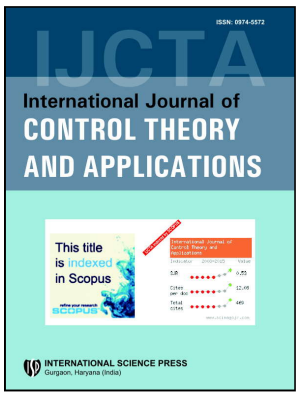

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 13 • 2017

Path Building Algorithm to improve Quality of Service in MANET'S

Kaushik N.S. Iyer[1] and Sowmya K.S.[2]

[1] Student, Dept. of Information Science and Engineering. BMS College of Engineering Bangalore - 560 019, India, Email: kaushiiyer@gmail.com

[2] Assistant Professor, Dept. of Information Science and Engineering. BMS College of Engineering Bangalore - 560 019, India, Email: sowmyaks.ise@bmsce.ac.in

Abstract: In an Ad hoc network many a times extra data and resources are spent in route request and the response for each and every transmission which forms the core principles in reactive style routing. This paper presents an algorithm known as the "Path Building Algorithm(PBA)" which routes packets and data over multiple hops by building paths from the source node to the destination node and hybrid in nature. The efficiency of the path is sent back to the source which makes for a transmission cycle. Thus all forms of cycles are formed and cached at every node in a structure known as the path table, which stores not only the most efficient paths from the source to a destination but also multiple alternatives. This algorithm works on calculating and storing all forms of request and responses based on a network efficiency coefficient K based on transmission time within a cluster which is nothing but a logical collection of nodes with a leader chosen either based on computational power or by some other cost as the parameter, thus the algorithm performs almost seamlessly with mobile nodes changing its location rapidly to a certain factor without having to request a route at every instant or store multiple paths to the same destination that does not improve the efficiency of the network at a larger scale. Also this algorithm reduces on the amount of path information to be stored and hence reduces the packet size and since the responses are already known the response packet may be used to refine the instant when the route request has to be fired. Hence providing better Quality of Service by switching to an alternate route even without any computation required.

Keywords: MANET, QoS, PBA.

1. INTRODUCTION

Increasing number of mobile devices with various features and capabilities enables us to create scalable local networks for communication. These networks have mostly peer-to-peer data communication. These networks pose many issues which involve routing protocols, maintaining data about the nodes, creating and maintaining routing tables, reconstruction of the network when a node drops etc.

A mobile ad hoc network (MANET) is a continuously self-configuring, infrastructure-less network of mobile devices connected wirelessly[1].The devices in a MANET are not coupled and are free to move in any

direction or drop out of the network at any time. Each device in the network forwards traffic that may not be related to it, hence behaves as a router. This gives rise to many routing protocols which focus on different aspects of a network like performance, stability etc. MANETs are mostly peer-to-peer networks which are self forming and self healing. The types of MANETs include Vehicular ad-hoc networks, Smart Phone ad-hoc networks, Internet based MANETs etc[2].

The routing protocols used in ad-hoc networks are Table Driven routing, On Demand routing, Hybrid routing and Hierarchical routing. The Table Driven algorithms work by maintaining lists of destinations in tables distributed across the network. The On Demand based algorithms work by flooding the network with Route Requests. The hybrid Routing algorithms work by combining certain techniques of both the above types of the algorithms. In Hierarchical routing algorithms, the routing algorithm is chosen based on an hierarchic level to which the nodes belong.

The challenges faced in MANETs are Limited Bandwidth, Dynamic Topology, Routing Overhead, Packet loss and Transmission Errors, Security etc.[3] The Packet loss and Transmission errors are handled by the routing algorithms. Some of the most commonly used reactive routing protocols are Ad-Hoc On Demand Distance Vectoring (AODV), Dynamic Source Routing (DSR), Associativity Based Routing (ABR), Signal Stability Based Adaptive Routing Protocol (SSA) and Temporarily Ordered Routing Algorithm (TORA).

A Cluster is a logical collection of nodes which may be geographically near where distance is taken as the cost or by some other parameter taken as the cost.A cluster leader is elected based on this cost and is considered as an intermediate point of contact to all nodes within that cluster.Whenever a node drops out before a reply from a node is received by the sender node, the whole network is flooded with route requests for finding a path from the replying node to the destination. This creates a lot of overhead on the network and also results in increased bandwidth usage, delay in communications etc. This flooding can be avoided in certain cases by building and storing paths between all the nodes in the network.

2. LITERATURE SURVEY

Ad hoc means ‘to this’ in latin and is a very popular way of routing data packets. it is especially effective in a mobile network. In an ad hoc network a particular variable is chosen as the cost and the cost is used to determine how the flow of the packets is done between nodes. Hence an ad hoc network can be used to find the path of least cost between nodes. Once this path is set, the source and the destination can communicate. Hence it is often impromptu because of the lack of an infrastructure between their nodes. It is ad hoc because it has no fixed time in the network and can go off at any time, hence it has no protocol. Based on the way and the timing of the flooding of the packets there exists different types of routing protocols. Such as proactive, reactive and hybrid routing which comprise of both of the above[4].

Considering a reactive system, if a particular source wants to communicate with a destination it broadcasts or floods a message to all the nodes nearby its presence and then they further flood it until the destination is reached. This is done by checking if the packets received is meant for the current node or not. Once the checking is done it further floods to its nearby nodes with the current node as the source and the process further continues. Once the required destination is reached a confirmation is sent to the immediate source and then the immediate source checks the packet I.D. and then this source sends a confirmation to its source so on until the actual source is reached. Once this is completed, it is said that a route has been set for the required communication. The process of flooding the packets is known as a route request and the confirmation sent is known as a route reply.

Based on the explanation given above on ad hoc there exists two steps in the setup of a ad hoc route.

1. Route Discovery.
2. Route Maintenance.

AODV it is a novel algorithm to communicate between nodes in a on-demand fashion and provides a loop free route even during repair [5]. Modern computers have huge battery and hence the mobility and portability is also increasing. Hence we require communication algorithms that can handle such movement of routers or nodes and provide an alternative route for less cost. The Destination-Sequenced Distance algorithm or famously known as the DSDV algorithm provides a mechanism where a set of cooperating nodes can form an ad hoc network. But there are some disadvantages, one of the main being that the overhead costs grow at a rate of

$O(n^2)$. According to this algorithm all the nodes that do not take part in the communication need not have the details of the communication path.

It is a reactive kind of routing protocol and one particular node is aware of its neighbours based on a broadcast messages known as "hello" messages whose time to live or TTL = 1, and hence any changes that occurs to the current network, like if a node goes down, it is made known only to those which get impacted from them. This keeps the process simple but has high latency time when a lot of activity is happening in between the nodes. Hence it is possible that a valid route may be expired but we may be unable to determine it because of the high mobility of these nodes[6]. Also AODV runs on the assumption that all the nodes will cooperate.If not the entire communication fails.Hence we need to make sure that we remember the route which is the most efficient and if taken must be able to provide with an alternate route without having to redo the whole process again. Hence we need structures and modified routing tables that can save such data.

DSR(Dynamic Source Routing) is another algorithm where the complete path of the nodes to be travelled is determined dynamically or is specified specifically in the packet header by identifying the next hop as the node to be visited next on its way to the destination node and then caches the route and any further route to be determined is found based on the cache and the route discovery algorithm[7]. Unlike the AODV algorithm the DSR algorithm can have asymmetric edges meaning that a communication path between node S to D need not imply that a communication must exist between D to S, and also there might exist more than one route between two source nodes. If a particular route is being accessed more than once then the hops would remember the path and hence the forwarding would happen without any overheads. But this cache cannot be maintained forever hence each path has an expiration. Usually when the route required is found in the route cache it saves this route along with an expiration time associated with it and when it crosses this, it is deleted like cookies. Hence there exists a need to choose the expiration time wisely else the transmission overhead will prove too costly.

Also whenever a path cannot be found, the packet can either be stored in its network layer until a path is found or the path recognition algorithm could be run once more. Another feature is that a host using a particular path needs to make sure that a particular path can be used, like if a node in a particular path goes off the range, then the path is not viable anymore and hence a route maintenance algorithm needs to be run, which if fails has to be followed by a route discovery process once again. Hence it is seen that this has a advantage over AODV that it maintains route cached but it may also lead to multiple routes being maintained between a source and a destination and the entire path has to be sent along the packet which means that the size of the packet is not fixed and for exponential nodes, it proves too costly to be sending such large data per packet or even caching them. Hence a set of routing tables which are symmetrical in the sense that each others presence effect every other node in the system and provide the same transmission rates with far less cache and a more structured way to show remember the past without having to worry about an expiration are needed.

3. ALGORITHM AND DESCRIPTION

The proposed algorithm maintains all forms of request and reply cached and maintain a structure that has details of the transmission to occur even before the nodes actually start communicating, this makes sure

that we do not have to restart the request in a ad hoc process. Flooding is the most common and widely used request process and it has a common enemy - network clogging, but this comes with a advantage that it need not maintain any form of data in its system and hence it is a pure reactive style routing. But the proposed algorithm intends to present a hybrid style of routing with a initial setup as a set of nodes which belongs to a cluster and has a cluster leader and all the cluster leaders communicate with each other in a Ad hoc fashion.

But to increase transmission inside a cluster and provide better Quality of Service the proposed algorithm executes the following steps simultaneously at every node:

- Ad hoc style flooding of packets and the reply from the nearby nodes is the cost to transfer packets to that node(here cost is mainly in form of time taken).
- The cost obtained from the flooding is placed in a cost matrix in the cluster leader.
- Using the cost matrix in the cluster leader the path building algorithm as explained in section III B is executed.
- The path so calculated will have a network efficiency associated with it in terms of the cost which is in turn dependent on the time of transmission.
- Based on this network efficiency a path table is generated and that acts like the required cache.
- Now the transmission can start and without any further computations, an immediate reply can be sent for the whole transmission with a simple reply packet without having to wait for the packet to make the complete transmission.

Consider the network shown in Fig. 1. The algorithm shall be implemented for the following setup with node 1 as the cluster leader:

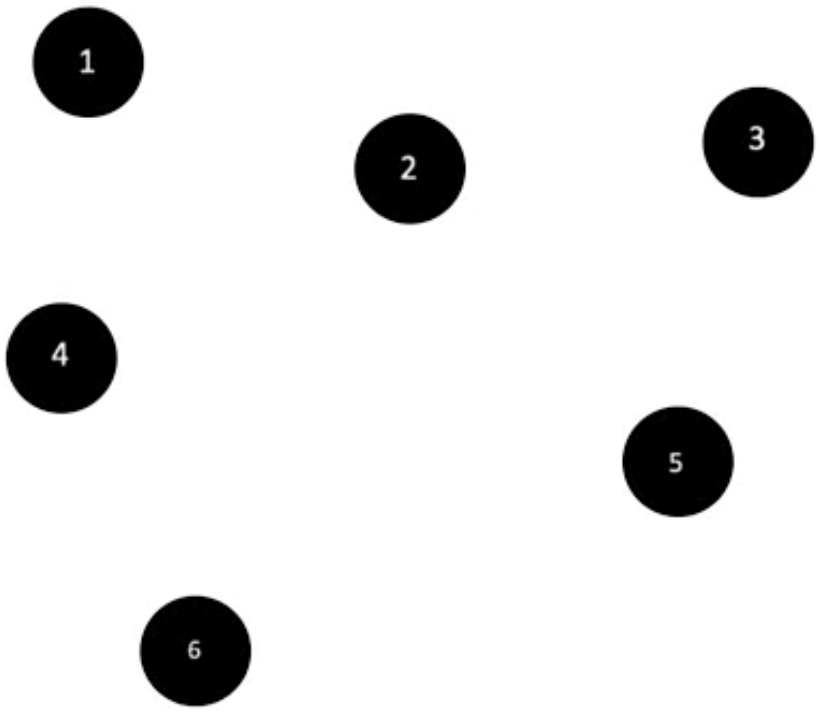

Figure 1: Nodes in the network

3.1. Cost Matrix

The cost matrix is the output of the Ad hoc style flooding where every node sends its RREQ packets to its nearby surroundings and as every other node receives this packet it replies with the amount of time left of the Total Travel Time (TTT) and the ratio of the remaining time to the TTT. This is the corresponding entry in the matrix, with the source nodes as the rows and the destination nodes as the column in the matrix. If a node is unreachable within TTT this means that its entry will be 0. The time taken to transmit to the same node is also considered 0 as shown in Fig.2. The cluster leader is also a node in the cost matrix.

$$\begin{bmatrix} 0.0 & 0.5 & 0.2 & 0.1 & 0.3 & 0.8 \\ 0.5 & 0.0 & 0.2 & 0.1 & 0.3 & 0.8 \\ 0.5 & 0.2 & 0.0 & 0.1 & 0.3 & 0.8 \\ 0.5 & 0.2 & 0.1 & 0.0 & 0.3 & 0.8 \\ 0.5 & 0.2 & 0.1 & 0.3 & 0.0 & 0.8 \\ 0.5 & 0.2 & 0.1 & 0.3 & 0.8 & 0.0 \end{bmatrix}$$

Figure 2: Cost Matrix

3.2. Path Building Algorithm

Consider situation shown in Fig. 3 where a packet travels from one source node to a destination node via an intermediate node. The cost of this is represented as:

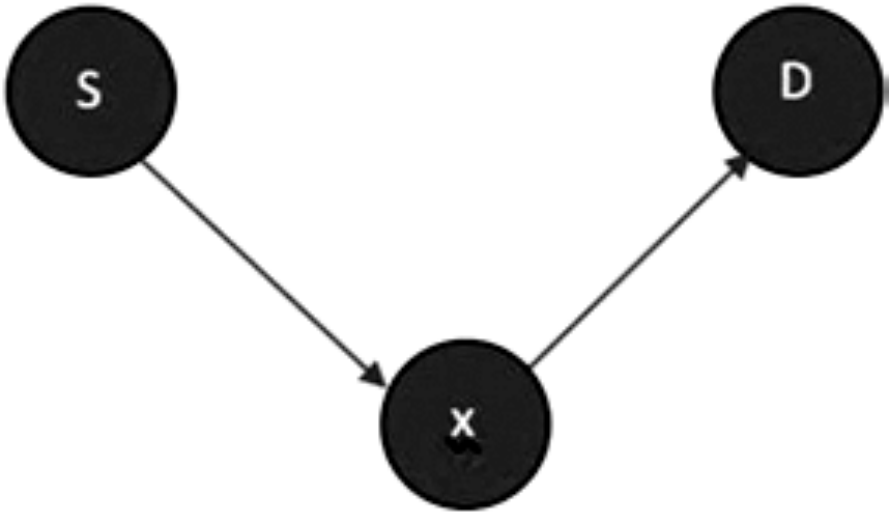

Figure 3: A path of length 2

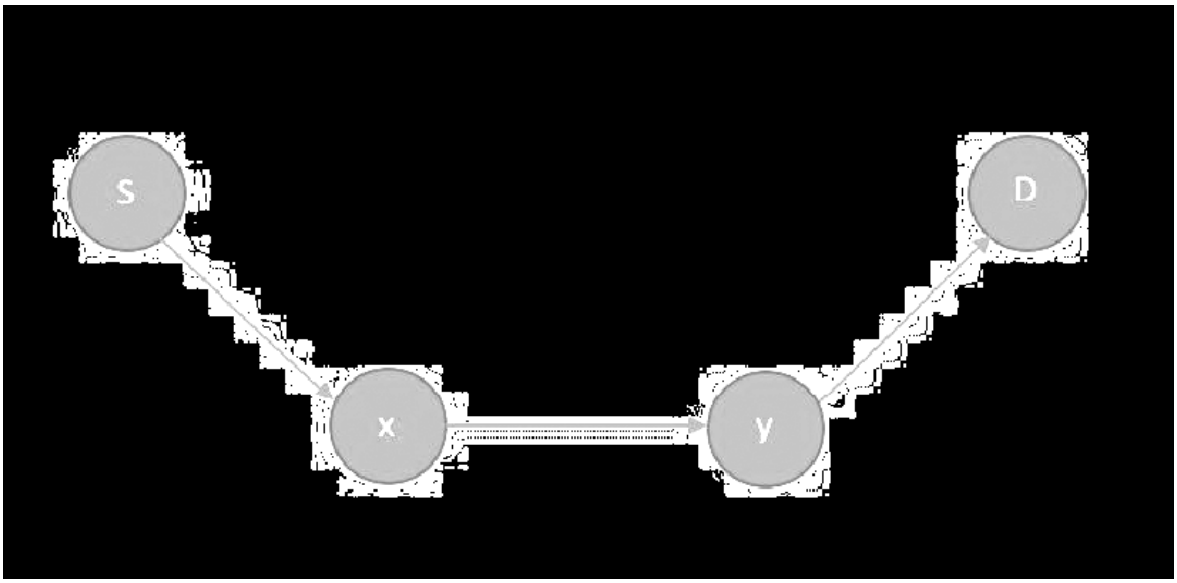

Figure 4: A path of length 3

$$cost[S][D] = cost[S][x] * cost[x][D]$$

Similarly for travelling through two intermediate nodes the cost[S][D] shown in Fig.4 is given as:

$$cost[S][D] = cost[S][x] * cost[x][y] * cost[y][D]$$

A recurring trend is seen as the paths are built for various path lengths. The number of cost terms would increase exponentially for increasing path lengths and hence the complexity would be $P*V^2$ if a path is built for a network of nodes V and path length P. Hence this needs to be reduced but also similar results must be obtained.The path building algorithm intends to store all path whose lengths are power of 2 in a Set which has the following tuple representation:

$$Psrc=\{([v1….., vdst],K)\}$$

where:

- psrc, is the path with S as the source
- vdst, v1 and so on are the nodes that belong to the path from node S to node D
- K represents the network efficiency of the path mentioned.

The set initially comprises of tuples with all the paths of length one from vsrc to all the other nodes in the cluster from the cost table. Hence for the cost table shown above, the initial set will be represented as :

$$Psrc=1 = \{([2], 0.5), ([3], 0.2), ([4], 0.1), ([5], 0.3), ([6], 0.8)\}$$

To build all paths of length two, the following tuple designation can be used, in which node 2 is the destination:

$$Psrc = 1 = \{([x, 2], k)\}$$

All the paths which have x as the destination node and the tuples which have x as the source and any vertex other than x and its source node as the destination have to be merged.Once this process of merging two sets are over a new set with all the path of length two is obtained. Hence a path of length two is built from paths of length one. For the given example , it can be represented by:

$$Psrc = 1 = \{([3, 2], 0.04), ([4, 2], 0.02), ([5, 2], 0.06), ([6, 2], 0.16)\}$$

Similarly to build path of length three, sets of length two should be merged with sets of length one shown in Fig. 4.

The tuple representation is:

$$Psrc = 1 = \{([x, y, 2], k)\}$$

where y is a node in the network and does not belong to {x, 2}

The result obtained will be a set with varying path sizes and network efficiencies. From this the path table is filled with the most efficient route and an alternate one so that this entire process is repeated seldom. The process of merging two sets has a complexity of $O(n^2)$ Only sets of path lengths having powers of 2 are stored to build a path of different lengths. A simple illustration of path building is shown in Fig. 5:

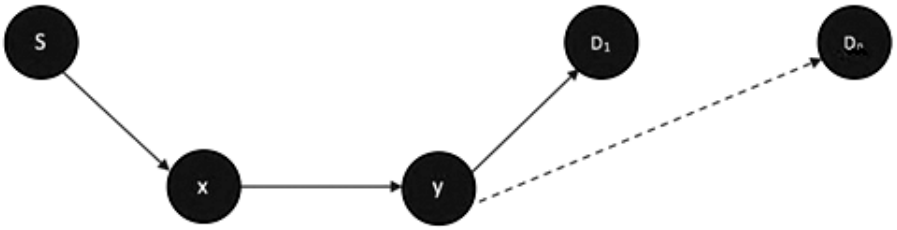

Figure 5: An Illustration of path building.

3.3. Path Table

As explained earlier the path table is stored in the cluster leader where the rows form the sources and the columns form the destinations. by running a simple code we get the following results shown in Fig.6 where we build paths of length 2. Here N/A is to show that loops are not to be counted

- Source node is 1
- Rows are destination nodes
- Columns are intermediate nodes

/	2	3	4	5	6
1	N/A	N/A	N/A	N/A	N/A
2	N/A	0.04	0.02	0.06	0.16
3	0.1	N/A	0.01	0.03	0.08
4	0.05	0.02	0.03	0.09	0.24
5	0.15	0.06	N/A	N/A	0.64
6	0.40	0.16	0.08	0.24	N/A

Figure 6: Results obtained for various sources and intermediate nodes for path length 2.

From the data obtained above the structure of the table is as showing Fig.7 :

/	2		3		4		5		6	
1	0.5	2	0.2	3	0.1	4	0.3	5	0.8	6
	-	-	-	-	-	-	-	-	-	-
2	0.16	6	0.1	2	0.24	6	0.64	6	0.4	2
	0.06	5	0.08	6	0.09	5	0.15	2	0.24	5
3										

Figure 7: Path Table for the given configuration

As we see that not only the most efficient path is stored but also it is accompanied with an alternate path for each edge length. Hence it is made sure that the path building algorithm is not executed again unless it is absolutely necessary. For a network of V vertices, there can be maximum path of length V-1 but it is seen that most communications do not happen without any losses. Hence an attempt to transfer data through a path of length closer to V-1 results in losses being multiplied and the efficiency of the network further going down. Hence it is advisable not to build paths greater than V/2 or V/3. The cluster leader is also a column in the path table, the path table should also maintain a path table hence it should be more powerful.

For the given path table, it is open to interpretations as to when the path table must be reconstructed by restarting the path building algorithm such as normalisation of the table i.e. if it contains too many null values or if the advised network efficiency by the table is not satisfactory.

3.4. Indexing and index array

The array accompanies the path table for every source to a particular destination. The table may have multiple paths of varying lengths. The best out of this needs to be selected. The parameter used to select this is the network efficiency K, which is present in the path table. So the index array will simply consist of most efficient path length to be taken for array index i as the destination. The index array for the given example is shown below:

$$[\,1, 1, 2, 2, 1]$$

If the most preferred path length is two, but if the transmission returns an RRER packet then the alternate path present within the same cell is switched with the most preferred path and the alternate path is set to null. Consider the situation where both the path are invalid. In this case, the array needs to be reindexed by choosing the next best path length with comparable network efficiency. The array below shows a situation where node 6 becomes unresponsive:

$$[\,1, 1, 2, 1, 1]$$

It is open to interpretation as to whether the array must be reindexed or the path building algorithm needs to be rerun to restructure the table.

4. PATH REQUEST AND RESPONSE

For an ad hoc network the initial flooding consists of RREQ packets accompanied by the RREP response. But once the path table is setup, all transmissions between the cluster nodes occur using data packets of the structure shown in Fig. 8:

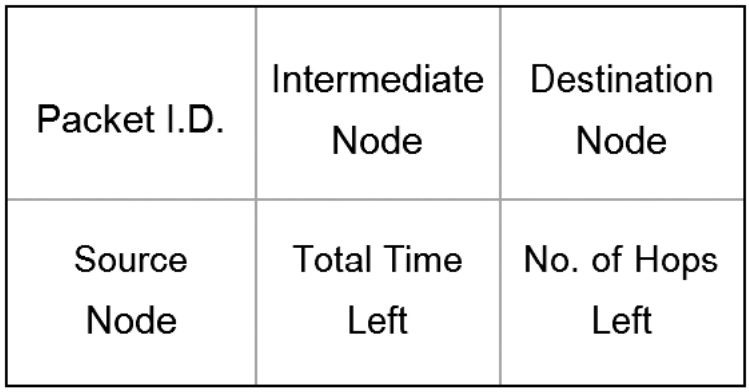

Figure 8: Structure of request packet

For every request sent, the packet reaches form one hop to another and at every hop the response packet is sent for the whole transmission. This is possible because the paths for all forms of requests and responses have been stored in the path table. This results in a transmission where the destination node need not send a response to the source node which would imply that the source node is waiting and is unable to clear its routing table. Hence creating a response mechanism which is heuristic in the sense that the response is obtained before the transmission is completed is needed. This results in a mechanism where the number of nodes waiting for response reduces as the hop count reduces.The response sent back to the source node consists of the total time taken for the transmission which can act like a factor to determine whether the rerun of the path building algorithm is required. The structure of the response packet is as shown in Fig. 9.

Figure 9: Structure of response packet

5. CONCLUSION

This paper has presented a protocol which is a hybrid system. It consists of a set of nodes within a cluster with a cluster leader. The protocol highlights on saving all the request and response times in terms of the total travel time(TTT) and the difference between the TTT and request time known as the network coefficient K in a table known as the path table which not only stores the most efficient path but also an alternative path whose network efficiencies are comparable. They are further indexed so that the routing between intermediate nodes occurs without any lag.

To summarise, the path table is built with an Ad hoc style flooding and is followed by the cost matrix, which is stored in the cluster leader. The path building algorithm is then implemented and it results in a set of optimum paths with varying path lengths. Then the two most optimal ones are chosen and are stored in the path

table. They are further indexed into an index array. The index array is used to get the most optimal route and then a request is prepared for the intermediate nodes. When a particular intermediate node is reached, the total number of hop count is reduced and then the required destination is checked using the remaining hop count. A response is sent back to the source node for the whole transmission immediately because the coefficients are present in the path table already.

As explained earlier the route cache is in the form of a path table and is more structured and also it is made sure varying number of paths to the same destination node are not stored. Hence for the initial overhead cost of flooding the network and building the cost matrix has a complexity of O(n). The path building algorithm which stores only paths of power 2 saves the number of iterations required to build paths for larger numbers of n and has a complexity of $O(n^2 * \log(m))$. But once this is finished the transmission can start with a complexity of O(1).

Another feature is that addition of a new node does not mean that the existing nodes have to rerun the request. It has to be executed only by the new one.

The given structure is explained only within a particular cluster. If it were to be expanded to cluster level implementation then the packet size would increase. This algorithm has not been tested for very high number of nodes within the same cluster. Also if there is a very volatile movement where the nodes are rapidly disappearing then the overhead will increase in a quadratic fashion. If the newly entering node has significantly higher network coefficient compared to the existing ones it might change a lot of table entries.

ACKNOWLEDGMENT

We would like to thank our college and university for giving us an opportunity to execute our idea. We would like to thank Mahesh H P for assistance with the introduction and also extend our gratitude to him and Bhargav Mysore for sharing their pearls of wisdom with us.

REFERENCES

[1] Chai-Keong Toh (2002). "Ad Hoc Mobile Wireless Networks: Protocols and Systems 1st Edition".Prentice Hall PTR. Retrieved 2016-04-20

[2] Zakir Hussain, Dr. Balakrishna R,"A survey on MANETS - Types, Characteristics, Applications and Protocols used".

[3] Gurpinder Singh, Asst. Prof. Jaswinder Singh,"MANET: Issues and Behavior Analysis of Routing Protocols",International Journal of Advanced Research in Computer Science and Software Engineering Volume 2 Issue 4 April 2012.

[4] Preetha K G, A Unnikrishnan and K Poulose Jacob, "a probabilistic approach to reduce the route establishment overhead in AODV algorithm for MANET",International Journal of Distributed and Parallel Systems (IJDPS) Vol. 3, No. 2, March 2012.

[5] Charles E Perkins and Elizabeth M Royer, "Ad hoc On Demand Distance Vector Routing".

[6] Prashanth Kumar Maurya, Gaurav Sharma, Vaishali Sahu and Mahendra Shrivastava," n Overview of AODV Routing Protocol ",IJMER Vol. 2 Issue 3, May-June 2012 ISSN: 2249-6645 - A.

[7] David B. Johnson and David A. Maltz,"Dynamic Source Routing in Ad Hoc Wireless Networks".