A Theory in Optimization of Ad-hoc Routing Algorithms

M. Kargar, F.Fartash, T. Saderi, M. Ebrahimi Dishabi

***Abstract*—** In this paper optimization of routing in ad-hoc networks is surveyed and a new method for reducing the complexity of routing algorithms is suggested. Using binary matrices for each node in the network and updating it once the routing is done, helps nodes to stop repeating the routing protocols in each data transfer. The algorithm suggested can reduce the complexity of routing to the least amount possible.

***Keywords*—** Ad-hoc Networks, Algorithm, Protocol, Routing Train**.**

I. INTRODUCTION

AD-HOC networks are getting popular for their ease and speed in deployment, decreased dependence on infrastructure, being the only possible solution to interconnect a group of nodes and many commercial products available today.

A. Terminology

Ad hoc network is a collection of wireless nodes that can dynamically be set up anywhere and anytime without using any pre-existing network infrastructure [1, 2, 3, 4, 5]. The major characteristics of ad hoc networks are dynamic topologies, being bandwidth-constraint, energy-constrained operation, and limited physical security [6, 7].

B. Prevalent Protocols

Some of the most common routing protocols are named and addressed bellow:

Destination Sequenced Distance Vector (DSDV) is a table-driven/ proactive protocol in which routing is done by using the Bellman-Ford Algorithm [10] for each node by the information which exists at their tables [1].

Clusterhead-Gateway Switch Routing (CGSR) is a table-driven/proactive protocol in which cluster-heads are selected using an election. The route is found through cluster-heads which is usually done by DSDV Protocol [1].

Optimized Link State Routing (OLSR) is a table-driven/proactive protocol in which routing is done by using the Dijkstra Algorithm [10] for each node by the information which exists at their tables [11, 12].

Wireless Routing Protocol (WRP) is a table-driven/proactive protocol in which each node sends a hello message to its neighbors and considers them as its successors and does this job till reaches the destination [13].

Zone Routing Protocol (ZRP) is a hybrid protocol in which each node denotes route request from the nodes in its zone. The node which has the destination node in its zone denotes route reply [8].

Dynamic Source Routing Protocol (DSR) is an on-demand-driven/reactive protocol in which routing is done by the packet propagation through the network [1, 2, 4, 12, 14, 15, 16, 17]. For further information about packet broadcasting refer to [18].

Ad-hoc On-demand Distance Vector Routing (AODV) is an on-demand-driven/reactive protocol. It is just like DSR. The difference between this protocol and DSR is in hello messages which make AODV reply faster when there is no route to the destination [4, 12, 14, 16, 17, 18].

Temporally-Ordered Routing Algorithm (TORA) is an on-demand-driven/reactive protocol. It uses three packets: query, update and clear [14, 16]. So the whole graph is updated for each node after the questioning has finished.

Landmark Routing (LANMAR) is a cluster-based/hierarchical protocol. It uses some nodes as land-marks and finds the destination by their guidance [20].

Core-Extraction Distributed Ad-hoc Routing (CEDAR) is a cluster-based / hierarchical protocol. It sets a dominator for each cluster and the nodes do questioning from the dominators. After finding the routes they give the nodes the routes they should pass [21].

C. The Problem

Routing protocols in ad hoc networks are divided into four main groups: 1-Table-Driven / Proactive, 2-Hybrid, 3-On-Demand-driven /Reactive, 4-Clusterbased / Hierarchical [8]. Each group contains so many routing algorithms with special advantages and disadvantages in comparison with others. Some of the mostly used ones have been studied and optimized, resulting in certain methods, theories, and algorithms. The problem with most of these optimizing

Mas'ood Kargar is with the Department of Computer Engineering, Islamic Azad University of Tabriz Branch, Tabriz, Iran. He is also collaborating in Islamic Azad University of Tabriz Branch Computer Research Laboratories. IAUT-CRL (phone: +98-9141035597; email: kargar@iaut.ac.ir).

Farzaneh Fartash is with the Department of Information Technology Engineering, International University of Chabahar (phone: +98-9141002139; e-mail: fartash@iuc.ac.ir).

Taha Saderi is with the Department of Computer Engineering, Islamic Azad University of Tabriz Branch (phone: +98-9143165992; e-mail: saderi@iaut.ac.ir).

Mohammad Ebrahimi Dishabi is with the Department of Computer Engineering, Islamic Azad University of Miyaneh Branch, Miyaneh, Iran. (email: mrebrahimy@m-iut.ac.ir).

algorithms, however, is that they have been produced for special protocol and can not be used on the others.

D. Solution

Here we suggest a new way for optimizing routing protocols by training the network. In this method, after a period of time, the complexity of routing protocol reduces to the most optimized range. Obviously running the routing protocols for n-times is not a good idea. In all protocols we always want to find the optimized way to reduce the time spent. Therefore if we repeat optimizing for n-times then it is an overhead itself. As a solution a new method for reducing the times that the routing protocols are repeated, is suggested. Although the presented algorithm is no more than a theory and hasn't been practiced, but we think it will help a lot if it becomes accepted, simulated and finally examined.

E. The Claim

We claim that such a solution does exist. By using binary matrices for each node in the network and finding a route through running a routing protocol and at last updating the matrices considering the route given, fulfils the need for repeating the routing protocols in each data transfer. For further information about rerouting in the protocols refer to [9].

F. Objective

The objectives we want to achieve by suggesting a new method is complexity reduction in routing algorithms. By this theory, we reach to a point that the nodes do not need to ask for the route and they learn the way that they should send their data. So only the nodes on the route are visited and the data is passed through them.

G. Paper outline

This paper is organized in four sections. In Section 1 we started the problem, an idea solution to the problem and our claim regarding the solution. In Section 2 we speak about the works done for routing optimization and address some of them. In section 3 the Training Algorithm is given and its benefits are described. In section 4 the reason of suggesting new method in the ad hoc networks is concluded.

II. Previous Work

Although there are some works and surveys done for the optimization of routing or increase in security of ad-hoc networks but still there are shortages to be fulfilled. Some of the papers describing the jobs done for routing optimization are mentioned and introduced below as the previous work for our suggestion in this paper.

In [22] a model of life time route optimization in wireless ad-hoc networks has been surveyed. The route lifetime value is one of the most important parameters for the design of an on-demand ad-hoc routing protocol. This parameter determines the duration of an active path/route in the routing table to transmit the packets reliably. This is to ensure that the routing table does not attempt to discover a new route and/or delete an existing active route within its lifetime. So, too long route lifetime may lead to retardation in updating the routing table even though some paths are broken [22].

In the referred paper, adaptive route lifetime determination through a fuzzy logic system is proposed. Fuzzy logic is chosen due to the uncertainty associated with node mobility estimation and drawbacks of mathematical models. Definition of fuzzy sets (membership functions) and a set of rules (rule-base) have been proposed to design the new method, called fuzzy ART. This new method is evaluated with the AODV routing protocol, we believe it can be generalized for other ad-hoc routing protocols, as well [22].

In [23] a Dynamic Source Routing Protocol using Self Healing and Optimizing Routing Technique based on fuzzy logic concepts is presented. The paths generated by conventional dynamical source routing protocol deviate far from the optimal paths because of the lack of knowledge about the global topology and the mobility of nodes. Routing optimality affects the network performance, especially when the load is high. Longer route consumes more bandwidth, power and is more prone to disconnections. Self Healing and Optimizing Routing Technique (SHORT) is a technique that monitors the route and tries to shorten it, if a short-cut is available. The proposed fuzzy logic method is evaluated and compared with conventional method using GloMoSim [23].

In [24] the work concentrates particularly on securing routing protocols, which are still immature and under rapid development. Because of high dynamics and other limits shown before, the design of ad-hoc routing protocols is more complicated and usually a nice piece of trade-off among multiple factors, which include improving routing optimum, minimizing traffic volume and restricting power use. Though a lot of new protocols have been proposed and implemented, we understand that security issues are rarely concerned or even so, hardly practical [24].

In [25] time-slots are assigned for each node in the network to access the control channels so that it is guaranteed that each node can broadcast the control packet to any one-hop neighbor in one scheduling cycle. The objective is to minimize the total number of different time-slots in the scheduling cycle. It leads to a determined access scheduling on the control channels. Each node is assigned with one / several chance(s) (time-slot in a TDMA system) to access the control channel, and the broadcasting on the assigned time-slots is guaranteed to be received correctly by its neighbor(s). The access delay on the control channels is upper-bounded by the length of the scheduling cycle. Note that a single time-slot can be reused by two nodes if they do not interfere with each other. The objective of the access scheduling problem is to minimize the length of the scheduling cycle [25].

In [26] three major optimization schemes for the well-known AODV routing protocol are described in order to get some of the proactive protocols features in it. The described schemes are: Reverse path setup, Forward path setup and Route Scattering. The targeted characteristics are: traffic independent control and shortest path routes.

In [27] routing optimality using different metrics such as path length, energy consumption along the path, and energy aware load balancing among the nodes are defined and a framework of Self-Healing and Optimizing Routing Techniques (SHORT) for mobile ad hoc networks is proposed. While using SHORT, all the neighboring nodes monitor the route and try to optimize it if and when a better local sub-path is available. Thus SHORT enhances performance in terms of bandwidth and latency without incurring any significant additional cost. In addition, SHORT can be also used to determine paths that result in low energy consumption or optimize the residual battery power.

III. Training Algorithm

A. Algorithm Representation

As described above, all routing algorithms mostly optimize the route found and pay less attention to the reduction of complexity in routing. The protocol used should repeat the routing every time that data transfer is required. Of course some optimizations are done such in DSR [28] but the main problem of these optimization methods is that they are just specific for the protocol given. In the method used for optimizing the DSR protocol, a cache is used for saving the passed route and routing may be repeated for some times.

The algorithm that we suggest here is protocol-free. It can be adjusted in all routing protocols and we can say it is high level. It can also use different routing protocols in one network. In fact we are training the network to learn from routing and after doing the routing protocol for a few times the nodes themselves know that from which route they should transfer data. This is the time when the complexity of routing for transferring data, reduces to "L" which is the exact route distance from source to destination. In other words instead of running O(V*E) times we run O(L) times which equals with the number of the edges that should be passed. This is the most optimized complexity a programmer or manufacturer may seek for.

In addition, in this algorithm, not only we have thought about the time cost, but also we have used a binary matrix in training, that reduces the memory cost to minimum and helps the training period of time to be spent so fast.

Before writing the algorithm, the suppositions at the network are mentioned:

1- The network is supposed to be a graph with nodes and edges.

2- Nodes are represented by numbers.

3- The edges from one node to its neighboring nodes are numbered consequently. By giving a number for each node, the ordering of edges becomes easier. Of course the number of nodes related to one node needn't be serial.

4- Each node has an array for training and an array for controlling bandwidth and traffic. The bandwidth array is not used in training and it is just for controlling which increases the transferring quality. Using table for saving the information by most of the protocols makes the algorithm easy to setup.

5- There is one adjacent matrix for the whole graph.

6- "G" is the graph, "w" is the weight, "s" is the source, "d" is the destination, "R" is the matrix for routing and "Ro" is the route array.

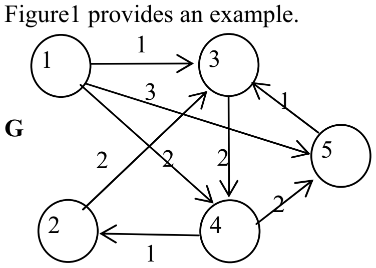
Here is the code of Training Algorithm:

```
Training_Algorithm (G,w,s,d)
If ! R[d]
 Ro = get_route_from_a_routing_algorithm(G,w,s,d)
send_update(G,Ro,0,d)
-----------------------------------
Send_data(data,s,d)
Training_ Algorithm (G,w,s,d)
Transfer_data(data,s,d)
-----------------------------------
Transfer_data(data,s,d)
If s <> d
J=index_vertex(s, lg R[d])
Transfer(data,j,d)
-----------------------------------
Send_update(G,Ro,start,d)
If Ro[start] <> d
K=index_Edge(Ro[start], Ro[start+1])
R[d]=2^k
Send_update(G,Ro,start+1,d)
-----------------------------------
Index_Edge(i,j) As integer
K=0
Flag = false
While (flag)
Do
If A[i][k]
Counter=counter+1
K=k+1
If (counter=j)
Flag=false
Return counter
-----------------------------------
Index_vertex(i,k) As integer
K=0
Flag = false
While (flag)
Do
If A[i][k]
Counter=counter+1
L=L+1
If (counter=k)
Flag=false
Return L
```

As it is shown, before data transfer, first the algorithm checks that if the graph is trained or not. If it is trained, then data is sent, else it gets route as an array from a routing protocol. After this, it trains the nodes by sending update through the route given. In fact updating does the job of training.

Each node has a binary matrix with one dimension. The rows of that matrix or the indexes of the array show the numeric label of each node. The bits given in rows show the edge that the node can send its data through it. By using logarithm and powering in the algorithm, the binary matrix can be filled so easily. In addition, using the binary matrix takes less memory cost and the access takes O(1) times and this causes the training and routing operations run simply.

Figure1 provides an example.



Node 1:

$$\begin{matrix}1\\2\\3\\4\\5\end{matrix}\begin{bmatrix}000\\001\\000\\001\\010\end{bmatrix}$$

1	2	3	4	5
0	2	1	2	4

Node 2:

$$\begin{matrix}1\\2\\3\\4\\5\end{matrix}\begin{bmatrix}000\\000\\000\\000\\000\end{bmatrix}$$

1	2	3	4	5
0	0	1	1	1

Node 3:

$$\begin{matrix}1\\2\\3\\4\\5\end{matrix}\begin{bmatrix}000\\000\\000\\000\\000\end{bmatrix}$$

1	2	3	4	5
0	1	0	1	1

Node 4:

$$\begin{matrix}1\\2\\3\\4\\5\end{matrix}\begin{bmatrix}000\\000\\000\\000\\001\end{bmatrix}$$

1	2	3	4	5
0	1	1	0	2

Node 5:

$$\begin{matrix}1\\2\\3\\4\\5\end{matrix}\begin{bmatrix}000\\000\\000\\000\\000\end{bmatrix}$$

1	2	3	4	5
0	1	1	1	0

Fig.1 A sample for the matrices designed for the arrays of the nodes in a graph G.

While sending update, the matrix in each node on the route is updated. For example if the route from 1 to 2 first passes from 4, and the edge from 1 to 4 is the second edge of 1, then in the matrix at the fifth row the number $[9]_2$ (=$[2]_{10}$) should be written; which means from 1 to 2 we should pass the second edge.

The function index_edge gets the two adjacent nodes and gives the number of edge between them which is ordered by the number of the nodes. For instance in order to find out that the second edge of 1 is related to which node, we refer to the adjacent matrix.

Figure2 provides the adjacent matrix for the graph above.

	1	2	3	4	5
1	0	0	1	1	1
2	0	0	1	0	0
3	0	0	0	1	0
4	0	1	0	0	1
5	0	0	1	0	0

Fig. 2 A sample for the adjacent matrix designed for the graph G.

In the adjacent matrix the algorithm counts the number of 1s and the number of the comparing operation repetition. At the matrix in the example when it reaches to the second 1, it has done the comparing operation for 4 times. So it recognizes that the second edge of node 1 is related to the node 4. Therefore in order to reach to the node 2, it should pass the node 4.

The function index_vertex gets the number of a node and an edge connected to it and gives the number of the adjacent node related to that edge. To find the order of the edge, this algorithm uses logarithm in the basis of 2. In the example above (log 100) equals with 3 which means to the mentioned node we should pass the third edge. By this method we avoid the overload of clustering and the high complexity of for-loop.

We have also an adjacent matrix for the whole graph which its memory cost in a graph with so many nodes would be inconsiderable though its processing would be time consuming.

While the data transferring is active and one of the nodes runs away from the network, the number of rows in training matrix in which the route passes from that node (the number of the edge related to that node is 1) sets to zero. So the routing and training algorithms for the related destinations should be repeated.

B. Benefits of Using This Method

As it is mentioned, this algorithm doesn't optimize the routing protocol, but it guides and trains the network to learn the protocol once and use the route given for thousands of times. In addition while sending update amongst the route found, the nodes on the route are also updated. The important point is that if the destination is far from the source and it passes through the most of the nodes, then large amount of nodes become updated and this causes less repetition of routing protocol.

Being protocol-free, this method can be theoretically applied in all kinds of mobile or wireless ad-hoc networks and there would be no limitation in using this algorithm.

Propagating packets or messages for requesting and replying through the network causes the traffic to be heavier. But in our method after initial routing we never use such packets or messages.

As it is said before, the time complexity reduces to "L" after training, which means that the data has been transferred through the right path. Using the binary matrix for each node, the memory complexity for n nodes is n bytes. Less memory occupied and less time spent, makes this method a desirable way in data transfer on networks)

IV. Conclusions

Repeating the routing protocols for each node is a time consuming task. Currently applicable algorithms, in this regard, suffer from a high complexity of time which is discussed in section 2. In section 3 we have introduced a new algorithm, which by training the network reduces the number of times that the routing protocols are being repeated. After a few repetition of routing the nodes themselves learn from where they should send data.

A. Future Work

The method we presented in this paper sets the stage ready for an interesting topic of research:

- **Traffic Control on Ad-Hoc Networks.**
- **A New Approach:**

The algorithm introduced in this paper can set the bandwidth array, in addition to the training. The bandwidth array is adjusted to control the traffic of the route. While sending data and routing to the destination the algorithm increments a counter. This means that the path from the source to the destination has been used for the number of times the counter shows. Now, if we notice that one path is used so many times and one less, we can maintain the bandwidth of the path. This is a good way to control the traffic on the network. If the bandwidth has reached to the highest amount, new edges may be produced. It seems to be a good idea to manage the traffic on the network.

Acknowledgment

We would like to thank Dr. Ayaz Isazadeh, Dr. Mir Kamal Mirnia and Dr. Ahamad Habibizad Navin for his guidance and true helps in preparing the paper.

References

[1] I. Stojmenovic, “Position-Based Routing in Ad Hoc Networks”, IEEE Communications Magazine, pp. 2-3, July 2002.
[2] L. Barriere, P. Fraingniaud and L. Narayanan, “Robust Position-Based Routing in Wireless Ad Hoc Networks with Unstable Transmission Ranges”, Rome, Italy 2001 ACM, p.19.
[3] L. Zhou and Z. J. Haas, “Securing Ad Hoc Networks”, IEEE network, special issue on network security, p.1, November/December, 1999.
[4] S. P. Kon and R. V. Boppana, “On Reducing Packet Latencies in Ad Hoc Networks”, The University of Texas at San Antonio, WCNC 2000.
[5] Y. Hu, A. Perrig and D. B. Johnson, “Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols”, WiSe 2003, San Diego, California, USA Copyright September 2003 ACM. p.1.
[6] A. Bruce McDonaldk and Y. Taieb Znatiy, “A Path Availability Model for Wireless Ad-Hoc Networks”, Proc. IEEE WCNC’99, p.1.
[7] S. Buchegger and J. Le Boudec, “Cooperative Routing in Mobile Ad-hoc Networks: Current Efforts against Malice and Selfishness”, Proceedings of Mobile Internet Workshop, Informatik 2002, Dortmund, Germany, October 2002, p.2.
[8] S. Rupp and W. Fu, “Routing Mechanisms in Ad-Hoc Networks”, Oral Presentation, July 2003.
[9] Tian He, John A Stakovic, Chenyang Lu and Tarek Abdelzaher, “SPEED: A Stateless Protocol for Realtime Communication in Sensor Networks”, Department of Computer Science, University of Virginia, Dec 2002. p.9.
[10] T.H. Cormen, C.E. Leiserson and R.L. Rivest, “Introduc-tion to Algorithms”, MIT Press, 1990.
[11] J. Leguay, T. Ur Friedman and S. Fdida, “CONNECTIVITY AWARE ROUTING IN AD-HOC NETWORKS”, funded in part by Thales Communications and LIP6 through Euronetlab, in part by E-Next, and in part by the ANRT (Association Nationale de la Recherche Technique) through the CIFRE, 2004.
[12] L. Butty´an and J. Hubaux, “Report on a Working Session on Security in Wireless Ad Hoc Networks”, Mobile Computing and Communications Review, Vol. 6, No. 4, p.6.
[13] S. J. Philip, “Routing in Mobile Ad hoc Networks”, university at Buffalo, New York. Fall 2004.
[14] Y. B. Ko and N. H. Vaidya, “Location-Aided Routing (LAR) in mobile ad hoc networks”, Wireless Networks 6 (2000), supported in part by Texas Advanced Technology Program and National Science Foundation, J.C. Baltzer AG, Science Publishers. p. 307.
[15] D. B. Johnson and D. A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks”, Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Kluwer Academic Publishers, 1996.
[16] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu and J. Jetcheva, “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols”, in Proceedings of Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98), October, 1998, Dallas, Texas, USA. Copyright 1998 ACM. P.2.
[17] N. Milanovic, M. Malek A. Davidson and V. Milutinovic, “Routing and Security in Mobile Ad Hoc Networks”, the IEEE Computer Society, February 2004.
[18] J. Wu and F. Dai, “BROADCASTING IN AD HOC NETWORKS BASED ON SELF-PRUNING”, International Journal of Foundations of Computer Science World Scientific Publishing Company, supported in part by NSF, 2003, pp.2-4.
[19] P. Papadimitratos and Z. J. Haas, “Secure Routing for Mobile Ad hoc Networks”, In Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), San Antonio, TX, January 2002. p.3.
[20] M. Gerla, X. Hong and G. Pei, “Landmark Routing for Large Ad Hoc Wireless Networks”, In Proceeding of IEEE GLOBECOM 2000, San Francisco, CA, Nov. 2000.
[21] K. Wu and J. Harms, “QoS Support in Mobile Ad Hoc Networks”, Crossing Boundaries – an interdisciplinary journal Vol.1, No.1 - Fall 2001. p.99.
[22] E. Natsheh, S. Khatun and A. Jantan, “A Model of Route Lifetime Optimization with Linguistic Knowledge in Wireless Ad-hoc Networks,” Mathware & Soft Computing 16 (2006), pp. 23-24.

[23] C. Venkatesh, N. Yadaiah and A.M. Natarajan, "Dynamic source routing protocol using fuzzy logic concepts for ad hoc networks," Academic Open Internet Journal, www.acadjournal.com, Volume 15, 2005.
[24] Yi-an Huang, " Anomaly Detection for Wireless Ad-Hoc Routing Protocols," A thesis submitted to the Graduate Faculty of North Carolina State University in partial fulfillment of the requirements for the Degree of Master of Science, 2001, p. 30.
[25] Hongju Cheng, Xiaohua Jia, and Hai Liu, "Access Scheduling on the Control Channels in TDMA Wireless Mesh Networks," ICCSA 2007, LNCS 4706, Part II, pp. 247–248, 2007.
[26] Mustapha GUEZOURI and Abdelaziz OUAMRI, "Optimizing Routes Quality and Scattering in the AODV Routing Protocol," JCS&T Vol. 7 No. 3, October 2007, pp. 209-211.
[27] Chao Gui and Prasant Mohapatra, "SHORT: Self-Healing and Optimizing Routing Techniques for Mobile Ad Hoc Networks," MobiHoc'03, June 2003, Annapolis, Maryland-USA, Copyright 2003 ACM, p.279.
[28] N. H. Vaidya, "Mobile Ad Hoc Networks: Routing, MAC and Transport Issues", University of Illinois at Urbana-Champaign, 2001.