Volume 3, Issue 9, September 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Performance of Genetic Algorithm Based Intelligent Reactive Protocols Routing for MANET

Arun Biradar*
Professor
Department of Computer Science & Engg.
East West Institute of Technology
Bangalore, Karnataka, India

Dr. Ravindra C. Thool
Professor
Department of Inofrmation Technology
SGGS Institute of Engg. & Technology
Nanded, Maharashtra, India

Abstract— Mobile Ad hoc Networks (MANET) are wireless network which have no central bridge, no fixed routers and no centralized administration. All nodes may move randomly and are connecting dynamically to each other.The main aim of the routing protocol is to have an efficient route establishment between a pair of nodes, so that messages can be delivered in a timely manner. Routing in the MANETs is a challenging task which has led to development of many different routing protocols for MANETs.There are many protocols that have been developed to aid in routing in these types of networks and are generally classified as either proactive or reactive. Each of these protocols is designed with some certain mobility scenario in mind. To achieve effective routing in a given scenario, the right protocol must be chosen. Choosing the right protocol involves evaluating the performance metrics that defines the effectiveness of a routing protocol namely; Packet Delivery Ratio, Throughput and Average End to End Delay.
There are number of issues in MANET among those routing is one here we are focusing on routing in MANETS. Here we have compared the performance of AODV and AOMDV. This paper aim is to propose an genetic algorithm based intelligent routing technique using ad-hoc on-demand multipath distance vector routing (AOMDV) for MANET.GA mechanisms allow a node to change routing information quickly and efficiently to adjust an ever changing local topology, initiating fewer link breakages. We use NS-2 simulator to simulate this protocol and conclude with performance evaluation and comparison are performed through extensive simulation studies.Simulation results show the improvements over the conventional routing algorithm in terms of various parameters.

Keywords— Mobile Ad hoc Networks (MANETs), Ad-hoc On-demand Distance Vector (AODV), Ad-hoc On-demand Multipath Distance Vector (AOMDV), Route Request (RREQ), Route Reply (RREP), Route Error (RERR).

I. INTRODUCTION

With the development in computer and wireless communication technologies, mobile wireless networks are increasingly widespread used. The wireless Ad Hoc networks are received more and more attention due to their characters of no network administration and infrastructure.Wireless Ad Hoc networks owns some characteristics like self-organization, multi-hop and a frequently-changed topology. The uniqueness of Wireless Ad Hoc Network made it impossible to adopt the routing technologies utilized in traditional networks. So various routing protocols that are tailored to wireless Ad Hoc networks have been put up in recent years by researchers around the world and the related analyses are being done from different aspects. Since nodes in the wireless Ad Hoc networks move freely and randomly, how to improve the route reliability becomes the central challenge in such dynamic wireless networks. We focus on the routing protocols of MobileAd Hoc networks and we have used the multi-path routing techniques as our main research point.

Among the on-demand protocols, multipath protocols have a relatively greater ability to reduce the routediscovery frequency than single path protocols. On demand multipath protocols discover multiple pathsbetween the source and the destination in a single routediscovery. So, a new route discovery is needed onlywhen all these paths fail. In contrast, a single pathprotocol has to invoke a new route discovery whenever the only path from the source to the destination fails.Thus, on-demand multipath protocols causes fewer interruptions to the application data traffic when routes fail. They also have the potential to lower the routingoverhead because of fewer route discovery operations. On-demand multipath protocol called ad hoc on-demand multipath distance vector (AOMDV). AOMDV is based on a prominent and well-studied on-demand single pathprotocol known as ad hoc on-demand distance vector (AODV) [4,5]. AOMDV extends the AODV protocol to discover multiple paths between the source and the destination in every route discovery. Multiplepaths so computed are guaranteed to be *loop-free* and *disjoint*. The Genetic Algorithm based routing of AOMDV is efficient than the conventional AOMDV in terms of End-to-end delay, throughput, PDF.

II. BACKGROUND

Wireless Ad-hoc Networks are a collection of two or more devices equipped with wireless communications and networking capability. These devices can communicate with other nodes that immediately within their radio range or one that is outside their radio range. For the later, the nodes should deploy an intermediate node to be the router to route the packet from the source toward the destination. The Wireless Ad-hoc Networks do not have gateway, every node can act as the gateway. Almost every wireless network nodes communicate to base-station and access points there by co-operating to forward packets hop-by-hop.

Many routing protocols have been proposed for MANETs. These routing protocols can be divided into three categories:

A. Proactive (table-driven) Routing Protocols
B. Reactive(on-demand) Routing Protocols
C. Hybrid routing protocols

Based on when and how the routes are discovered. In table driven routing protocols consistent and up-to-date routing information to all nodes is maintained at each node whereas in on-demand routing the routes are created only when desired by the source host. Next two sections discuss current table-driven protocols, hybrid protocols as well as on demand routing protocols.

A. Proactive(table-driven) Routing Protocols

In Proactive routing protocols each node maintains one or more tables containing routing information to every other node in the network. All nodes update these tables so as to maintain a consistent and up-to-date view of the network. When the network topology changes the nodes propagate update messages throughout the network in order to maintain consistent and up-to-date routing information about the whole network. These routing protocols differ in the method by which the topology change information is distributed across the network and the number of necessary routing-related tables. Examples: The Destination-Sequenced Distance-Vector (DSDV), Optimized Link State Routing protocol (OLSR).

D. Reactive(on-demand) Routing Protocols

Reactive protocols take a lazy approach to routing. In contrast to table-driven routing protocols all up-to-date routes are not maintained at every node, instead the routes are created as and when required. When a source wants to send to a destination, it invokes the route discovery mechanisms to find the path to the destination. The route remains valid till the destination is reachable or until the route is no longer needed.

Examples: Ad hoc On-demand Distance Vector Routing (AODV), The Dynamic Source Routing Protocol (DSR), Ad hoc On demand Multipath Distance Vector routing protocol (AOMDV).

E. Hybrid routing protocols

This protocol combines best features of the proactive and reactive protocols. Nodes within a certain distance from the concerned node or within a particular geographical region are said to be within the routing zone of a given node. For routing within this zone, a table driven approach is used. For nodes that are located beyond this zone, an on-demand approach is used. Examples: Zone Routing Protocol (ZRP),Wireless Routing Protocol (WRP).

III. REACTIVE PROTOCOLS

This paper focuses on reactive protocols for MANET mainly unipath AODV, multipath AOMDV as below.

A. Ad-hoc On-demand Distance Vector Protocol (AODV)

Ad hoc On-Demand Distance Vector (AODV) is a reactive routing protocol which creates a path to destination when required. Routes are not built until certain nodes send route discovery message as an intention to communicate or transmit data with each other. Routing information is stored only in the source node, the destination node, and the intermediate nodes along the active route which deals with data transmission. This scenario decreases the memory overhead, minimize the use of network resources, and run well in high mobility situation.

Whenever an AODV router receives a request to send a message, it checks its *routing table* to see if a route exists.

Each routing table entry consists of the following fields:

· Destination address
· Next hop address
· Destination sequence number
· Hop count

If a route exists, the router simply forwards the message to the next hop. Otherwise, it saves the message in a *message queue*, and then it initiates a route request to determine a route. The following flow chart illustrates this process:

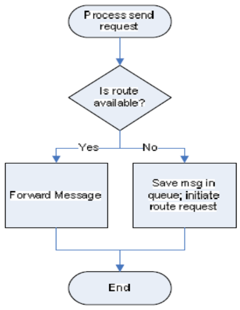

Fig. 1 Request Process in AODV

AODV nodes use four types of messages to communicate among each other. *Route Request (RREQ)* and *Route Reply (RREP)* messages are used for route discovery. *Route Error (RERR)* messages and *HELLO* messages are used for route maintenance. The following sections describe route determination and route maintenance in detail.

1) AODV Route Discovery

When a node needs to determine a route to a destination node, it floods the network with a *Route Request (RREQ) message*. The originating node broadcasts a RREQ message to its neighboring nodes, which broadcast the message to their neighbors, and so on. To prevent cycles, each node remembers recently forwarded route requests in a route request buffer As these requests spread through the network, intermediate nodes store reverse routes back to the originating node. Since an intermediate node could have many reverse routes, it always picks the route with the smallest hop count.

When a node receiving the request either knows of a "fresh enough" route to the destination, or is itself the destination, the node generates a *Route Reply (RREP) message*, and sends this message along the reverse path back towards the originating node. As the RREP message passes through intermediate nodes, these nodes update their routing tables, so that in the future, messages can be routed though these nodes to the destination.

Notice that it is possible for the RREQ originator to receive a RREP message from more than one node. In this case, the RREQ originator will update its routing table with the most "recent" routing information; that is, it uses the route with the greatest destination sequence number.

2) Route Request Buffer

In the flooding protocol described above, when a node originates or forwards a route request message to its neighbors, the node will likely receive the same route request message back from its neighbors. To prevent nodes from resending the same RREQs (causing infinite cycles), each node maintains a *route request buffer*, which contains a list of recently broadcasted route requests. Before forwarding a RREQ message, a node always checks the buffer to make sure it has not already forwarded the request. RREQ messages are also stored in the buffer by a node that originates a RREP message. The purpose for this is so a node does not send multiple RREPs for duplicate RREQs that may have arrived from different paths. The exception is if the node receives a RREQ with a better route (i.e. smaller hop count), in which case a new RREP will be sent. Each entry in the route request buffer consists of a pair of values: the address of the node that originated the request, and a route request identification number (RREQ id). The RREQ id uniquely identifies a request originated by a given node. Therefore, the pair uniquely identifies a request across all nodes in the network. To prevent the route request buffers from growing indefinitely, each entry expires after a certain period of time, and then is removed. Furthermore, each node's buffer has a maximum size. If nodes are to be added beyond this maximum, then the oldest entries will be removed to make room.

3) Sequence Numbers

Each destination (node) maintains a monotonically increasing sequence number, which serves as a logical time at that node. Also, every route entry includes a destination sequence number, which indicates the "time" at the destination node when the route was created. The protocol uses sequence numbers to ensure that nodes only update routes with "newer" ones. Doing so, we also ensure loop- freedom for all routes to a destination. All RREQ messages include the originator's sequence number, and its (latest known) destination sequence number. Nodes receiving the RREQ add/update routes to the originator with the originator sequence number, assuming this new number is greater than that of any existing entry. If the node receives an identical RREQ message via another path, the originator sequence numbers would be the same, so in this case, the node would pick the route with the smaller hop count. If a node receiving the RREQ message has a route to the desired destination, then we use sequence numbers to determine whether this route is "fresh enough" to use as a reply to the route request. To do this, we check if this node's destination sequence number is at least as great as the maximum destination sequence number of all nodes through which the RREQ message has passed. If this is the case, then we can roughly guess that this route is not terribly out-of-date, and we send a RREP back to the originator. As with RREQ messages, RREP messages also include destination sequence numbers. This is so nodes along the route path can update their routing table entries with the latest destination sequence number.

4) Link Monitoring & Route Maintenance

Each node keeps track of a *precursor list*, and an *outgoing list*. A precursor list is a set of nodes that route through the given node. The outgoing list is the set of next-hops that this node routes through. In networks where all routes are bi-directional, these lists are essentially the same. Each node periodically sends HELLO messages to its precursors. A node decides to send a HELLO message to a given precursor only if no message has been sent to that precursor recently. Correspondingly, each node expects to periodically receive messages (not limited to HELLO messages) from each of its outgoing nodes. If a node has received no messages from some outgoing node for an extended period of time, then that node is presumed to be no longer reachable. Whenever a node determines one of its next- hops to be unreachable, it removes all affected route entries, and generates a Route Error (RERR) message. This RERR message contains a list of all destinations that have become unreachable as a result of the broken link. The node sends the RERR to each of its precursors. These precursors update their routing tables, and in turn forward the RERR to their precursors, and so on. To prevent RERR message loops, a node only forwards a RERR message if at least one route has been removed.

The following flow chart summarizes the action of an AODV node when processing an incoming message.

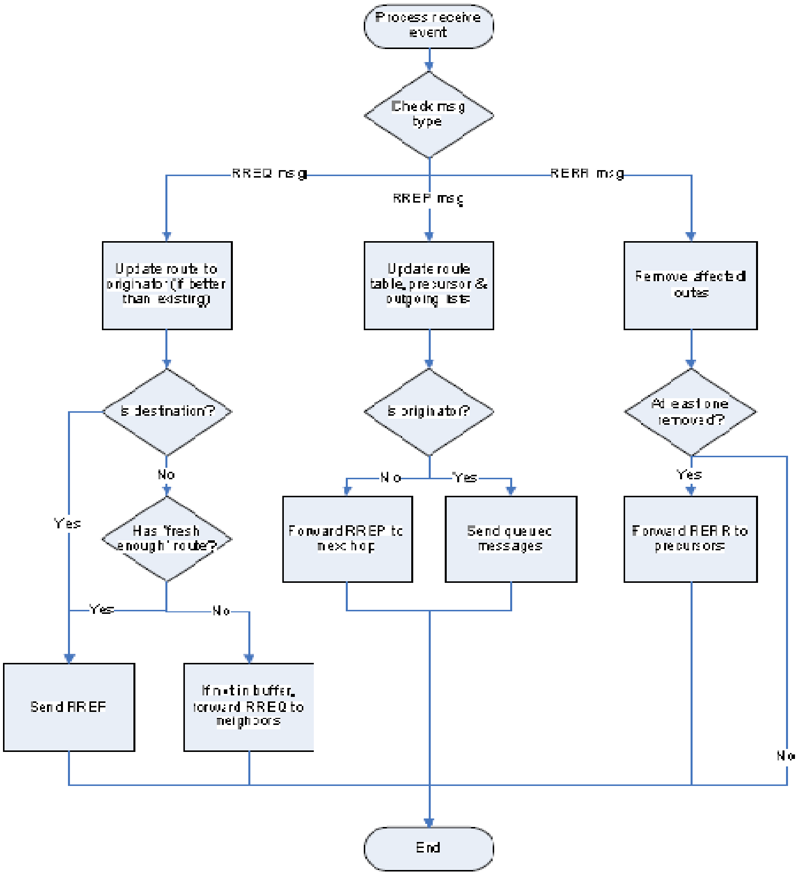

Fig. 2 Action of AODV Node

B. Multipath Routing

Mobile ad hoc networks (MANETs) are characterized by a dynamic topology, limited channel bandwidth and limited power at the nodes. Because of these characteristics, paths connecting source nodes with destinations may be very unstable and go down at any time, making communication over ad hoc networks difficult. On the other hand, since all nodes in an ad hoc network can be connected dynamically in an arbitrary manner, it is usually possible to establish more than one path between a source and a destination. When this property of ad hoc networks is used in the routing process, we speak of multipath routing.

In most cases, the ability of creating multiple routes from a source to a destination is used to provide a backup route. When the primary route fails to deliver the packets in some way, the backup is used. This provides a better fault tolerance in the sense of faster and efficient recovery from route failures.

Multiple paths can also provide load balancing and route failure protection by distributing traffic among a set of disjoint paths. Paths can be disjoint in two ways: (a) link-disjoint and (b) node-disjoint. Node-disjoint paths do not have any nodes in common, except the source and destination, hence the do not have any links in common. Link-disjoint paths, in contrast, do not have any links in common. They may, however, have one or more common nodes (see figure 3).

In order to use multiple paths simultaneously they need to be as independent as possible. So not only do they need to be disjoint, also route coupling must be taken into account, because routes can interfere with each other. Route coupling takes place when a path crosses the radio coverage area of another path. There is a protocol that uses this property of radio broadcast to create backup-routes, but in the case of multiple-path data transport route coupling is unwanted. Routes may be link- or even node-disjoint but still interfere with each other due to route coupling.

Consider the node-disjoint routes of figure 3a again. In the situation of figure 3c, when node *a* for example sends data to node b (both route 1), node *d* on the other route cannot transmit data to *e* on route 2, since the nodes (and thus routes) are in each other's radio coverage area and interfere with each other. Since none of the routing protocols take the route coupling into account, we will ignore it in the sequel. Disjointness will be the only measure used for path independence.

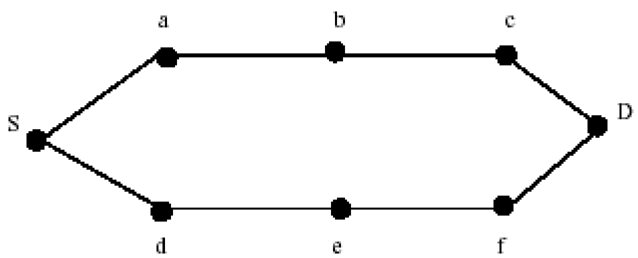

Fig. 3a Two node-disjoint paths from source *S* to destination *D*.

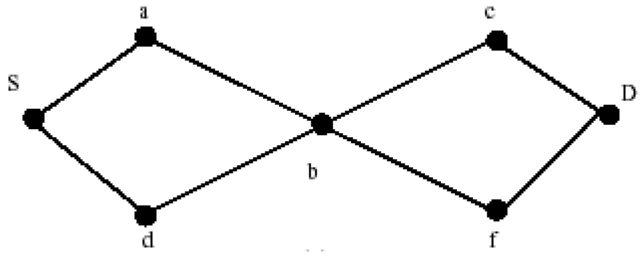

Fig. 3b Two link-disjoint paths from source *S* to destination *D.* Note that they are not node-disjoint, since they share node *b*.

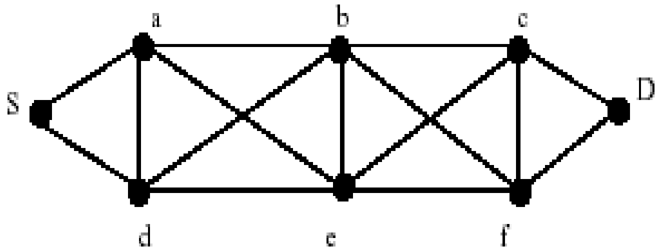

Fig. 3c The two node-disjoint paths from figure 1a, when they are in each other's radio coverage.

1) Ad-hoc On-demand Multipath Distance Vector Protocol (AOMDV)

Ad-hoc On demand Multi path Distance Vector Routing (AOMDV) Protocol is an extension to AODV protocol for multiple loop-free and link disjoint paths.

The main idea in AOMDV is to compute multiple paths during route discovery. It consists of two components:

- A route update rule to establish and maintain multiple loop-free paths at each node.
- A distributed protocol to find link-disjoint paths.

Before describing AOMDV, we first discuss AODV, from which it is derived. In AODV, when a source needs a route to a destination, it initiates a route discovery process by flooding a RREQ for destination throughout the network. RREQs should be uniquely identified by a sequence number so that duplicates can be recognized and discarded. Upon receiving a non-duplicate RREQ, an intermediate node records previous hop and checks whether there is a valid and fresh route entry to the destination in routing table. If such is the case, the node sends back a RREP to the source; if not it rebroadcasts the RREQ. A node updates its routing information and propagates the RREP upon receiving further RREPs only if a RREP contains either a larger destination sequence number (fresher) or a shorter route found.

In AOMDV each RREQ, respectively RREP arriving at a node potentially defines an alternate path to the source or destination. Just accepting all such copies will lead to the formation of routing loops. In order to eliminate any possibility of loops, the "advertised hop count" is introduced. The advertised hopcount of a node i for a destination d represents the maximum hopcount of the multiple paths for d available at i. The protocol only accepts alternate routes with hopcount lower than the advertised hopcount, alternate routes with higher or the same hopcount are discarded.

The advertised hopcount mechanism establishes multiple loop-free paths at every node. These paths still need to be disjoint. For this we use the following notion:

When a node S floods a RREQ packet in the network, each RREQ arriving at node I via a different neighbor of S, or S itself, defines a node-disjoint path from I to S. (For proof see [5]).

In AOMDV this is used at the intermediate nodes. Duplicate copies of a RREQ are not immediately discarded. Each packet is examined to see if it provides a node-disjoint path to the source. For node-disjoint paths all RREQs need to arrive via different neighbors of the source. This is verified with the firsthop field in the RREQ packet and the firsthop_list for the RREQ packets at the node.

At the destination a slightly different approach is used, the paths determined there are link-disjoint, not node-disjoint. In order to do this, the destination replies up to k copies of the RREQ, regardless of the firsthops. The RREQs only need to arrive via unique neighbors.

2) AOMDV Properties

- Extension of AODV.
- RREQs from different neighbors of the source are accepted at intermediate nodes.
- Multiple link-disjoint routes are created (with modification at the destination they can be node-disjoint).
- Maximum hopcount to each destination ("advertised hopcount") is used to avoid loops.
- Multiple routes are established in single route discovery process.
- Nodes maintain next-hop info for destinations (multiple next-hops possible).
- No complete route(s) information known at a source.

IV. GENETIC ALGORITHM

Holland (1975) introduced the formalism of genetic algorithm (GAs) by analogy with how biological evolution occurs in Nature. Deep down under, a computer program is nothing but a string of 1s and 0s, something like 110101010110000101001001….. This is similar to how chromosomes are laid out along the length of a DNA molecule. We can think of each binary digit as a 'gene', and a string of such genes as a digital 'chromosome'.

For example, chromosome A may be 1101, B may be 0111, etc. In a GA, an individual in the population is represented schematically as the sequence of its chromosomes, say ABCDEF. Another individual may be abcdef, a third may be aBCdeF, etc.

The essence of Darwinian evolution is that, in a population, the fittest have a larger likelihood of survival and propagation. In computational terms, it amounts to maximizing some mathematical function representing 'fitness'.

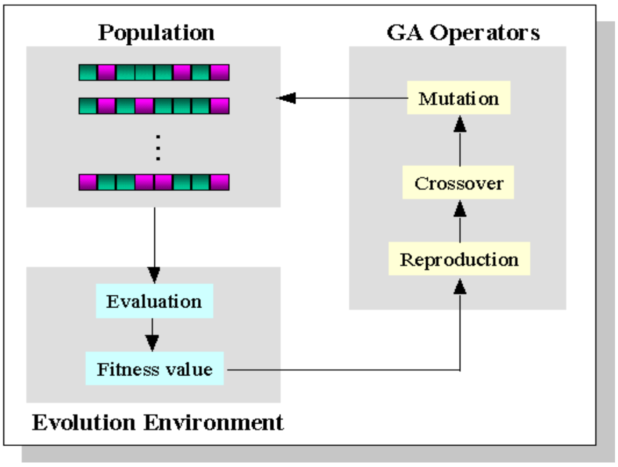

Fig. 4 Structure of GA

It is important to remember that whereas Darwinian evolution is an open-ended and *blind*, process, GAs have a goal. GAs is meant to solve particular pre-conceived problems.

For solving a maximization problem (e.g. for finding the maximum possible value of a complicated function (say fitness) of an artificial genome), the steps involved are typically as follows:

1. We first let the computer produce a population of, say, 1000 individuals, each represented by a randomly generated digital chromosome.
2. The next step is to test the relative fitness of each individual (represented entirely by the corresponding chromosome) regarding its effectiveness in maximizing the function under consideration; e.g. the fitness function. A score is given for the fitness, say on a scale of 1 to 10. In biological terms, the fitness is a probabilistic measure of the reproductive success of the individual. The higher the fitness, the greater is the chance that the individual will be selected (by us) for the next cycle of reproduction.
3. Mutations are introduced occasionally in a digital chromosome by arbitrarily flipping a 1 to 0, or a 0 to 1.
4. The next step in the GA is to take (in a probabilistic manner) those individual digital chromosomes that have high levels of fitness, and produce a new generation of individuals by a process of digital sexual reproduction or crossover. The GA chooses pairs of individuals, say ABCDEF and abcdef, and produces two new individuals, say, ABCdef and abcDEF.

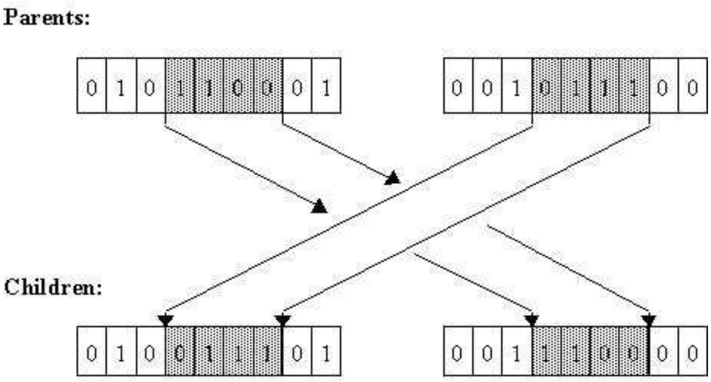

Fig. 5 Crossover

5. The new generation of digital individuals produced is again subjected to the entire cycle of gene expression, fitness testing, selection, mutation and crossover.

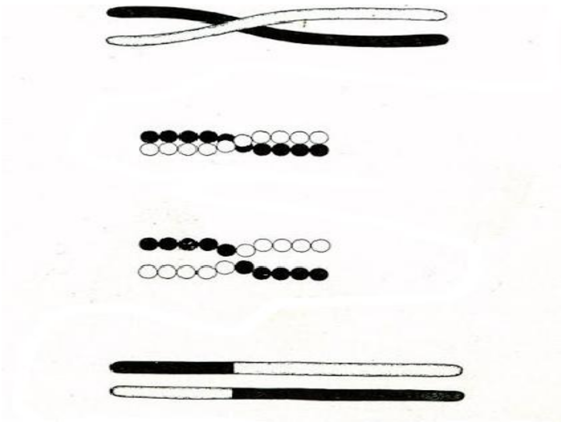

Fig. 6 Scheme to illustrate a method of Crossing Over of Chromosomes

6. These cycles are repeated a large number of times, till the desired optimization or maximization problem have been solved.

V. RESULT ANALYSIS

A. Performance Metrics

We use different parameters to compare the performance are described below:

1) Packet Delivery Fraction

The ratio of the data packets delivered to the destinations to those generated by the CBR sources is known as packet delivery fraction.

2) Average End-to-End Delay

Average end to end delay includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times of data packets.

3) Normalized Routing Load

The number of routing packets transmitted per data packet delivered at the destination. Each hop wise transmission of a routing packet is counted as one transmission. The routing load metric evaluates the efficiency of the routing protocol.

4) Throughput

Total number of delivered data packets divided by the total duration of simulation time. We analyze the throughput of the protocol in terms of number of messages delivered per second.

B. Result Analysis

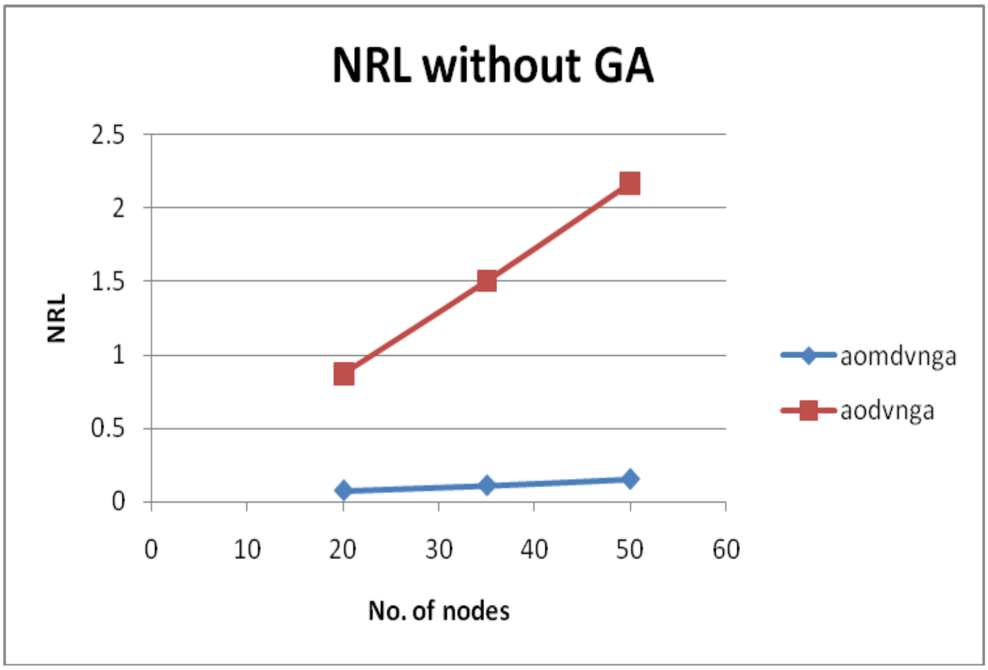

Fig. 7 NRL without GA

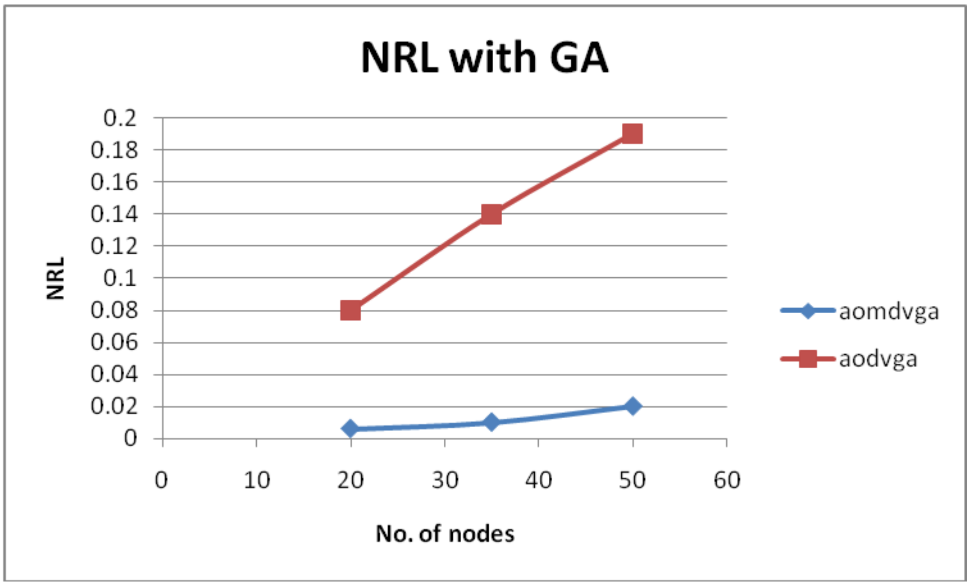

Fig. 8 NRL with GA

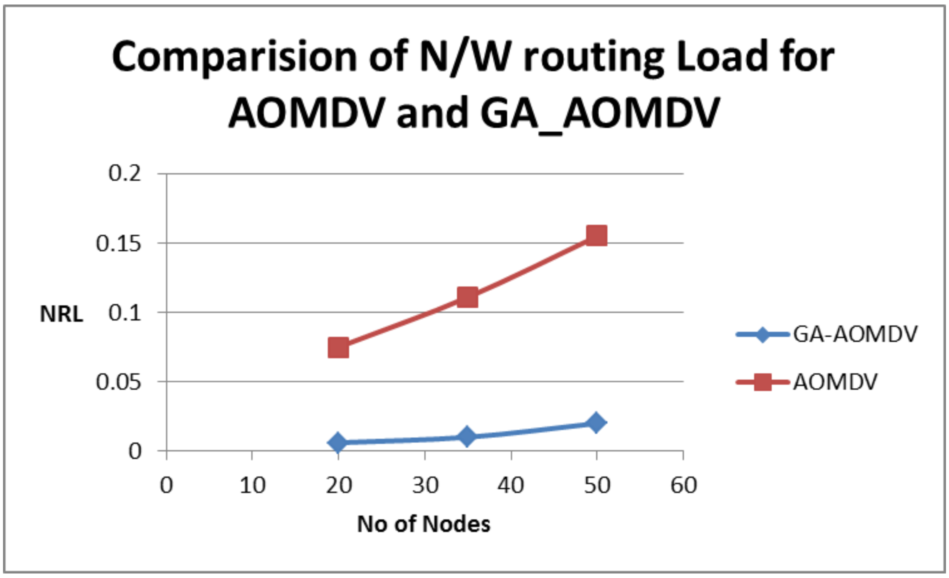

Fig. 9 NRL with GA-AOMDV

VI. CONCLUSIONS

In this paper the performance of AODV and AOMDV routing protocols is analysed by using NS2 simulation. The comprehensive simulation results of average end to end delay normalized routing load, Packet Delivery Fraction and throughput by varying the number of nodes and number of connections. After analyzing in different situation of network, it can be concluded that Genetic Based AOMDV Performs better than traditional AOMDV.

ACKNOWLEDGMENT

We are thankful to the Department of Computer Science & Engineering and Information Technology of Shri Guru Govind Singhji Institute of Engineering and Technology (SGGSIE&T), Nanded, Maharashtra, India.

REFERENCES

[1] S. Das, C. Perkins and E. Royer, "*Ad Hoc On Demand Distance Vector (AODV) Routing*", IETF RFC3561, July 2003.

[2] D. Johnson, "*The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR*)", IETF Internet Draft, draft-ietf-manet-dsr-09.txt, April 2003.

[3] P. Jacquet and T. Clausen, "*Optimized Link State Routing Protocol*", IETF Internet Draft, draft-ietf-manet-olsr-11.txt, July 2003.

[4] M. Lewis, F. Templin and R. Ogier, "*Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)*", IETF Internet Draft, draft-ietf-manet-tbrpf-09.txt, June 2003.

[5] M. Marina and S. Das, "*On-demand Multipath Distance Vector Routing in Ad Hoc Networks*", in Proceedings of the International Conference for Network Procotols (ICNP), Riverside, Nov. 2001.

[6] V. Park and M. Corson, ”A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks”, Proceedings of IEEE INFOCOM '97, April 1997.

[7] R. Leung, J. Liu, E. Poon, A. Chan and B. Li, “*MP-DSR: A QoS-Aware Multi-Path Dynamic Source Routing Protocol for Wireless Ad-Hoc Networks*”, In Proc. of the 26th IEEE Annual Conference on Local Computer Networks (LCN 2001), pp. 132-141, November, 2001.

[8] D.B. Johnson, D.A. Maltz, and J. Broch, "*DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks*", Ad Hoc Networking, pp. 139-172, 2001.

[9] J. Schaumann, “Analysis of the Zone Routing Protocol”, December 2002.

[10] Z. Haas and M. Pearlman, "*The zone routing protocol (ZRP) for Ad Hoc networks*", IETF Internet Draft, draft-ietf-manet-zone-zrp-04.txt, July 2002.

[11] Z. Haas, “A New Routing Protocol for the Reconfigurable Wireless Networks”, Proceedings of IEEE ICUPC’97, San Diego, CA, pp. 562-566, October 1997.

[12] M. Pearlman, Z. Haas, P. Sholander and S. Tabrizi, “On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad Hoc Networks”, MobiHoc’2000, August 2000.

[13] L. Wang, Y. Shu, M. Dong, L. Zhang and O. Yang, “*Adaptive Multipath Source Routing in Ad Hoc Networks*”, IEEE ICC 2001, Page(s): 867 -871 vol.3, June 2001.

[14] L. Wang, Y. Shu, Z. Zhao, L. Zhang and O. Yang, “*Load Balancing of Multipath Source Routing in Ad Hoc Networks”,* Proceedings of IEEE ICC’02, April 2002.

[15] J. Raju and J. Garcia-Luna-Aceves, “*A New Approach to On-demand Loop-Free Multipath Routing*”, In Proc. Of the 8th Annual IEEE International. Conf. Computer Communications and Networks (ICCCN), Boston, MA, Oct 1999, pp. 522--527.

[16] S. Corson and J. Macker, “*Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations*”, IETF WG Charter, http://www.ietf.org/html.charters/manet-charter.html, January 1999.

[17] Goldberg, D.E., *Genetic Algorithms in search, optimization, and machine learning* 1989, Boston: Addison Wesley

[18] C. W. Ahn, and R. S. Ramakrishna, “A "Genetic algorithm for Shortest Path Routing Problem and the Sizing of Populations" *EEE Transactions on Evolutionary computation, vol.6, no.6*, Dec. 2002.

[19] Mustafa Al-Ghazal, Ayman El-Sayed, Hamedy Kelash “Routing Optimization using Genetic Algorithm in Ad Hoc Networks”, *IEEE Int. Symposium on Signal Processing and Information Technology*, pages 497-503,2007

[20] Gihan Nagib and Wahied G. Ali" Network Routing Protocol using Genetic Algorithms" *International Journal of Electrical & Computer Science IJECSIJENS Vol:10 No:02*,2010.

[21] http://1.bp.blogspot.com

