Volume 5, Issue 5, May 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Survey of Link Reversal Routing Algorithms for Mobile Ad hoc Networks with Different Densities and Mobility

Asst. Prof. Nandita Goyal, Aman Kaushal, Ravi Kumar
Department IT, ABES Engineering College
Ghaziabad, India

Abstract: This paper presents analysis of performance of link reversal algorithms by taking into consideration the dependence of performance on parameters like network density and network mobility. Link reversal (LR) algorithms provide a simple mechanism for routing in such types of communication networks whose topology is frequently changing, such as mobile ad hoc networks. A Link Reversal algorithm routes by imposing a direction on each network link such that the resulting graph is destination oriented (DAG).Whenever a node loses routes to the destination, is reacts by reversing some (or all) of its incident links. There may exist network scenarios where an algorithm which is performing badly in one scenario may perform better in another scenario due to the variation in network density and network mobility.

Index Terms –MANETs, Link reversal, Network Density, Mobility.

I. INTRODUCTION

The term MANETs is becoming very popular these days. Mobile ad hoc networks (MANETs) are an infrastructure-less, dynamic network consisting of a collection of wireless mobile nodes that communicate with each other without the use of any centralized authority and their location is not fixed. Mobile ad-hoc networks are becoming popular as they are less expensive and easy to be built in comparison to infrastructure based networks. One of the important problems to be addressed in mobile ad-hoc networks is to provide a routing algorithm which performs well in these types of networks. Because of the mobility of the communication partners and because there is no dedicated hardware which is fixed, we cannot maintain and use global routing tables. Instead all communication nodes have to cooperate to send a message from one node to another. Through the movement of the communication nodes, nodes may leave or join the network. Also it will happen that existing links between two nodes break down due to weak radio signals and movements or that new links will arise. Link reversal algorithm is a design technique which is used in numerous distributed systems and network [1]. These are adaptive, efficient, scalable, local responses to changes. They guarantee loop-free routes always. They also provide multiple routes: this alleviates congestion, and also means that many changes don't require any reaction at all. They establish routes quickly. There is low communication overhead; nodes maintain information only about neighbours. The reactions to topological changes are local. This algorithm provide a simple mechanism for routing in communication networks whose topology [2] is changing frequently, such as in mobile ad hoc networks. A common feature of link reversal algorithms is a directed acyclic graph (DAG) structure in which the vertices represent computing nodes executing the algorithm, and the directions of the edges are reversed by the nodes under certain conditions in order to achieve some property required by the problem specification. In link reversal algorithms each computing node is responsible for reversing its incident edges only when a particular local property of the node is satisfied. The goal of link reversal algorithms is to eventually satisfy some global property in the entire system. The first link reversal routing algorithms were introduced by E.M. Gafni and D.P.Bertsekas in 1981 [1].

Many routing algorithms for mobile ad-hoc networks have been proposed and enhanced[5] [6]. But it is not clear that,how different algorithms will behave in different environments. An algorithm may be best in a special network but worst in another. Therefore our goal is to classify network and mobility scenarios and to analyze howdifferent algorithms behave in these scenarios. For classification we have used two parameters network density and network mobility and we have analysed full link reversal and partial link reversal algorithms in terms of number of reversals required, path length and storage overhead. This paper describes our approach for analysis and classification of link reversal algorithms. The organization of this paper is as follows: In section II, we will give an overview about Link reversal algorithms. In section III, we will discuss the parameters namely network density and network mobility which can be used for the classification and analysis of link reversal algorithms. In section IV, we will discuss the metrics for performance evaluation of link reversal algorithms. Section V will discuss the classification of scenarios for performance evaluation of link reversal algorithms. The paper is concluded in section VI.

II. LINK REVERSAL ALGORITHMS

A. Full Reversal Algorithm

The full reversal algorithm is simple and straight forward if mobile adhoc network are converted to a 1destination oriented graph (DAG). This algorithm works in a very simple manner such as, whenever a node becomes a sink; it reverses its entire incoming links. One of the advantages of the full reversal algorithm is that it does not need to update at every link failure. "If a link gets lost for some reason, there might be still another directed path to the destination from every node and so the graph could still be destination oriented" [1].This algorithm works very suitably but one main problem is, the link reversal due to single node, reverses all of the connected nodes due to which overhead in the network increases. The algorithm was implemented using heights. At the beginning each node has an initial height. If a route request arises then the destination sets its height to 0. If a node becomes a sink it raises its height to the height of the highest neighbour + 1. This corresponds to reversing all links. It was shown that the algorithm always terminates and generates no loops [9].

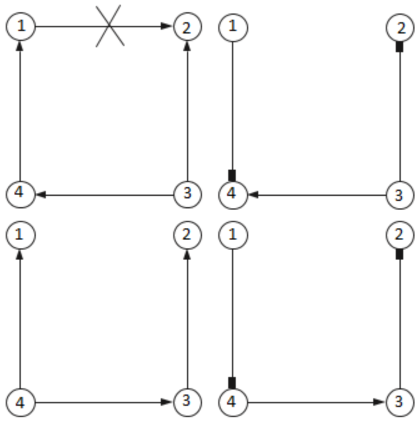
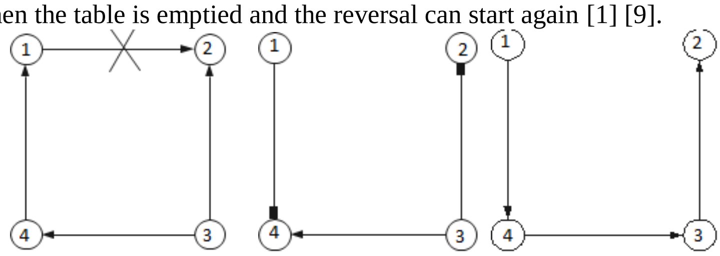


Fig. 2.1

When node 2 sinks all the nodes connected to it (node 1 and node 3) get reverse. Due to which node 2 will have path from source to the destination.

B. Partial Reversal Algorithm

The partial link reversal algorithm is the updating of full link reversal algorithm [8].This algorithm works same as full link reversal algorithm, but in this algorithm whenever a node finds that it has become sink, it doesn't reverses all of the links connected to it, only links which are not reversed till now are reversed .Due to this property of link reversal an updated feature from the above algorithm, this algorithm is preferred and used widely nowadays. Partial link reversal algorithms have less overhead. Each node maintains a table. A link is inserted into the table if it is reversed. If all links for a node were reversed then the table is emptied and the reversal can start again [1] [9].



(1) (2) (3)

Fig. 2.2

III. NETWORK PARAMETERS

A. Network Density

The density λ, is defined as number of node in one unit disk. The open unit disk (or disc) around *P* (where *P* is a given point in the plane), is the set of points whose distance from *P* is less than 1.

The density may vary in the whole graph from unit disk to unit disk. Also if nodes are moving the density in disks change. Networks with low densities for example are sensor networks where the sensor devices are mobile but have a low transmission radius. Because of the low radius only few nodes are in a unit disk.

On the contrary there exist networks with high densities, for example a network for a conference room where many people communicate with mobile devices which have a high transmission radius. So, we can use the following density values:

λ= 11 for low density networks
λ= 20 for middle density networks
λ= 30 for middle density networks
λ= 40 for dense networks

B. Network Mobility

The fact that nodes move is probably the most important attribute of MANETs [3] [4]. Mobility dictates network and application level protocols[10]. The mobility model which we have considered is as follows:

Each node is in one of two states:
*State*1: the node will not move in the coming round
*State*2: the node will move to a new position in the coming round
If the node is in S*tate*1 then it will move with probability *p* from *State*1 to S*tate*2. With probability $r = 1 - p$, the node will stay in its state S*tate*1.
If the node is in S*tate*2 then it will move with probability *q* from S*tate*2 to *State*1. It will stay with probability $s = 1 - q$ in S*tate*2.
We have assumed that the following condition shall always hold:
$p << q$

We set q and p constants as:
q = 0.9
p = 0.02 for low mobility
p = 0.1 for high mobility
The probability to stay in *state*1 and *state*2 is 1 -*p* and 1 -*q* respectively.

The maximum distance factor for a node ε,
ε = .05 for low mobility
ε =0.1 for middle mobility
ε = 1.0 for high mobility
The maximum distance moved by a node is calculated as:
$dmax = \varepsilon * r_{UDG}$
Wherer_{UDG}*is* the radius of the unit disks in the graph.
The direction in which a node moves is chosen randomly.

IV. PERFORMANCEMETRICS

A. Path Length

The performance of a routing algorithm is defined and analyse in terms of the path lengths of the route generatedby the algorithm. The goal of each routing algorithm is to use as few nodes as possible for a routing request. If paths are shorter, then the energy consumption is lower and fewer collisions happen.

B. Storage Overhead

In mobile networks it is important that the power consumption is as low as possible. Also the network traffic should be as low as possible such that few collisions happen. Therefore it is essential that as few overhead messages as possible are sent over the network. We require an algorithm that not only provides short routes but also uses as few overhead messages as possible.

C. Number of reversals required

Total number of link reversals required till stabilization of the network. The algorithm will lesser number of reversals generates lower overhead and is therefore considered better.

V. CLASSIFICATION OF SCENARIOS

A. Dense Networks with High Mobility:

For these type of networks we haveconsidered the following parameters;
For dense networks we have assumed the value of density and distance,
ε= 1.0
p=0.1
ʅ>=30

Where p is the probability of the node to move from State1 to State2,ε is the distance vector form and ʅ is the density of the network. Here partial link reversal algorithm will perform better than full link reversal algorithm on the basis of number of reversals and Storage overhead as for dense networks there will be large number of reversals by full link reversal algorithm due to which overhead will increase. On the basis of path length both full reversal and partial reversal algorithms will give approximately same results. An example for this type of network is - person communicating through a moving platform using mobile devices.

B. Sparse Network with High Mobility:

This class of network can be studied using following parameters,
ε= 1.0
p= 0.1
ʅ<30
Here, for sparse networks with high mobility partial link reversal algorithm and full link reversal algorithm perform approximately similar but the storage overhead for partial link reversal algorithms will be less.

C. Dense Network with Low Mobility:

This class of network can be studied using following parameters,

ε<= 0.1

p= 0.02

ʅ>=30

Here,partial link reversal algorithm and full link reversal algorithm perform approximately similar

D. Sparse Network with Low Mobility:

This class of network can be studied using following parameters,

ε<= 0.1

p= 0.02

ʅ<30

Here, for sparse networks partial link reversal algorithm and full link reversal algorithm perform approximately similar.

VI. CONCLUSION

In this paper we have done a theoretical analysis of performance of link reversal algorithms taking into account two network parameters namely network mobility and network density.Many interesting points still remain which could be analysed in future. It would be interesting to know how algorithms other than Link reversal algorithms would behave in the same environments. If a dependence on either mobility or network density could be found, simulated and proved one could try to combine different routing algorithms to form a new one which is better as well as independent of these properties.

REFERENCES

[1] Costas Busch, Srikanth Surapaneni and Srikanta Tirthpura "Analysis of Link Reversal Routing Algorithms for Mobile Ad Hoc Networks ", *SPAA'03,* June 7–9, 2003 San Diego, California, USA.

[2] R. Rajaraman. ,"Topology control and routing in ad hoc networks", A survey. SIGACT News, June 2002

[3**]** Schindelhauer, "*Mobility* in Wireless Networks".In 2010**.**

[4] Charles Wang , "*Wireless Link Reversal Algorithm Simulation"* Information Trust InstituteUniversity of Illinois Urbana-ChampaignIL, United States ..

[5] M. S. Corson and A. Ephremides. "A distributedrouting algorithm for mobile wireless networks".ACM/BaltzerWireless Networks Journal, 1(1):61{82,February 1995..

[6] E. M. Gafni and D. P. Bertsekas. ,"*Distributedalgorithms for generating loop-free routes in networkswith frequently changing topology",* IEEE Tran. Oncommun. COM-29:11{18, 1981.

[7] N. Malpani, J. L. Welch, and N. Vaidya. ,"*LeaderElection Algorithms for Mobile Ad Hoc Networks",*. InProceedings of the 4th internationalworkshop ondiscrete algorithms and methods for mobile computing and communication, Aug. 11, 2000.

[8] Miia Vainio," Link Reversal Routing" P.O.Box 407 pdf file.

[9] Marc Schiely, "*Survey of Mobile Ad-hoc Routing Algorithms*", Dept. of Computer ScienceSwiss Federal Institute of Technology (ETH) ZurichWinter 2003 / 2004.[10**]** C. E. Perkins,"*Ad Hoc Networking*", Addison Wesley,ISBN 0-201-30976-9 ,2000.

