Proactive Source Routing Protocol for Mobile Ad Hoc Networks Using BFST Algorithm

[1]V.Venkataramana, [2]Ramyashree.M

[1,2] Dept. of DECS ,VTU Regional Center, Mysuru
Email: Venkatv979@gmail.com, ramyaravikumar83@gmail.com

Abstract: Opportunistic data forwarding has drawn much attention in the research community of multihop wireless net- working, with most research conducted for stationary wireless networks. One of the reasons why opportunistic data forwarding has not been widely utilized in mobile ad hoc networks (MANETs) is the lack of an efficient lightweight proactive routing scheme with strong source routing capability. A mobile ad hoc network (MANET) is a wireless communication network, where nodes that are not within the direct transmission range of each other require other nodes to forward data It can operate without existing infrastructure and support mobile users, and it falls under the general scope of multihop wireless networking. This networking paradigm originated from the needs in battlefield communications, emergency operations, search and rescue, and disaster relief operations. It has more recently been used for civilian applications such as community networks. A great deal of research results have been published since its early days in the 1980s . The most salient research challenges in this area include end-to-end data transfer, link access control, security, and providing support for real-time multimedia streaming . The network layer has received a great deal of attention in the research on MANETs. In fact, the two most important operations at the network layer, i.e., data forwarding and routing, are distinct concepts. Data forwarding regulates how packets are taken from one link and put on another. Routing determines what path a data packet should follow from the source node to the destination.

I. INTRODUCTION

A mobile ad hoc network (MANET) is a continuously self-configuring, infrastructure-less network of mobile devices connected without wires. Ad hoc is Latin and means "for this purpose". Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic unrelated to its own use, and therefore be a router. The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route traffic. Such networks may operate by themselves or may be connected to the larger Internet. They may contain one or multiple and different transceivers between nodes. This results in a highly dynamic, autonomous topology.

MANETs are a kind of Wireless ad hoc network that usually has a routable networking environment on top of a Link Layer ad hoc network. MANETs consist of a peer-to-peer, self-forming, self-healing network in contrast to a mesh network has a central controller (to determine, optimize, and distribute the routing table). MANETs circa 2000-2015 typically communicate at radio frequencies (30 MHz- 5 GHz) The growth of laptops and 802.11/Wi-Fi wireless networking have made MANETs a popular research topic since the mid-1990s. Many academic papers evaluate protocols and their abilities, assuming varying degrees of mobility within a bounded space, usually with all nodes within a few hops of each other. Different protocols are then evaluated based on measures such as the packet drop rate, the overhead introduced by the routing protocol, end-to-end packet delays, network throughput, ability to scale, etc.

Opportunistic networks are one of the most interesting evolutions of MANETs. In opportunistic networks, mobile nodes are enabled to communicate with each other even if a route connecting them never exists. Furthermore, nodes are not supposed to possess or acquire any knowledge about the network topology, which (instead) is necessary in traditional MANET routing protocols. Routes are built dynamically, while messages are en route between the sender and the destination(s), and any possible node can opportunistically be used as next hop, provided it is likely to bring the message closer to the final destination. These requirements make opportunistic networks a challenging and promising research field.

II. BREADTH FIRST SEARCH

Breadth first search (BFS)is an algorithm for traversing or searching tree or graph data structures. It starts at the and explores the neighbor nodes first, before moving to the next level neighbors. Compare BFS with the equivalent, but more memory-efficient Iterative deepening depth-first search and contrast with depth-first search.

Breadth-first search starts at a given vertex s, which is at level 0. In the first stage, we visit all the vertices that are at the distance of one edge away. When we visit there, we paint as "visited," the vertices adjace1nt to the start vertex s - these vertices are placed into level 1. In the second stage, we visit all the new vertices we can reach at the distance of two edges away from the source vertex s. These new vertices, which are adjacent to level 1 vertices and not previously assigned to a level, are placed into level 2, and so on. The BFS traversal terminates when every vertex has been visited.

III. ALGORITHM

The system makes use of an embedded system based on the GSM technology. An interfacing mobile is connected to the microcontroller. When a person attempts fuel theft then the microcontroller commands the GSM modem to send a text message as an alert to the vehicle owner and further an alarm is raised by the buzzer installed within the system. In this system we interfaced the microcontroller AT89S52 with SIM 900A modem to decode the message.

Input: A graph G and a vertex v of G

Output: All vertices reachable from v labeled as discovered

procedure BFS(G,v) is

let Q be a queue

Q.enqueue(v)

label v as discovered

while Q is not empty

v ← Q.dequeue()

for all edges from v to w in G. adjacent Edges(v) do

if w is not labeled as discovered

Q.enqueue(w)

labelw as discovered

IV. PROPOSED SYSTEM

Proactive source routing protocol is proposed to provide opportunistic data forwarding in MANETs. As per PSR each node maintains a breadth-first search

Spanning tree of the network rooted at itself. This information is periodically exchanged among neighboring nodes for updated network topology information. Thus, PSR allows a node to have full-path information to all other nodes in the network, although the communication cost is only linear to the number of the nodes. This allows it to support both source routing and conventional IP forwarding.

BFST of the entire network is maintained at each node of the network. For this nodes periodically broadcast the tree structure to their best knowledge in each an iteration. Based on the information collected from neighbors during the most recent iteration, a node can expand and refresh its knowledge about the network topology by constructing a deeper and more recent BFST. This knowledge will be distributed to its neighbors in the next round of operation. On the other hand, when a neighbor is deemed lost, a procedure is triggered to remove its relevant information from the topology repository maintained by the detecting node.

V. EXPERIMENTALDESIGN

In PSR, Entire network is considered 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). Two nodes u and v are connected by edge e = (u, v) ∈ Eif they are close to each other and can directly communicate with given reliability. 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}.

Once network is considered as graph, PSR follows the below steps to manage BFST of the network.

A. Route Update

The update operation of PSR is iterative and distributed among all nodes in the network. At the beginning, node v is only aware of the existence of itself; therefore, there is only a single node in its BFST, which is root node v. By exchanging the BFSTs with the neighbors, it is able to construct a BFST within N [v], i.e., the star graph centered at v, which is denoted Sv .In each subsequent iteration, nodes exchange their spanning trees with their neighbors. From the perspective of node v, toward the end of each operation interval, it has received a set of routing messages from its neighbors packaging the BFSTs. Including those from whom v has received updates in recent previous iterations, node v has a BFST, which is denoted Tu , cached for each neighbor u ∈ N (v). Node v constructs a union graph, i.e.,

v = Sv∪(Tu − v).................(1)

Here, we use T − x to denote the operation of removing the subtree of T rooted at node x. As special cases, T − x = T if x is not in T , and T − x = ∅ if x is the root of T . Then, node v calculates a BFST of Gv , which is denoted Tv , and places Tv in a routing packet to broadcast to its neighbors.

B. Neighborhood Trimming

The periodically broadcast routing messages in PSR also double as “hello” messages for a node to identify which other nodes are its neighbors. When a neighbor is deemed lost, its contribution to the network connectivity should be removed; this process is called neighbor trimming. Consider node v. The neighbor trimming procedure is triggered at v about neighbor either by the following cases:

1) No routing update or data packet has been received from thisneighbor for a given period of time.

2) A data transmission to node u has failed, as 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.,

Gv = Sv ∪w∈N (v)(Tw − v) ....(2)

3) finally, computing BFST Tv .

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

C. Streamlined Differential Update

In addition to dubbing route updates as hello messages in PSR, we interleave the "full dump" routing messages, as stated previously, with "differential updates." The basic idea is to send the full update messages less frequently than shorter messages containing the difference between the current and previous knowledge of a node's routing module. Both the benefit of this approach and balancing between these two types of messages have been extensively studied in earlier proactive routing protocols. In this paper, we further streamline the routing update in two new avenues. First, we use a compact 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 even shorter.

VI.EXPECTEDSIMULATIONRESULTS

We compare the performance of PSR with that of OLSR, DSDV, and DSR. The reasons that we select these baseline protocols that are different in nature are as follows. On one hand, OLSR and DSDV are both proactive routing protocols, and PSR is also in this category. On the other hand, OLSR makes complete topological structure available at each node, whereas in DSDV, nodes only have distance estimates to other nodes via a neighbor. PSR sits in the middle ground, where each node maintains a spanning tree of the network. Furthermore, DSR is a well-accepted reactive source routing scheme, and as with PSR, it support source routing, which does not require other nodes to maintain forwarding lookup tables. All three baseline protocols are configured and tested out of the box of ns-2.

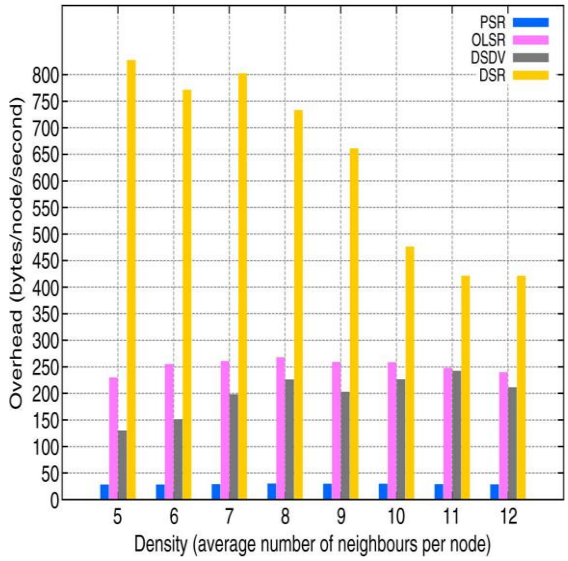

Fig 1. Routing overhead with density

TCP With Velocity:

We also study the performance of PSR and compare it to OLSR, DSDV, and DSR with different rates of node velocity. In particular, we conduct another series of tests in networks of 50 nodes deployed in a 1100 × 1100 (m2) square area with vmax set to 0, 4, 8, 12, . . . , 32 (m/s). The network thus has an effective node density of around seven neighbors per node,i.e., a medium density among those configured earlier. As with before, 20 TCP one-way flows are deployed between 40 nodes, and we measure the routing overhead, TCP throughput, and end-to-end delay.

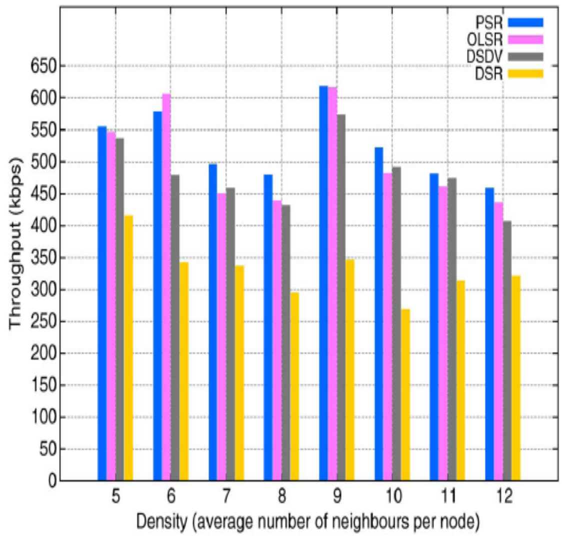

Fig 2. TCP throughput with density

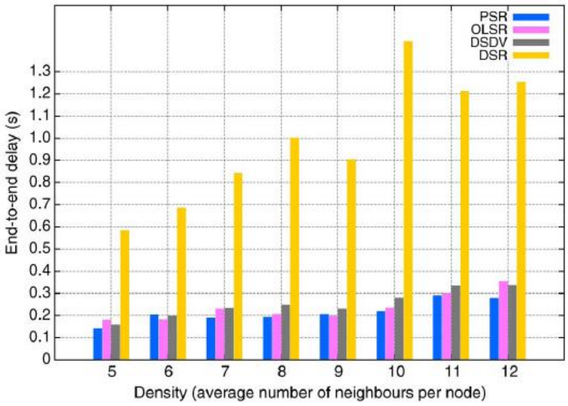

Fig 3. End-to-end delay in tcp with density

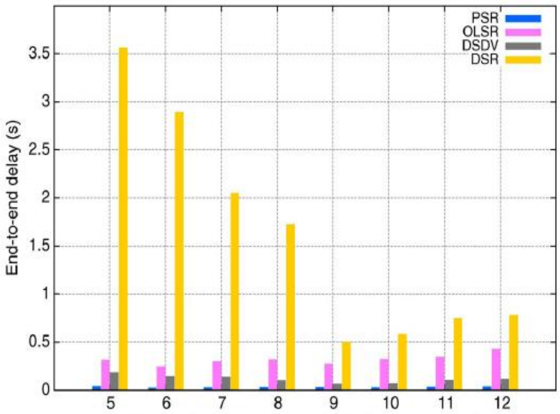

Fig 4. End-to-end delay in UDP with density

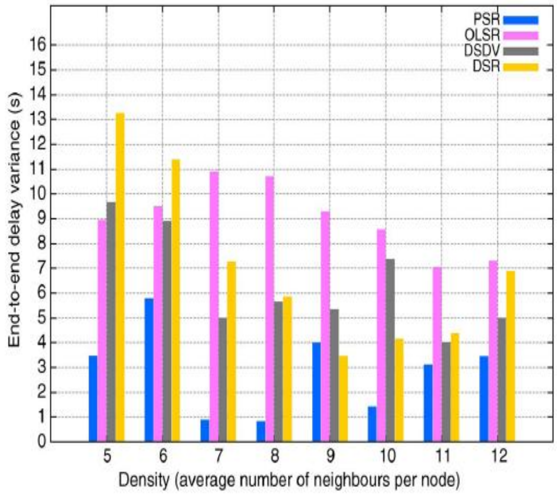

Fig 5. End-to-end delay jitter in UDP with density

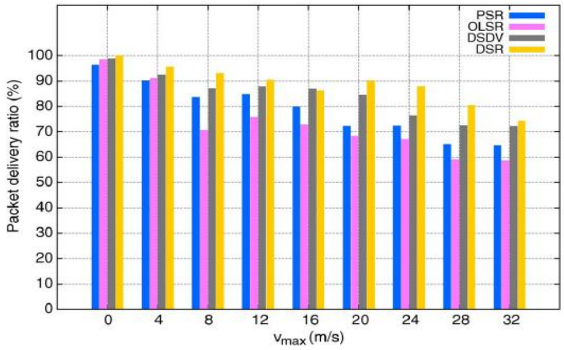

Fig 6. PDP in UDP with velocity

VII. CONCLUSION

In this paper, we pro-pose a lightweight proactive source routing (PSR) protocol. PSR can maintain more network topology information than distance vector (DV) routing to facilitate source routing, although it has much smaller overhead than traditional DV-based protocols [e.g] destination-sequenced DV (DSDV)], link state (LS)-based routing [e.g] optimized link state routing (OLSR)], and reactive source routing [e.g., dynamic source routing (DSR)]. Our tests using com-puter simulation in Network Simulator 2 (ns-2) indicate that the overhead in PSR is only a fraction of the overhead of these baseline protocols, and PSR yields similar or better data transportation performance than these baseline protocols.

REFERENCES

[1] I. Chlamtac, M. Conti, and J.-N. Liu, “Mobile ad hoc networking: Imperatives and challenges,” Ad Hoc Netw., vol. 1, no. 1, pp. 13–64, Jul. 2003.

[2] 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. 2012.

[3] R. Rajaraman, “Topology control and routing in ad hoc networks: A survey,” ACM SIGACT News, vol. 33, no. 2, pp. 60–73, Jun. 2002.

[4] 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. 2009.

[5] P. Larsson, “Selection diversity forwarding in a multihop packet radio net-work with fading channel and capture,” ACM Mobile Comput. Commun. Rev., vol. 5, no. 4, pp. 47–54, Oct. 2001.

[6] S. Biswas and R. Morris, “ExOR: Opportunistic multi-hop routing forwireless networks,” in Proc. ACM Conf. SIGCOMM, Philadelphia, PA,USA, Aug. 2005, pp. 133–144.