the nodes are not in direct transmission range. In order forward data, intermediate nodes are required. It has routing protocol similar to that of wired network. The applications of MANET include battlefield communications, emergency operations, search and rescue, disaster relief operations, and also for civilian applications such as community networks. These traditional routing protocols select the best sequence of nodes between the source and destination to forward each data packet through that sequence. There are two important functions at the network layer, i.e., data forwarding and routing. Data forwarding means how data packets are selected from one link and put on another. Routing means, a path that the data packet should choose from the source to the destination. Opportunistic data forwarding means, a path in which data packets are treated in a multihop wireless network. Here an intermediate node is required which looks up a forwarding table for next hop. Opportunistic data forwarding provides multiple downstream nodes to perform on the broadcast data packet. Here, propose a *proactive source routing* (PSR) *protocol* to allow opportunistic data forwarding in MANETs.

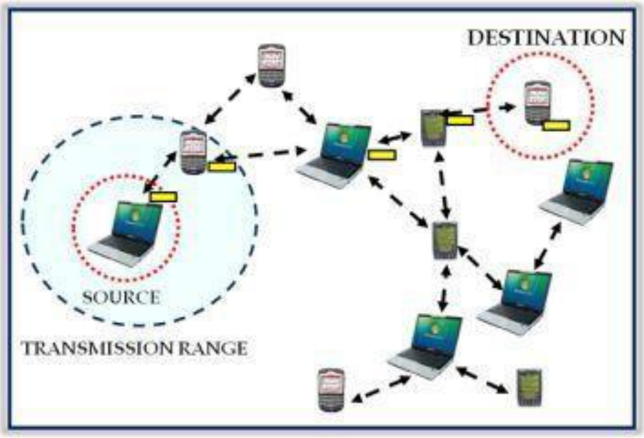

Figure 1: Mobile Ad-hoc Network

MANETs has some characteristics, such as limited network bandwidth, and its dynamic topology change occurs due to the motion of mobile nodes. The nodes continuously exchange beacon messages to find out changing locations of neighboring nodes. It will increase traffic, and this leads to frequent packet losses. In this paper, in order to reduce traffic a beaconless query processing method is used. Here every node compute a breadth-first search spanning tree of the rooted network, for updating network topology it is periodically exchanged between neighboring nodes. PSR provides a node to have full-path information to all other nodes in the network.

2. Motivation and Related Works

The earlier works of Opportunistic data forwarding involves selective diversity forwarding [2]. Here the transmitter selects the best forwarder from several receivers. The best forwarder is one which successfully receives its data. The transmitting node requests the selected node to forward the data. It can't be implemented in practical networks since its overhead is high. In [3, 4, 5], the authors proposed *k*NN query processing methods based on the query propagation method proposed in [6] in location-aware sensor networks. In [4], the authors proposed a free *k*NN query processing method called

DIKNN. In this method, first a source sends a query message to the nearest node from the query point. The nearest node to the. query point estimates search range (*k*NN boundary) based on the information on the density of nodes. The *k*NN boundary partitioned into sectors. In each sector, a sensor node collects the information on nodes within its communication range. The query propagates to the next node in the sector. Finally to acquire *k*NNs, the results are individually sent back from each sector to the query-issuing node. So many lightweight table driven routing protocols proposed for the Internet to address its scalability issue. The path-finding algorithm (PFA)[6] is mainly on DVs which improves by incorporating the predecessor of a destination in a routing update, then the entire path can be reconstructed by connecting the predecessors and destinations. Hence the source nodes have a tree topology rooted at it. The link vector (LV) algorithm [7] reduces the overhead of LS algorithms in data forwarding in routing updates to a great deal by only including links to be used. M. Al-Rabayah and R. Malaney proposes [8], A New Scalable Hybrid Routing Protocol for VANETs.. Here a routing protocol which combines AODV protocol with greedy forwarding geographic routing protocol that is mainly designed to point out the link failure within VANET. The technique involves the added features reactive with location_ based geographic routing for the use of all the location information available in the network. In [9], topology Control in Wireless Ad Hoc and Sensor Networks proposed. Topology Control (TC) is one of the most important techniques to reduce energy consumption and radio interference used in wireless ad hoc and sensor networks. The goal of this technique is to control the topology of the graph representing the communication links between network nodes for maintaining some global graph property.

3. Existing System

One of the initial works on opportunistic data forwarding is a light weight source routing protocol. In this protocol the nearest neighbors are find out by frequently sending the beacon messages. But it will increase unnecessary query messages and replies. This will lead to increase in traffic and due to this packet loss may occur.

4. Proposed System

A lightweight proactive source routing (PSR) protocol with nearest neighbors query to provide opportunistic data forwarding with less traffic in MANETs. Here inorder to reduce traffic, the nearest neighbors are find out by using a beacon less query processing method called spiral method. Let model the network as undirected graph G = (V, E), where V is the set of nodes (or vertices) in the network, and E is the set of wireless links (or edges). If they are close to each other and can directly communicate with given reliability, two nodes u and v are connected by edge e = (u, v) ∈ E. Given node v, we use N(v) to denote its open neighborhood, i.e., {u ∈ V |(u, v) ∈ E}. Similarly, we use N[v] to denote its closed neighborhood, i.e., N(v) ∪ {v}

Before going to the method, first describe the assumed environment. The system environment is assumed to be a MANET.The communication range is a circle of a fixed size. In MANET, mobile nodes collect the information using *k*NN queries on other mobile nodes. The query-issuing node sends a query message associated with the query point. Among all nodes in the entire network, collect the information on the *k* nearest nodes from the query point. Every node has the information of total number of mobile nodes (*n*), the size of the area in which mobile nodes are present, and also its current location (using a positioning system such as GPS).

A. NN-Query

In this paper in order to find out nearest neighbors query processing method is used, SPI(Spiral) method. In the SPI (Fig 2) method, the total area is dynamically divided into a set of hexagonal cells. The size of hexagonal cell is based on the communication range of the mobile nodes. The query point is located at the center of a hexagonal cell. The global coordinator starts sending the *k*NN query, in a spiral manner, to the nodes located nearest the central point of each hexagons (*local coordinator*). Each local coordinator acquire the information on all nodes in its hexagonal cell. It then forwards the query with this collected information, to the next hexagon in the spiral. Finally the node that surely collects the *k*NN result transmits the result to the query-issuing node.

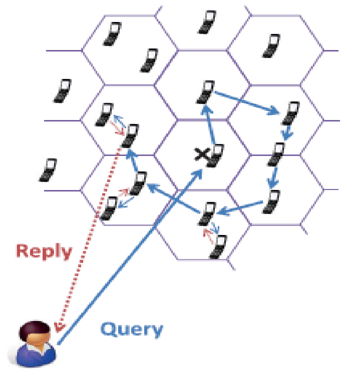

Figure 2: SPI Method

In this method, first the query issuing node broadcasts a neighbor search message. When a particular node receives this message, which is closer to the query point than the source node, and within the *searching range*, then it stores the ID of the source node as the parent node and sets the waiting time, *Reply_Delay* (*RD*), for transmitting reply, according to the following equation:

$$RD = \text{Max delay}\left(\frac{\propto - d}{\propto}\right) \quad (1)$$

where *Max_delay* is a positive constant specifying the maximum waiting time before sending a reply, α is the radius of the searching range, and *d* is the distance between the source node's location and the foot of the receiving node's perpendicular to the line from the source node to the query point.

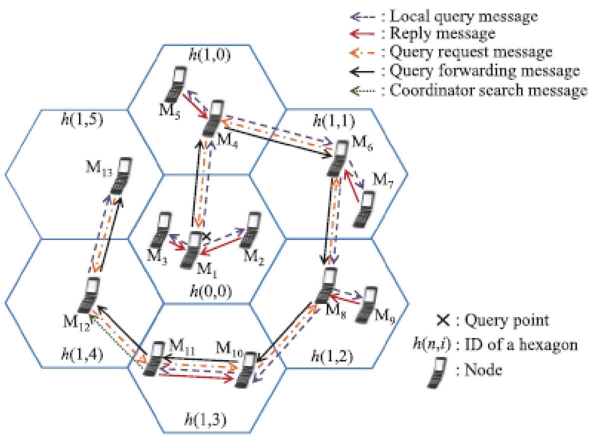

Figure 3: Query processing in SPI method

Fig 3 shows an example of executing the SPI method, where *M*1 is the global coordinator. Here *M*1 broadcasts a local query message to its neighboring nodes. *M*2 and *M*3 receives local query message from *M*1, they reply with their own information to *M*1 as a local reply since they are within the same hexagon as *M*1. *M*1 stores the information on itself. When *M*4 receiving the query message from *M*1, sends a query request message to *M*1 when *WT* has passed. When *M*1 receives this message from *M*4, *M*1 transmits a query message with the information on *M*2, *M*3, and itself. Such procedure continuous in all hexagons on the 1st lap, and finally M_{13} acquires the information on *k*NNs and initiates a reply to the query-issuing node.

B.Route Update

The update operation of PSR is iterative and distributed among all nodes in the network. At the beginning, node v is only know existence of itself; so, there is only a single node in its BFST, which is root node v. It is able to construct a BFST within N[v] by exchanging the BFSTs with the neighbors, i.e., the star graph centered at v, which is denoted Sv. In each iteration, nodes exchange their spanning trees with their neighbors. It has received a set of routing messages from its neighbors packaging the BFSTs, from the perspective of node v, toward the end of each operation interval. Note that, in fact, more nodes may be situated within the transmission range of v, but their periodic updates were not received by v due to bad channel conditions. From each neighbor to update its own BFST, node v incorporates the most recent information. It then broadcasts this tree at the end of the period to its neighbors. Formally, v has received the BFSTs from some of its neighbors. Including those from whom v has received updates in recent previous iterations, node v has a BFST, which is denoted Tu, for each neighbor u ∈ N(v). Node v constructs a union graph, i.e.,

$$G_v = S_v \cup \bigcup_{u \in N(v)} (Tu - v) \qquad (1)$$

Use T − x to denote the operation of removing the subtree of T rooted at node x. Consider T − x = T if x is not in T, and T − x = ∅ if x is the root of T. Then, node v find a BFST of Gv, which is denoted Tv, and places Tv in a routing packet to broadcast to its neighbors. The given update of the BFST happens multiple times within a single update interval so that a node can incorporate new route information to its knowledge base more quickly. To the extreme, Tv is modified every time a new tree is received from a neighbor. There is a tradeoff between the routing agent's adaptivity to network changes and computational cost. Here, choose routing adaptivity as a higher priority assuming that the nodes are becoming increasingly powerful in packet processing. Nevertheless, this does not increase the communication overhead at all because one routing message is always sent per update interval. Assume that the network diameter, i.e., the maximum distance, is D hops. Each node in the network has constructed a BFST of the entire network rooted at itself after D iterations of operation, since nodes are timer driven and, thus, synchronized. This information can be used for any source routing protocol.

C. Neighborhood Trimming

For a node to identify which other nodes are its neighbors, the periodically broadcast routing messages in PSR also double as "hello" messages. When a neighbor is deemed lost, its contribution to the network should be removed; this process is called neighbor trimming.

Consider node v. The neighbor trimming procedure is at v about neighbor u is either by the following cases:

1) No routing update or data packet has been received for a particular period of time.
2) A data transmission to node u has failed, reported by the link layer.
   Node v responds by:
1) first, updating N(v) with N(v) − {u};
2) then, constructing the union graph with the information of u removed, i.e.,

$$G_v = S_v \cup \bigcup_{w \in N(v)} (Tw - v) \ (2)$$

3) finally, computing BFST Tv.

Consider Tv, which is thus calculated, is not broadcast immediately to avoid excessive messaging. It is able to avoid sending data packets via lost neighbors with this updated BFST at v. So multiple neighbor trimming procedures may be triggered within one period.

D Streamlined Differential Update

In PSR, interleave the "full dump" routing messages with "differential updates." The main idea is to transmit the full update messages less frequently than shorter messages containing the difference between the current and previous knowledge of a node's routing module. Further streamline the routing update in two new avenues. First, use a tree representation in full-dump and differential update messages to halve the size of these messages. Second, every node attempts to maintain an updated BFST as the network changes so that the differential update messages are shorter.

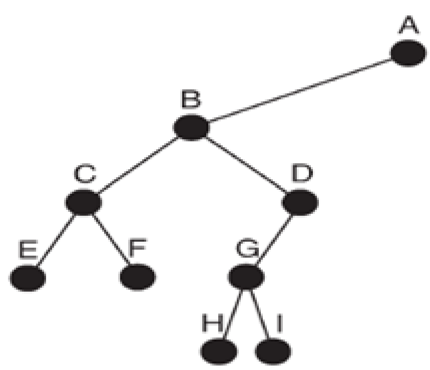

Figure 4: Binary tree

1) Binary tree representation. For the full-dump messages, our aim is to broadcast the BFST information stored at a node to its neighbors in a short packet. First convert the general tree into a binary tree using left-child sibling representation of the same size(s). Then serialize the binary tree using a bit sequence of 34 × s bits. Specifically, scan the binary tree layer by layer. In addition, append two more bits to indicate if it has the left and/or right child. For example, the binary tree in Fig.4 is represented as A10B11C11D10E00F00G11H00I00. The size of the update message is a bit over half compared with the traditional approach. The difference between two BFSTs can be represented by the set of nodes that have changed parents. We observe that these edges are often clustered in groups. That is, many of them form a tree subgraph of the network. A differential update message usually contains a few small trees, and its size is noticeably shorter.

2) Stable BFST. The size of a differential update is determined by how many edges it contains. Since there can be a large number of BFSTs rooted at a given node of the same graph, we need to change the BFST maintained by a node as little as possible when changes are detected. So modify the computation described earlier here, such that a small portion of the tree needs to change either when a neighbor is lost or when it reports a new tree.

Consider node v and its BFST Tv. When it receives an update from neighbor u, denoted Tu, it first removes the subtree of Tv rooted at u. Note that to reach every other node, the BFST of (Tv − u) ∪ Tu may not contain all necessary edges for v. Therefore construct union graph

$$(T_v - u) \cup \bigcup_{w\in N(v)} (Tw - v) \quad (3)$$

before calculating its BFST. To minimize the changes to the tree, add one edge of Tw − v to Tv − u at a time. When node v thinks that a neighbor u is lost, it deletes edge (u, v) but still utilizes the network structure information contributed by u earlier. That is, if it has moved away from v, node u may still be within the range of one of v's neighbors.

Tv should be updated to a BFST of

$$(Tv-u)\cup(Tu-v)\bigcup_{w\in N(v)}(Tw - v) \quad (4)$$

It provides every node with a neighbour table for the entire network. Here nodes periodically broadcast the table information to their best knowledge in each iteration. Based on the information collected from neighbours, a node can refresh its knowledge about the network topology by adding such recent information. This will be distributed to its neighbours in the next round of operation. When a neighbour is deemed lost, a procedure is triggered to remove its relevant information from the topology maintained by the detecting node. Differential update mechanism is also useful to reduce more routing overhead.

5. Simulation Results

The performance of PSR with that of DSR is compared. It has less overhead bytes. Also the traffic is reduced compared to beacon based methods. The simulation result shows that it has high throughput. The packet delivery ratio(PDR) is high for the proposed system since beacon less query processing method is used. The simulation results are shown in following figures. All are configured and tested in ns-2.

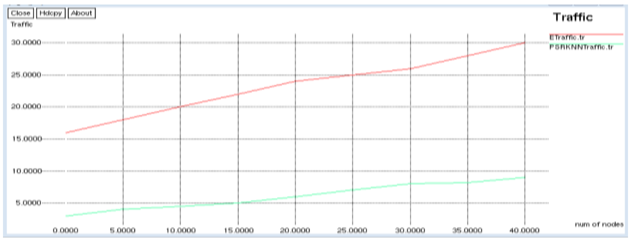

Figure 1: Traffic

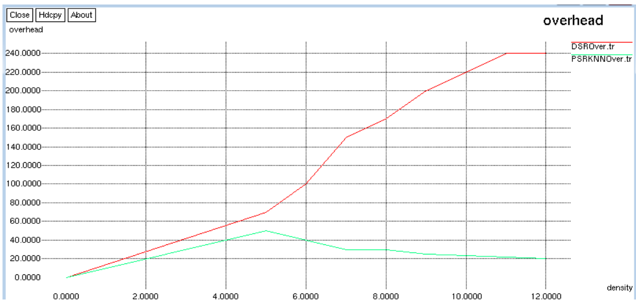

Figure 2: Routing overhead with density

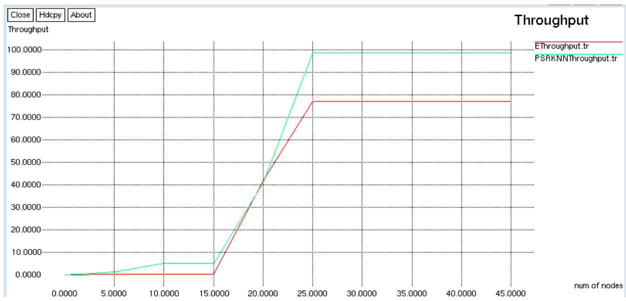

Figure 3: Throughput

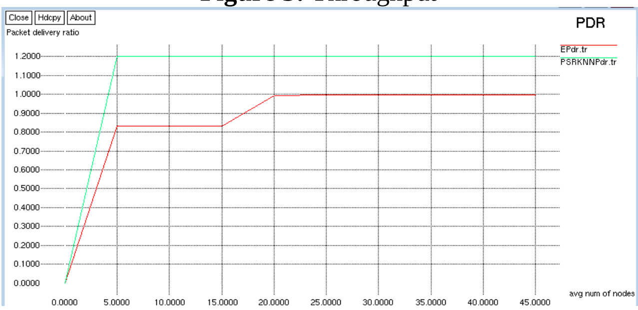

Figure 4: PDR

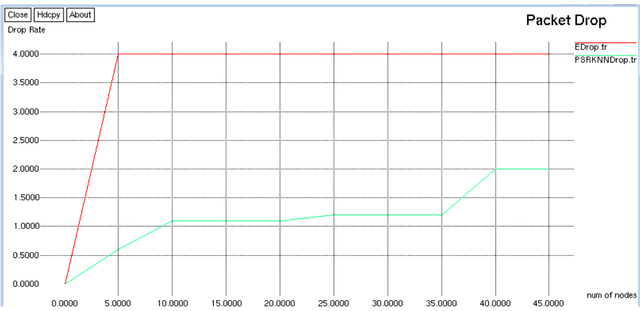

Figure 5: Packet dro

6. Conclusion

This paper has been motivated to support opportunistic data forwarding and to reduce unnecessary query messages and replies in *MANETs*. Here each node has the full-path information to reach all other nodes. First, ***to find out nearest neighbors a beacon less query processing method is used In these method, the query-issuing node first forwards a query using geo-routing to the nearest node from the point specified by the query (query point). Then, the nearest node from the query point forwards the query to other nodes close to the query point, and each node receiving the query replies with the information on itself. Second,*** to exchange routing information, it sends the periodic route update. Third it packages a converted binary tree to reduce the size of the payload by about a half. Finally, interleave full-dump messages with differential updates so that, in relatively stable networks, the differential updates are much shorter than the full-dump messages. As a result, the routing overhead of PSR is only a fraction or less compared with DSR, Also the traffic is reduced compared with beacon based methods by this experiments.

7. Acknowledgement

I would like to express my gratitude to the following people for their support and guidance for the success of this paper. I express my deep sense of gratitude to Mrs. Ann Susan Vargheese, Assistant Professor, ECE, MZC Kadammanitta, PTA, Kerala India. I deeply indebted to Dr.Rajaram N, Principal, MZC, PTA, Kerala.

References

[1] Zehua, Wang, Student Member, IEEE Chen, Member, IEEE, and Cheng Li, Senior Member, IEEE"PSR: A Lightweight Proactive Source Routing Protocol For Mobile Ad Hoc Networks" IEEE Trans. On Vehic Tech, Vol. 63, No. 2, Feb 2014

[2] P. Larsson, "Selection diversity forwarding in a multihop packet radio network with fading channel and capture, " *ACM Mobile Comput. Commun. Rev.*, vol. 5, no. 4, pp. 47–54, Oct. 2013.

[3] T.-Y. Fu, W.-C. Peng, and W.-C. Lee, "Parallelizing itinerary-based KNN query processing in wireless sensor networks, " *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 5, pp. 711–729, May 2010.

[4] S.-H. Wu, K.-T. Chuang, C.-M. Chen, and M.-S. Chen, "Toward the optimal itinerary-based KNN query processing in mobile sensor networks, " *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 12, pp. 1655–1668, Dec. 2008.

[5] Y. Xu, T.-Y. Fu, W.-C. Lee, and J. Winter, "Processing k nearest neighbor queries in location-aware sensor networks, " *Signal Process.*, vol. 87, no. 12, pp. 2861–2881, 2007.

[6] Y. Xu, W.-C. Lee, J. Xu, and G. Mitchell, "Processing window queries in wireless sensor networks, " in *Proc. ICDE*, 2006, p. 70.

[7] S. Murthy, "Routing in packet-switched networks using path-finding algorithms, " Ph.D. dissertation, Comput. Eng., Univ. California, Santa Cruz, CA, USA, 2012.

[8] M. Al-Rabayah and R. Malaney, "A new scalable hybrid routing protocol for VANETs, " *IEEE Trans. Veh. Technol.*, vol. 61, no. 6, pp. 2625–2635, Jul. 2011.

[9] R. Rajaraman, "Topology control and routing in ad hoc networks: A survey, " *ACM SIGACT News*, vol. 33, no. 2, pp. 60–73, Jun. 2009.

[10] Y. P. Chen, J. Zhang, and I. Marsic, "Link-layer-and-above diversity in multi-hop wireless networks, " *IEEE Commun. Mag.*, vol. 47, no. 2, pp. 118–124, Feb. 2002.

[11] J. Behrens and J. J. Garcia-Luna-Aceves, "Distributed, scalable routing based on link-state vectors, " in *Proc. ACM Conf. SIGCOMM*, 1994, pp. 136–147.

[12] C. E. Perkins and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for mobile computers, " *Comput. Commun.Rev.*, vol. 24, pp. 234–244, Oct. 1994

Author Profile

Nijina A received the B.Tech degrees in Electronics and Communication Engineering from Kerala University at SHM College of Engineering in 2013. And now she is pursuing her M.Tech degree in Communication Engineering under the M.G University in Mount Zion College of Engineering.