Smart Route Discovery in MANET using Ordered Walk

Sivapriya G
M.TECH (CSE)
B.S Abdur Rahman University
Chennai(Tamilnadu),India

Senthil Kumaran T
M.E (Ph.D)
B.S Abdur Rahman University
Chennai(Tamilnadu),India

ABSTRACT

The notion of ordered walks is acquaint with a depth-first search (DFS) that does not rely on topographical or virtual coordinate information and is much more efficient than meager random walks. The welfare of using DFS as the building block of the signalling in MANET routing protocols are epitomized by the introduction of the Ordered Walk Search Algorithm (OSA) as a replacement of flooding, which is used as part of the Ordered Walk with Learning (OWL) protocol. Aim to take advantage of the smaller time complexity of BFS and combine it is the low communication complexity of DFS to further improve the efficient of the search through the use of known topology information. The uses of multiple DFS can lead to a quicker discovery of OSA, The present the ordered walks with learning (OWL) routing protocol, which use DFS to establish and repair paths from the sources to the destination with minimal signalling overhead and fast convergence. OWL performs one or multiple ordered walks to search for destination. Simulation experiments are used to compare the delivery and end-to-end delays of OWL and AODV, but with significantly less overhead. The use of ordered walks is a promising tool in achieving limited-signalling routing in MANETs.

Keywords

Mobile Ad-hoc Networks; Ordered Walk with Learning protocol; Ordered walk Search Algorithm; Depth First Search.

1. INTRODUCTION

Over the past few years, many routing protocols have been projected using various mechanisms to construct, maintain and repair paths between nodes in a network. In both proactive and on-demand solutions, overhead packets can be propagated to regions where they have no implication but nonetheless consume inadequate network resources. In meticulous, on-demand routing protocols can initiate a breadth-first search (BFS) by flooding the entire network to discover a node that may only be a few hops away. Such an ineffective use of network resources cannot be part of an efficient solution to routing in MANETs.

Section 2 summarizes prior work aimed at reducing the signaling overhead incurred in routing protocols for MANETs. As the succinct survey indicates, the prior work has been aimed at reducing the number of nodes engaged in signaling of routing algorithms based on breadth-first search (BFS), reducing the amount of breadth first searching signaling information that must be dispersed, or establishing virtual topologies that may be maintained more efficiently.

As an alternative, depth-first search, has been studied extensively in the past and many distributed algorithms for DFS have been reported (e.g., [4], [5], [6]). Surprisingly, however, DFS has not been used much to support the signaling of routing protocols in MANETs. To the knowledge, the only efforts that have addressed DFS focus on random walks [7], [8] or routing using location information (e.g., GPSR [9]).

BFS schemes would flood the network, or at least a very large number of nodes to find a destination that is far away. If too many nodes are performing breadth first searching simultaneously, the routing overhead can saturate the network making it impossible to deliver any packets. Route computations based on DFS involve a much smaller number of nodes and can incur significantly lower overhead resulting in much less disruption than BFS and therefore more efficient routing as the size of the network and the number of flows increases. However, a DFS scheme may incur much longer delays in finding the desired routes to the destination.

The main participation of this paper is to show that (1) routing in MANETs using depth-first search (DFS) is not only feasible but can in fact be much more proficient than routing based on breadth-first search (BFS) schemes, and (2) proficient DFS-based routing does not need to depend on geographical or virtual coordinate information.

Section 3 motivates the use of DFS instead of BFS as the basis for route signaling in MANETs in more detail. Apart from ordered walks, to the knowledge, the only efforts that have deal with depth first searching focus on random walks [7, 8], routing using location information [9], and protocols that establish and maintain virtual topologies using DFS, such as Virtual Ring Routing (VRR) [10].

Section 4 presents the ordered walk search algorithm (OSA) and analyzes its potential in terms of its time complexity and the signaling concerned. Instead of performing a search in a completely random manner, or assuming knowledge of the relative position of the destinations, OSA distributively constructs an approximated minimum spanning tree. Then ordered walk search algorithm (OSA) searches this tree efficiently such that the number of search messages is minimized and the resulting path is of reasonable length.

Section 5 presents the Ordered Walk with Learning protocol (OWL), an example of efficient routing in MANETs based on depth first searching (DFS) without the need for location information. OWL uses ordered walk search algorithm (OSA) to establish routes on demand based on DFS.

Section 6 present wide results of simulation experiments illustrating that, in terms of, end-to-end delay and the signaling overhead incurred, OWL attains better performance than popular on-demand routing protocols based on BFS.. Section 7 presents the concluding explanation of the study.

2. RELATED WORKS

On demand routing schemes were developed to reduce routing overhead in mobile ad-hoc networks. Today, on demand routing protocols have become synonymous with the flooding of route requests (RREQs) when a path needs to be established. While this approach may be the fastest result in a network that is not bandwidth-limited, it directs to the broadcast storm problem as identified by Ni et. al.[11], especially in volatile routing environments.

This inefficiency has been acknowledged by many in the past, and several optimizations over this blind flooding have been proposed. These approaches include the use of heuristics based on connected dominating sets to reduce the number of nodes retransmitting packets in the network [12], the use of geographical information to direct the flooding [9] and probabilistically sinking the number of retransmissions [11]. While these schemes improve the broadcast-storm problem, they do not address the need for flooding which is inherent in any BFS approach.

Connected dominating sets make most logic in dense networks where there would or else be too much redundant broadcast of control packets but sparse networks would see little benefit from such infrastructure. Apart from ordered walks, there have been only a few efforts to solve the problems incurred with flooding by using DFS instead of BFS.

These approaches have focused on the use of random walks [7,8] in which a route request(RREQs) starts at the source and travels along a single path, composed of successive random next-hop choices in the search for the destination. The limitation of such a brute force application of DFS is that the communication complexity incurred in reaching destinations may be comparable to that of flooding, but with much longer delays. The prosperous application of DFS to route discovery requires more creative solutions.

3. MOTIVATION FOR USING DFS

Breadth first search has remained the most accepted choice for on-demand route discovery in MANETs because, BFS is the fastest approach to establishing the preferred routes requiring on average O(logk N) time, where k is the average network connectivity. However, the cost paid for this search speed is the signaling overhead incurred, which is O(N) in a network with broadcast links. In practice, given that MANETs are bandwidths limited; this large communication complexity means that breadth first search may not succeed at establishing routes quickly, because signaling packets may endure long queuing delays or even losses due to multiple access interference. In proactive routing protocols, the time complexity for establishing routes is O(1) as routes are readily available, but this comes at the signaling complexity $O(N^3)$ as every node broadcasts its topology table (of size $O(N^2)$) in the worst case.

DFS may be a feasible option to evade the problems introduced by the broadcast storm associated with flooding. However, applying a DFS strategy to on-demand routing means that route requests (RREQs) are propagated from one node to a single neighbor and thus travel a single path from the source to the destination. Accordingly, using depth first searching (DFS) in a graph in which each step is completely random means that the time required to find a path to a given destination is O(N), which is the case of the destination being the last node searched, and the average complexity would be O(N/2) for both time and number of messages. Hence, compared to breadth first search (BFS), Depth first search (DFS) offers only a constant factor improvement but with a very huge penalty in the time complexity it deserves. Undoubtedly, depth first search proposal based on random walks make sense only in networks where bandwidth is at a premium and delays in finding paths are not momentous, which may be the case in some sensor networks with fixed topologies.

An advocate a novel approach to depth first search (DFS) applied to on demand routing in MANETs that takes benefit of two important characteristics of MANETs. Firstly, some local topology information is readily available to nodes due to the broadcast nature of radio links. In particular, a node can hear over time about the presence of neighboring nodes, and even the presence of its neighbors' neighbors. Secondly, MANETs are not completely random and, more importantly, source-destination dialogues follow patterns of interest, which means that sources will be able to find destinations more effectively, over time, than by random walking if their searches choose ''children'' in the DFS tree based on this prior knowledge that incorporates information gathered from past searches, some of which may have even failed. Accordingly, an approach to using DFS in on demand routing is based on the concept of ''ordered walks,'' which describe in the next section, and complement this DFS approach with learning gained from prior walks.

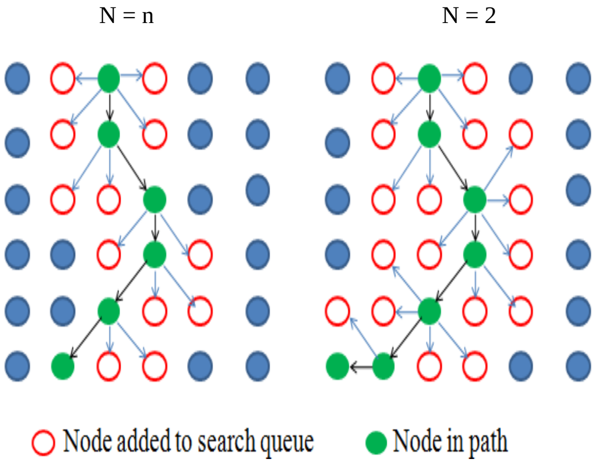

Figure 1. DFS pruning in OSA

4. ORDERED WALKS

With the ordered-walk search algorithm (OSA), an aim to take benefit of the smaller time complexity of breadth first search (BFS) and combine it with the small communication complexity of depth first search (DFS) to further improve the efficiency of the search through the use of known topology information. The fundamental idea is to approximate the edifice of a minimum spanning tree rooted at the source (as in BFS) and then perform depth first search (DFS) on this tree.

4.1 Approximating Minimum Spanning Tree

Given overall topology information, any node can construct a minimum spanning tree in an associated network. The source node becomes the root for the search. Nodes connected to the source are then at depth 1(i.e. One hop neighbor), their neighbors not yet incorporated are at depth 2 (i.e. Two hop neighbor), and this process continues until all nodes are added. Performing the search on a minimum spanning tree results in paths with a shorter predictable length than those obtained with a random walk search. Providentially, such a tree can be approximated and constructed distributively involving only two hop neighborhood information.

The initial step is set up the n-hop neighborhood. Figure. 1 shows n-hop neighborhood(i.e. N=n). This is be done by having nodes advertise distances to all other nodes inside (n - 1) hops. A node getting such an advertisement would store the distance to every node in its n-hop neighborhood from each of its neighbors in a routing table. By maintaining only the n-hop topology decreases the complication of the signaling. At each step in the search, say at depth x, a node must choose children that are at a greater depth (x +1) in the search tree than the node itself. If n = 2(i.e. two-hop neighbors), the reference point will be the previous hop in the search path and can prune all the 1 hop neighbors of this node.

4.2 The Ordered Walk Search Algorithm

In an ordered walk search algorithm (OSA), the mobile hosts first exchange their node sets of one-hop neighbor by the hello message. Then, each mobile host selects a subset of its one-hop neighbor information in such a way that the subset can cover all the two-hop neighbor information.

It decreases the number of forwarding because it is only a subset instead of all the one-hop neighbor nodes. In this case, each mobile host can build a minimum spanning tree consisting of all the neighbor nodes in its two-hop list. When the source node wants to transmit a datagram to a destination, the proposed protocol OWL initiates the route discovery procedure.

4.3 Precursor Lists and Temporary Cache

1. One-hop neighbor list: each mobile node needs to maintain a one-hop list for neighbor detection in the periodic exchange of hello messages. This list specifies the information about the list of neighbors.

2. Two-hop list: each mobile node also needs to preserve a two-hop list through the periodic exchange of the one-hop list of neighbor interfaces by hello messages. Upon receiving a hello message from a neighborhood, a mobile node should update its two-hop neighbor list. A two-hop neighborhood is neither the node itself nor a neighbor of the node, but is a neighbor of its neighbor.

3. The hops-compared cache: To observe the discovered path, each mobile node needs to maintain a hops-compared cache. The format of each and every entry in the array is (src_addr, dst_addr, hop_cnt, neig_addr), where src_addr is the source address, dst_addr is the destination address, hop_cnt is the hop count, and nei_addr is the neighbor's address. Each of the elements of the array has an expiration period after which they are nullified. If the array becomes full with valid entries, the oldest entry should be restored.

Algorithm 1 OSA (G, s, d)

```
Begin
    For each node pair $(S, D)_i$, i=1 to (N-1) Do
        /*D=2, 3, 4,..N*/
        PN= S; Loop=0; $Path_i$=Ø;
        /* S: source node; D: destination node; PN: PN nodes of the network; Path: output path set generated for node pair, initially set to be an empty Ø */
        Call Procedure PATH(input: PN, D; output: $Path_i$)
End;

Procedure PATH (input:PN,D; output: $Path_i$)
    Begin
        If (PN= D) Then
            output $Path_i$;
            Return;
        Else
            If (PN is not in $Path_i$) And (Loop <Hops) And (PN's two-hop   list does not contain D)
    /* Hops: number of hops referred */
    Then
    Begin
            Loop=Loop+1;
            Add PN to $Path_i$
            For each neighboring node N of node PN Do /* N: the neighbor of PN */
                PATH (N,D, $Path_i$)
      Loop = Loop - 1
    End;
End;
```

5. ORDERED WALK WITH LEARNING PROTOCOL (OWL)

5.1 Neighbor Detection

At the start, every mobile node exchanges a hello message to its neighbors on the network. OWL can collect and exchange the information of one-hop lists and construct a tree structure in the routing table of the source. The address of an interface of a neighbor node can be identifying directly from the exchanged information. Upon receiving a hello message, a mobile host should update its one-hop list. The mechanism for neighbor detection is the periodic exchange of hello messages. A mobile host can determine connectivity by listening for packets from its set of neighbors. Whenever a mobile host receives a hello message from a neighbor, the mobile host makes sure that it has an active route to the neighbor and creates one if necessary. If a route already be present, then the life span for the route should be increased. The route to the neighbor, if it be present, must consequently contain the newest destination sequence number from the hello message. The current mobile host can now begin to use this route to forward data packets.

5.2 Preferred Neighbor Selection

In an OWL protocol uses OSA algorithm for selecting preferred neighbor (PN). The source node first checks its two-hop neighbor list. If the destination node is in its two-hop list, then the datagram is transmitted by following the routing table’s path. If the destination node is not in its two-hop list, the source node uni-casts the Route Request (RREQ) to the preferred neighbor (PN) on the network. When the preferred neighbor (PN) receives this RREQ packet, the preferred neighbor (PN) also checks their two-hop list. If the destination node is in their two-hop list, then the preferred neighbor (PN) forwards directly the RREQ to the destination host. The destination host replies with a Route Reply (RREP) which follows the RREQ return path to the source node. If the destination node is not in their two-hop list, then it modifies the sequence-number and hop-count, and forwards this RREQ to the network. The procedure is repeated until it finds the destination node.

After the route discovery, data packet is transmitted to the destination. Each packet contains the <src_addr, dst_addr, hop_cnt, neig_addr> in their header, and then the mobile nodes monitor each of the received traffic packets in order to compare the headers. Meanwhile, the OSA can reduce the overhead and automatically find the shortcut path. Hence, this new protocol can evade the flooding packets and adjust the path to the shortest path.

5.3 A Simple Example

Consider the simple example in Figure. 2, where source node S requests to send data to destination node D. S chooses a neighbor with the highest neighbors count in common with itself. In this example, it first chooses X, which in turn selects Y as its successor because they have the highest neighbors count in common. Node Y does not have any two-hop neighbor that is not a child of a node already in the search path because nodes T and U are neighbors of node X. Node Y cannot proceed to a greater depth of the search tree so it must send a route error (RERR) to node S. Upon receiving this RERR, node S must choose a new successor with as few neighbors in common with both itself and node X.

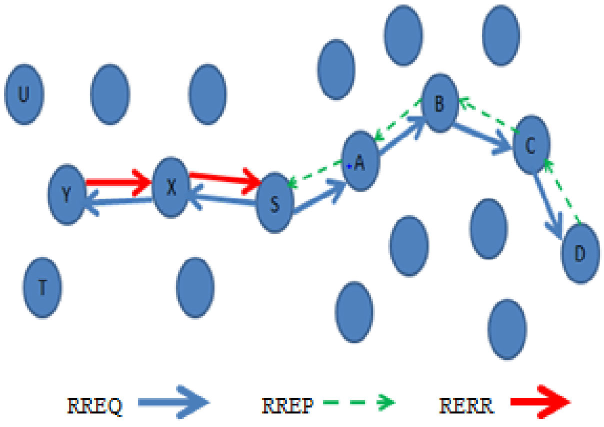

Figure. 2 An OWL example

Node D then initiates a RREP in response to the RREQ and this route reply travels along the identical path as the route request. Nodes close to this path, such as nodes E and F, listen in the route replies (RREP) and become active for destination D.

6. EXPERIMENTAL RESULTS

The compared performance of OWL with that of representative protocols for on-demand BFS in MANETs. AODV are used as example of on-demand routing. Routing in AODV is incremental, meaning routing decisions are taken on a hop by hop basis source routing. Additionally, the comparison highlights the ability to attain on-demand routing using a DFS approach without depending on location information, and exemplifies the fact that such an approach can render comparable results attained with the traditional BFS scheme used in AODV, but with only a fraction of the signaling overhead.

6.1 Simulation Environment

Scenario consists of 100 nodes uniformly distributed in a grid of size l500m x 1500m with the transmission range of the radios set to 250m. This choice of parameters satisfies the minimum standards for meticulous MANET protocol evaluation as prescribed by Kurkowski, et al [16] as it results in an average shortest path hop count [16] of 4.03 and average network partitioning [16] of 3.9%. Other relevant simulation parameters are summarized in Table 1 and Table 2. This scenario ensures that packets travel several hops from source to the destination and thus tests the robustness of the protocols. The mobility model chosen was that of random waypoint with minimum speed of lm/s and maximum speed of l0m/s with pause time of 30s. An experiment lasted for 900s.

Ten nodes were chosen at random to be sources of CBR flows and ten nodes were selected at random to be destination of these flows. Care was taken to avoid the case where a node was both the source and destination of any particular flow. There were no restrictions on nodes being multiple sources, multiple destinations or a source node of one flow and a destination node of another. Each source would send a maximum of size 512 Bytes at a rate of 4 packets per second. The start time of each flow was randomly determined using a uniform distribution and was within the duration of the experiment. Simulation results shown in Table 3.

Table 1 Constant simulation parameters

Simulation Parameter	**Parameter Values**
Simulator Used	NS-2 Version 2.34
Packet size	512 bytes
Topology size	1500*1500m^2
Maximum Speed	30 m/s
MAC Protocol	IEEE 802.11
Simulation Time	900s
Traffic Type	CBR

Table 2. Varied simulation parameters

Parameter	**Scenario A**	**Scenario B**
Number of nodes	100	100
Number of flows	5,10,15,20CBR	10 CBR
Avg.neighbor count	5.02 nodes	9.4 nodes
Transmission range	150m	250m
Pause time	10,20,30s	30s

Table 3. Simulation results

Scenario A	Delivery ratio	Latency	Net load
AODV	0.53	0.035	36.2
OWL	0.62	0.052	2.8
Scenario A	Delivery ratio	Latency	Net load
AODV	0.74	0.028	15.5
OWL	0.89	0.041	1.1

Using a time-based seed, 20 random scenarios were created with the above specifications and the results were used to compare the performance of the OWL with AODV protocols. The large number of randomly generated scenarios was used to avoid bias in the results.

Scenario B is similar to A, except that the transmission range is increased to 250m. The purpose of this was to increase the average neighbor count from 5.02 to 9.4 nodes. Also, the enlarged range makes some of the links more stable as nodes take longer to move out of range of each other. The value of the average shortest path would certainly be less than 4 nodes while the average network partition would be less than 4%.

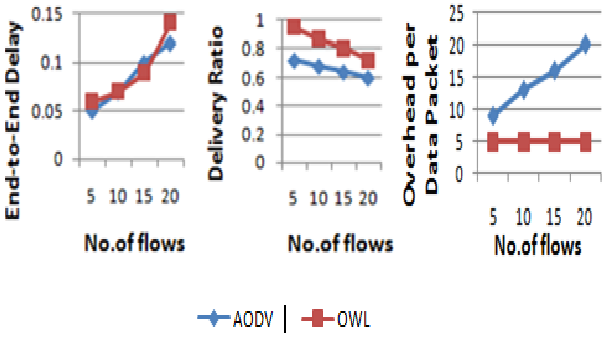

Figure. 3 Performance variation with Number of flows

The advantages of ordered walks become quite clear when this metric is considered. OWL requires considerably less overhead packets than the other protocols, because it efficiently searches the network without flooding and accordingly sources find their destinations without having to search every node. Even in the large network of 100 nodes, OWL incurs ten times less overhead than AODV which vouches for its scalability.

7. CONCLUSION

The disputed that most routing proposal designed for MANETs depends on some form of breadth first search (BFS), and presented the ordered walk search algorithm as a alternate for flooding. An ordered walk is a disseminated approximation of depth first search (DFS) that is assisted by known topology information to reduce the search tree. The introduced the OWL protocol as an example of the great probable for using DFS in route signaling for MANETs. We presented the results of simulation experiments demonstrating that OWL provides comparable or better delivery and end-to-end delay than AODV, but with considerably less signaling overhead. The use of ordered walks, as presented in this paper, is a promising tool in achieving minimum-signaling routing in MANETs. While OWL is a step towards attaining this goal, more work is needed to fully utilize the advantages of ordered walks in routing protocols.

8. REFERENCES

[1] Dabideen, S., & Garcia-Luna-Aceves, J. J. Efficient routing in MANETs using ordered walks. Springer Science Business Media, 2011.

[2] Muneer, B., Sanabel, F. and Ahmed, Y., “A New Dynamic Counter Based Broadcasting Scheme for Mobile Ad-hoc Networks”, Proceedings of Simulation modeling and theory, Elsevier, Vol. 19, No. 1, pp. 553 – 563, 2011.

[3] Muneer, B., Mustafa, B. and Ahmed, Y., “A new probabilistic broadcasting scheme for mobile ad hoc on-demand distance vector networks(AODV) routed networks”, Journal super computer, Springer, Vol. 53, No. 1, Mar.2010, pp. 196 – 211.

[4] Dabideen, S., & Garcia-Luna-Aceves, J. J. OWL: Towards scalable routing in manets using depth-first search on demand. Sixth IEEE international conference on mobile ad-hoc and sensor systems, 2009.

[5] Aminu, M., OuldKhaoua, M., Mackenzie, L.M. and Abdulai, J., “Dynamic probabilistic counter-based broadcasting in mobile ad hoc networks”, Proceedings of

the 2nd IEEE International Conference on adaptive science and technology, 2009.

[6] Jamal-Deen Abdulai , Mohamed Ould-Khaoua,Lewis M. Mackenzie, ‘Adjusted probabilistic route discovery in mobile ad hoc networks’, The Journal Computers and Electrical Engineering, vol.35,Issue.1, 2008 .

[7] Acer, U., Kalyanaraman, S., & Abouzeid, A. A. Weak state routing for large dynamic networks. In Proceedings of MobiCom, September, 2007.

[8] Beraldi, R., Querzoni, L., & Baldoni, R A hint based probabilistic protocol for unicast communications in manets. Elsevier Ad Hoc Networks, 4(5), . 2006.

[9] Tian, H., Shen, H., & Matsuzawa, T. Random walk routing for wireless sensor networks. In Proceedings of PDCAT, 2005.

[10] Philippe, J. Anis, L. Pascale Minet, and Laurent, V. “Performance of Multipoint Relaying in Ad Hoc Mobile Routing Protocols”, Proceedings of 7th International Symposium on Parallel architectures, Algorithms and Networks, pp. 539–543, 2004.

[11] Wu, J. Wei Lou, and Dai, F. “Extended Multipoint Relays to determine Connected Dominating Sets in MANETS”, IEEE International Conference on Networking Vol. 1, pp. 100–105, 2004.

[12] Perkins, C. E., & Royer, E. M. Ad hoc on-demand distance vector routing. IETF Internet draft, October, 2003.

[13] Servetto, S. D., & Barrenechea, G. Constrained random walks on random graphs: Routing algorithms for large scale wireless sensor networks. Proceedings of WSNA, 2002.

[14] Ni, S.-Y., Tseng, Y.-C., Hen, Y. S. V., & Sheu, J.-P. The broadcast storm problem in mobile ad-hoc networks. In Proceedings of MobiCom, 2001.

[15] Yashar, G. and Abtin Keshavarzian :”Study On Load Balancing in Mobile Ad-hoc Networks”, Proceedings of International Conference Communication (ICC’00) Vol.1,pp. 74-79, 2000.

[16] Wei, C. and Yi Zhang “A Multi-constrained Routing Algorithm Based on Mobile Agent for MANET Networks”, IEEE Selected Areas Communication Vol. 17, No. 8, pp. 1369–1379, 1999.