Asian Journal of Information Technology 5 (9): 1009-1016, 2006
© Medwell Online, 2006

Application of Genetic Algorithm in Mobile Ad Hoc Network

[1]M. Roberts Masillamani and [2]G.V. Uma
Department of Computer Science and Engg, College of Engineering,
Anna University, Guindy, Chennai, India-600 025, India

Abstract: This study focuses on maintaining message continuity in a mobile Ad Hoc network when there is mobility Genetic Algorithm (GA) is introduced to provide continuity in packet routing to reach the next possible neighboring node in order to avoid resending the packet from the source. GA learns the new routes through crossover and mutation. The algorithm gives next alternative route immediately once there is a break in the link. Simulations have been performed to analysis the performance of AODV with GA and AODV without GA.

Key words: Genetic algorithm, adhoc network, AODV, intelligent route rebuild

INTRODUCTION

An Ad Hoc network is a collection of mobile nodes that dynamically form a temporary network. Unlike traditional wireless and mobile networks in which mobile nodes communicate with a centralized structure, an Ad Hoc network operates without the use of existing network infrastructure. The main application for Ad Hoc networks has long been considered as the military tactical communication known as packet radio networks. Soldiers, tanks and planes are able to relay information for situational awareness on battlefield in an Ad Hoc network. An Ad Hoc network also could have commercial applications such as using laptop computers to participate in an interactive lecture, business associates sharing information during a meeting or conference.

Routing protocols for Ad Hoc wireless networks normally call for mobility management and scalable design. Mobility management is done through information exchanges between moving hosts in the Ad Hoc wireless network. In general, when information exchanges occur frequently, the network maintains accurate information of host locations and other relevant information. Frequent information exchanges can be costly, because they consume communication resources including bandwidth and power. With less frequent information exchanges, these costs diminish but there is more uncertainty about the host's location. Scalable design (one that works for large size networks) requires both routing protocols and resource consumptions to be scalable.

Routing in the Ad Hoc wireless network poses special challenges because of its infrastructure less network and its dynamic topology. The tunnel-based triangle routing of mobile IP[1] works well if there is a fixed infrastructure to support the concept of the home agent, since, all hosts move including the home agent. Traditional routing protocols for wired networks, that generally use either link state[2] or distance vector[3], are no longer suitable for Ad Hoc wireless networks. In an environment with mobile hosts as routers, convergence to new, stable routes after dynamic changes in network topology may be slow and this process could be expensive due to low bandwidth. Routing information has to be localized to adapt quickly to changes such as hosts movement. Routing protocols for Ad Hoc wireless networks[4-6] can be roughly divided into proactive and reactive. In proactive routing, each host continuously maintains complete routing information of the network. Both link state and distance vector belong to proactive routing. The reactive scheme on the other hand, invokes a route determination procedure only on demand through a query/reply study. Dynamic source routing protocol is a reactive routing pro-packet switching networks. Multipath routing is based on constructing first either edge disjoint paths or node disjoint paths with former being a special case of latter.

The multipath routing is also captured in Open Shortest Path First (OSPF)[7], a link state routing protocol in Internet, where traffic should be split equally between all the equal cost paths. Among protocols for Ad Hoc wireless networks, the Temporary Ordered Routing Algorithm (TORA)[8] maintains a Directed Acyclic Graph (DAG) for each destination with the destination being the sink of the DAG. Edge disjoint paths are maintained for each destination. Multipath extension to DSR is proposed[9] by constructing node disjoint paths. The destination keeps a record of the first arrived packet (including the complete path record initiated from a particular source). The subsequent packets arrived will be discarded until a packet with a node disjoint path with

Corresponding Author: Roberts Masillamani, M., Department of Computer Science and Engg, College of Engineering, Anna University, Guindy, Chennai, India-600 025, India

respect to the first one arrives. All subsequent packets are discarded. Unlike the approach in where the destination selects two node disjoint paths, the study generates two node disjoint paths during the query phase of the route discovery process by restricting the way the query packet is flooded.

Multi path routing and its applications[10,11] have been well studied in the networking literature, in particular for wired networks. Multi path routing enables fault tolerance and also facilitates load balancing. An early work on an application of multi path routing known as dispersity routing discusses how a message can be dispersed along multiple paths by splitting it in order to achieve smaller average delay and delay variance. Since then, there has been a significant amount of work done on multi path routing for both connection oriented[12,13] and connection-less technologies[14,15].

Distributed protocols are used to compute multiple disjoint and loop-free paths[16] describe a distributed algorithm to find shortest pairs of node- (link-) disjoint paths. Multi path distance vector algorithms that use diffusing computations to construct and maintain DAGs have been proposed[17,18]. All the above algorithms have high overheads that make them inefficient for bandwidth limited wireless networks. They are designed to work in the framework of proactive protocols, prevalent in the Internet, where overheads are not such major concern.

There has been some interest in the Ad Hoc networking to employ multi path routing algorithms. Two of the on-demand protocols, DSR[19] and TORA[20] have built-in capability to compute multiple paths. But either of them suffers from a different set of performance problems.

DSR uses source routing[21], by virtue of which it can detect loops easily and can gather a lot of routing information per route discovery. Aggressive use of route caching, lack of effective mechanisms to purge stale routes and cache pollution leads to problems such as stale caches and reply storms[22]. These problems not just limit the performance benefits of caching multiple paths, they can even hurt performance in many cases[23,24]. These problems are addressed[25]. TORA , on the other hand, builds and maintains multiple loop free paths without use of source routing. It also can detect network partitions. TORA uses an idea based on link reversals[26] to recover from link failures. Performance studies have shown that TORA suffers from high overheads primarily because of the requirement of reliable, in-order delivery of routing control messages.

Routing On-demand A cyclic Multi path (ROAM) is an on-demand, multi path distance vector algorithm based on diffusing computations. Like TORA, ROAM can also detect network partitions. But on the downside, state information must be maintained at each node during route discovery; this requires close coordination between nodes increasing overheads. Thus ROAM is better suited for static Ad Hoc networks or networks with low node mobility. A technique is proposed to allow AODV to maintain backup routes at the neighboring nodes of a primary route. This can be done with no additional overheads. The idea is to avoid dropping packets in flight when the primary route fails.

Every node still has utmost one route per destination just as in AODV. None of the above mentioned protocols explicitly take path breakage into account. Path breakage[27] has been considered in[28], but all of them use source routing. A proposal to extensions[29] to DSR to compute multiple disjoint paths for overhead reduction in mobile networks. They study the effect of number of multiple paths, path lengths on routing performance using analytical modeling and packet-level simulations. Split Multi path Routing is another disjoint multi path protocol using source routing. SMR is similar to multi path DSR except that the former uses a modified flooding algorithm and the data traffic is split among the multiple paths. Analysis of the performance impacts of alternative path routing for load balancing is done. They use a technique called diversity injection to compute node-disjoint paths. In[30], an improvement over the diversity injection technique[31] is proposed to find more node-disjoint paths.

AODV[32] combines the use of destination sequence numbers in DSDV[33] with the on-demand route discovery technique[34] in DSR to formulate a loop-free, on-demand, single path, distance vector protocol. Unlike DSR, which uses source routing, AODV is based on hop-by-hop routing study.

Problem definition: Wireless Ad Hoc networks are designed to collect information in unattended environments, e.g: monitoring a bush fire in remote forests. Such networks consist of small low-cost, resource-limited nodes that communicate wirelessly and cooperate to forward data in a multi-hop fashion. The open nature of the wireless communication, the lack of infrastructure, the hostile deployment environments, makes them vulnerable to a wide range of security attacks. In particular, a miscreant can very easily compromise certain sensor nodes and then inject false data into the network to mess up the monitoring operation.

Most of the events that occur in the monitored area exhibit high spatio-temporal correlation, which can be used to minimize the ill effects caused by falsely injected data. Intelligent building in adhoc network is an important factor for the ad hoc communication to be effective. In practice, due to mobility of nodes, frequent disconnections and reconnections take place. This leads

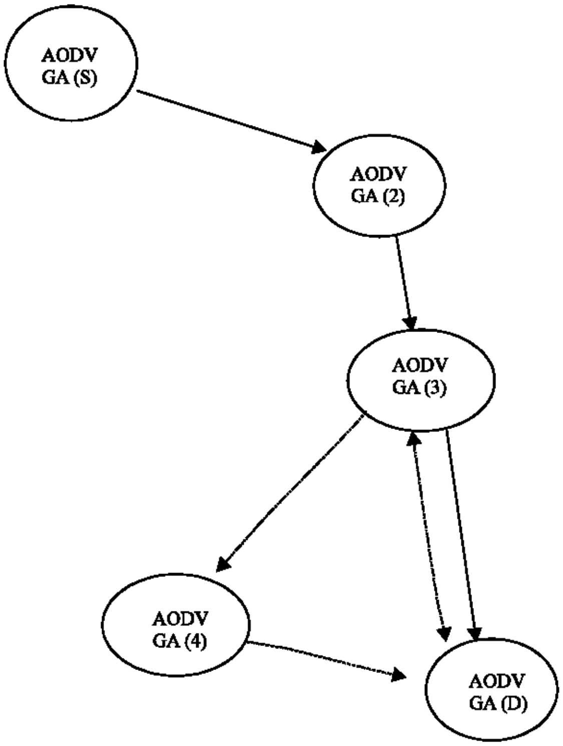

Fig. 1: Proposed Ad Hoc net

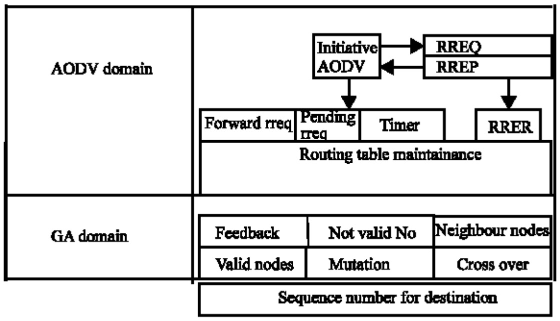

Fig. 2: System architecture

to not efficiently transferring all the packets from source to destination. In addition, high MAC load , more power consumption, flooding the route table and not getting deleted with the unused route information that complicates the functioning of the Ad Hoc network. This has led to the development of man protocols followed and preceded by AODV. Still problems exist in properly transferring packets from broken link to destination node.

To overcome the above problems and to reduce the MAC load, quick route rebuilding has to be done. To achieve this, the concept of genetic algorithm is attempted to coordinate with the AODV protocol for fast route rebuilding. The major advantages that are obtained by this study are

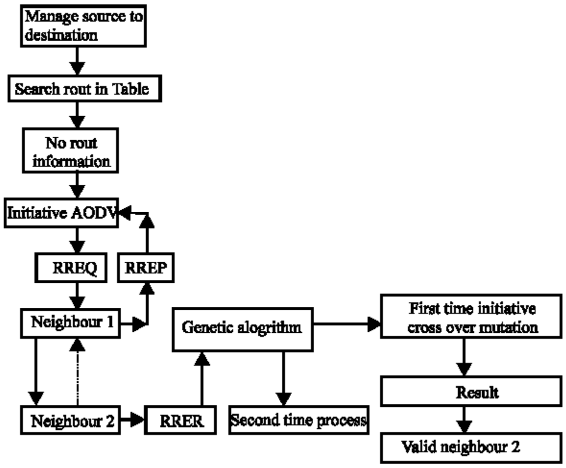

Fig. 3: System flow chart

- Reduced memory occupation
- Reducd route broadcasting and hence reduced MAC load.
- Minimized packet looping
- Quick delivery of packets to destination

Four nodes are shown in the proposed Ad Hoc Network (Fig. 1).
Each node contains two algorithms (Fig. 2)

- AODV
- GA

When the mobile node comes into activation, (Fig. 3) all the algorithms are initialized with initial conditions.

Random search algorithms: Random search algorithms have achieved increasing popularity. Random walks and random schemes that search and save the best must also be discounted because of the efficiency requirement. Random searches can be expected to do no better than enumerative schemes. The genetic algorithm is, a search procedure that uses random choice as a tool to guide a highly exploitative search through a coding of a parameter space.

The schemes mentioned and countless hybrid combinations and permutations have been used successfully in many applications. As more complex problems are attacked, other methods will be necessary. The gradient technique performs well in its narrow problem class.

The goals of optimization: Before examining the mechanics and power of a simple genetic algorithm, the goals to optimize a function or a process are decided. The optimization has two parts

- seek improvement to approach some
- optimal point.

There is a clear distinction between the process of improvement and the destination or optimum itself. In judging optimization procedures we commonly focus solely upon convergence.

Ga approach to autonomic routing: GA is a search strategy that is inspired by biological evolution[35] and which is based on the following features:

- Population is generated randomly. The population represents individual path from source to destination. In programming population is made of binary values.
- A fitness function[36] is used, which evaluates, the suitability of each chromosome for representing a path to the optimum.
- Roulette wheel concept is used for selecting individuals for reproduction based on their fitness. This represents choosing paths with better QoS.
- The genetic operators such as crossover and mutations are used. If SxNyD and SuNvD are two valid paths from source S to destination D that both use the intermediate node N and x, y, u, v are sequences of nodes on the two paths, then SuNyD and SxNvD are two paths that result from the crossover operation applied to the two previous paths.

A path w is a variable length sequence of nodes, which begins with the source node S and ends with the destination D. For any nodes a and b, ab is a subsequence in w only if there is a link (i.e. path of length 1) going from a to b. Thus w represents any viable path from S to D[37]. Each path w has a costC(w) function goal value C(w) which is the observed QoS for that path. C describes the effectiveness of the path w. A smaller value of C(w) means that w is more desirable. A simple example is when C(w) is the number of links on the path from S to D, i.e. the number of hops. Conventional shortest path routing[38] in the Internet Protocol (IP) tries to minimize the function C.

C(w) may also be the delay or loss of packets that is observed over the path w, or the variance in packet delay, or the power consumed for processing and forwarding the packet in the hops of a wireless network, or the overall security level of the path, or combinations of many of these metrics.

GA implementation[39], uses the pre-existing CPN system. New paths are generated in two different ways:

- Generate additional paths using the path crossover operation. If some new path SuNcD generated by a crossover from SuNvD and SaNcD was not already in the Stack, then it is placed there with the new goal value, assuming that the goal is additive.
- SPs discover routes and ACK packets bring back valid routes to the source, which are stored into the Stack. This is similar to mutation in GA. The paths in the Stack are organized in a list sorted in order of increasing C(w) value. Therefore the first path in the list is the fittest path.

The GA operates as follows. Every path[40] discovered by SPs and brought back by ACKs becomes an individual in the GA population. The paths with the same source S and destination D form a GA route backup called Stack. Paths in the same backup are combined to form new paths, provided they share the same intermediate node. Their fitness, is synthesized from those of their constituents. Whenever a DP needs to be forwarded to the destination, the path at the top of the Stack, is used as the source route.

The GA algorithm as a system routine runs in the background. The GA routine consists of a main loop which is in charge of polling the kernel for new paths, checking its internal data structures for size and consistency, selecting individuals for crossover and doing the actual crossover and periodically updating the CPN module's dumb packet route backup. The data exchange operations between the CPN kernel module and the GA routine is bi-directional as the module passes measurements to the routine and in return the routine periodically updates the module's route backup with the paths that, have the best fitness at that time. GA routine, in round robin mode, provides the identity of each of the paths who's QoS are better than other QoS. The round-robin policy on the best routes then acts as a simple load balancing mechanism to avoid saturation of any given path and also offers the possibility to gather measurement data concerning a much bigger set of hops. Another way of increasing the measurement data available to all connections from a given source, is to share the hops pool among several different connections emanating from the same source.

IMPLEMENTATION

Steps involved in execution of the intelligence in the Ad Hoc network

Step 1: Initialize AODV

Step 2: Initialize GA with 50 population

Step 3: Train GA to obtain best generation and subsequently get the best result

Step 6: When a link breaks between node 3 and the destination, immediately GA in node three is activated

Step 7: GA finds the node four as the next hop.

SIMULATION

Training genetic algorithm: The genetic algorithm generates chromosome population. The entire pattern/ data that represent the node numbers in the simulation network are given as input Table 1 to the genetic algorithm. New population is created during crossover and mutations. Subsequently fitness calculations are done for each chromosome. The chromosome is weighted. Figure 4 shows the minimum fitness value for each generation is plotted against generation numbers. It can be noted that the value oscillates continuously.

In Fig. 5, the maximum fitness value for each generation is shown. The value is limited to less than 3*e04. The maximum value for each generation keeps changing with some pattern. This indicates the generation is getting slow stability.

The average fitness value is preferred to select the generation number. In Fig. 6, the generation indicates that the best representation by the chromosomes is possible only at generation corresponding to maximum average fitness. The chromosome with maximum fitness in this generation is chosen to represent the optimal route.

Table 1: Node number in the ns2

Present node	Broken node	Destination node
3	9	7
2	5	6
5	7	3
8	4	6
10	2	9
5	2	3
7	2	6
6	9	5
4	9	5

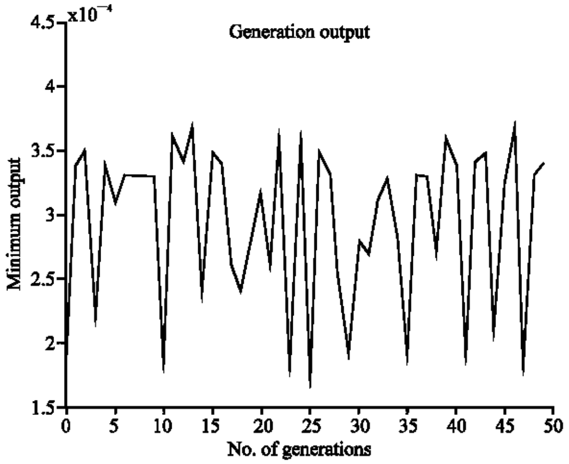

Fig. 4: Minimum fitness value for the 50 chromosomes

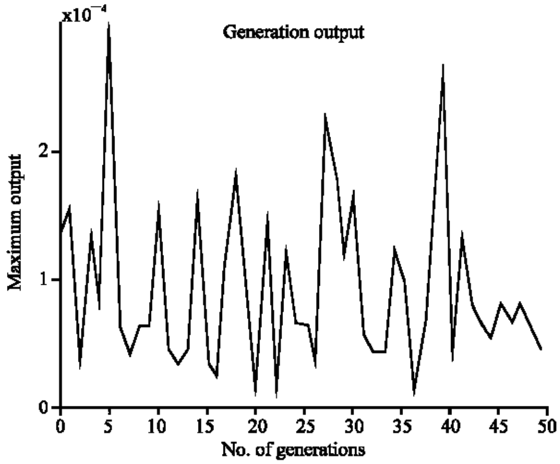

Fig. 5: Maximum fitness value for the 50 chromosomes

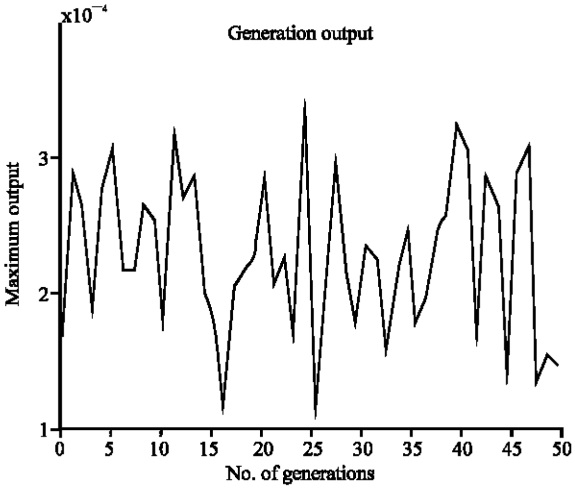

Fig. 6: Average fitness value for the 50 chromosomes

The initialization of 20 nodes under study is displayed using ns2 Fig. 7. Only few nodes are visible

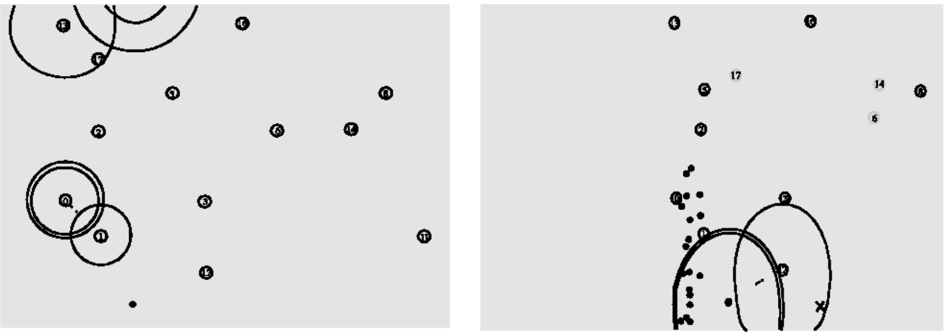

Fig. 7: Simulation of the intelligent Ad Hoc network.

Fig. 8: Performance of packet transfer between node 0 and node 5

Fig. 9: The packet transfer between node 1 and node 2

and the remaining nodes are viewed by panning the screen. Transmission between few nodes at the stipulated time

The packet loss is very minimal for the first 2.5 seconds and the loss gradually increases during subsequent instance Fig. 8. The maximum packet loss is, when the simulation is over. The amount of packet received over period of 10 seconds is 71.

The packet loss for 0.75 second is very minimal and suddenly increases to a greater value within next 1.5 seconds. The amount of packets received during the subsequent time is 13 packets (Fig. 9).

Result obtained through genetic algorithm: The following data is responsible for giving the next hop.

66 (CHROMOSOME 1-7 BITS)
72 (CHROMOSOME 8-14 BITS)
48 (CHROMOSOME 15-21 BITS)

CONCLUSION

This study is focussed to implement the concept of intelligent route rebuilding during the breakage of link. In the existing implementation, route is established based on the repeated route broadcast to the neighbours from the current node to know the destination node path. The present thesis has contributed towards developing intelligence in *ad hoc* network to achieve the following criteria

- Packet delivery fraction
- Average end-to-end delay of data packets
- Normalized routing load
- Normalized MAC load

The training of GA with the information of present node, broken node and destination node will be carried out periodically.

The information for the inputs to GA will be obtained from the minimum information stored in the present node, Information about the routes from the neighboring nodes also will be utilized for training the GA.

The GA will locate the next node keeping the already learnt routes using the best chromosome. Back propagation algorithm network will act as cache. The concept of cache is an important property of GA. Due to this property, fresh routes are learnt retaining the old weights and updating with new weights.

System will take care of forwarding packets to all the nodes corresponding to the outputs of GA. Based on the reply from neighbors, choose any one neighboring node for forwarding packet. New packets are connected to the moving mobile.

Future works: The present study has to be tested with different environment conditions for testing the quick rebuilding of the network. A combination of neural network and genetic algorithm to form neuro genetic algorithm to find out the performance of the Ad Hoc network

REFERENCES

1. Perkins, C.E., 1997. Mobile IP: Design principles and practices. Addison Wesley.
2. McQuillan, J.M., I. Richer and E. C. Rosen, 1980. The new routing algorithm for ARPANET. IEEE Transactions on Communications, 28,pp:711-719.
3. Hedrick, C., 1988. Routing information protocol. Internet request for Comments RFC 1058, pp: 1188-1208.
4. Karthikeyan Sundaresan and Raghupathy Sivakumar, Routing in Ad-hoc networks with MIMO links, Georgia Institute of Technology, School of Electrical and Computer Engineering, funded in part by the National Science Foundation, under grants ECS-0225497, ECS-0428329, ANI-0117840 and CCR-0313005.
5. Frederick Ducatelle, Gianni Di Caro and Luca Maria Gambardella, 2005. Ant agents for hybrid multipath routing in mobile Ad Hoc networks, Proceedings of the Second Annual Conference on Wireless On demand Network Systems and Services (WONS), St. Moritz, Switzerland.
6. Yuanzhu Peter Chen, Arthur L. Liestman, a zonal algorithm for clustering Ad Hoc networks, Intl. J. Foundations of Computer Sci. World Scientific Publishing Company, pp: 1-18.
7. Moy, J., 1991. OSPF version 2. Internet Request For Comments RFC.
8. Park, V.D. and M.S. Corson, 1997. A highly adaptive distributed routing algorithm for mobile wireless networks. Proc. of IEEE Infocom'97, pp: 1405-1413.
9. Nasipuri, A., R. Cartaneda and S.R. Das, 1999. On-demand multipath routing for mobile Ad Hoc networks. Proc.of IEEE Infocom 99.
10. Byers, J.W., M. Luby, M. Mitzenmacher and A. Rege, 1998. A digital fountain approach to reliable distribution of bulk data. In Proceedings of the ACM Sigcomm, pp: 56-67.
11. Maxemchuk, N.F., 1975. Dispersity Routing. In proceedings of the IEEE ICC, 41,pp: 10-41.
12. Bahk, S. and M.E. Zarki, 1992. Dynamic multi-path routing and how it compares with other dynamic routing algorithms for high speed wide area networks. In Proceedings of the ACM Sigcomm, pp: 53-64.
13. Cidon, I., R. Rom and Y. Shavitt, 1999. Analysis of multi-path routing. IEEE transactions on Networking, 7: 885-896.

14. Murthy, S. and J.J. Garcia-Luna-Aceves, 1996. Congestion-oriented shortest multipath routing. In proceedings of the IEEE Infocom, pp: 1028-1036.
15. Gelenbe, E., 2004. Cognitive packet network. U.S. Patent.
16. Marina, M.K. and S.R. Das, 2001. Performance of route caching strategies in dynamic source routing. In proceedings of the Int'l Workshop on Wireless Networks and Mobile Computing (WNMC) in conjunction with Int'l Conf. On Distributed Computing Systems (ICDCS), pp: 425-432.
17. Gelenbe, E. and R. Lent, 2004. Power-aware Ad Hoc cognitive packet networks. Ad Hoc Networks J., pp: 2.
18. Lee, S.J. and M. Gerla, 2000. AODV-BR: Backup Routing in Adhoc Networks. In proceedings of the IEEE Wireless Communications and Networking Conference(WCNC), pp: 1311-1316.
19. Johnson, D. and D. Maltz, 1996. Dynamic Source routing in adhoc wireless networks. In T. Imielinski and H. Korth, editors, Mobile computing, chapter 5. Kluwer Academic.
20. Park, V.D. and M.S. Corson, 1997. A highly adaptive distributed routing algorithm for mobile wireless networks. In Proceedings of the IEEE Infocom, pp: 1405-1413.
21. Hossam Hassanein, 2003. Audrey Zhou, Load-aware destination-controlled routing for MANETs, computer communications Elsevier, pp: 1551-1558.
22. Ni, S.Y., Y.C. Tseng, Y.S. Chen and J.P. Sheu, 1999. The broadcast storm problem in a mobile Ad Hoc network. In Proceedings of the IEEE/ACM Mobicom, pp: 151-162.
23. Holland, G. and N.H. Vaidya, 1999. Analysis of TCP performance over mobile Ad Hoc networks. In proceedings of the IEEE/ACM Mobicom, pp: 219-230.
24. Perkins, C.E., E.M. Royer and S.R. Das, 2000. Ad Hoc On Demand Distance Vector (AODV) routing, http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-07.txt. IETF Internet Draft (work in progress).
25. Hu, Y.C. and D. Johnson, 2000. Caching strategies in on-demand routing protocols for wireless Ad Hoc networks. In proceedings of the IEEE/ACM Mobicom, pp: 231-242.
26. Gafni, E. and D. Bertsekas, 1981. Distributed algorithms for generating loop-free routes in networks with frequently changing topology. IEEE Transactions on Communications, 29,pp: 11-18.
27. Sung-Ju Lee and Mario Gerla, Split Multipath Routing with maximally disjoint paths in Ad Hoc networks, Internet and Mobile Systems Labs.
28. Lee, S.J. and M. Gerla, 2001. Split Multipath Routing with maximally disjoint paths in Ad Hoc networks. In Proceedings of the IEEE ICC, pp: 3201-3205.
29. Nasipuri, A., R. Castaneda and S.R. Das, 2001. Performance of multipath routing for on-demand protocols in mobile Ad Hoc networks. ACM/Kluwer Mobile Networks and Applications (MONET), 6:339-349.
30. Wu, K. and J. Harms, 2001. Performance study of a multipath routing method for wireless mobile Ad Hoc networks. In proceedings of the IEEE Int'l Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), pp: 99-107.
31. Zaumen, W. and J.J. Garcia-Luna-Aceves, 1998. Shortest multi path routing using generalized diffusing computations. In proceedings of the IEEE Infocom, pp: 1408-1417.
32. Perkins, C.E. and E.M. Royer, 1999. Ad Hoc on-demand distance vector routing. In proceedings of the EEE workshop on mobile computing systems and Applications (WMCSA), pp: 90-100.
33. Perkins, C.E. and P. Bhagwat, 1994. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In proceedings of the ACM SIGCOMM, pp: 234-244.
34. Johnson, D. and D. Maltz, 1996. Dynamic source routing in Ad Hoc wireless networks. In T. Imielinski and H. Korth, Editors, Mobile computing, chapter 5. Kluwer Academic.
35. Holland, J.H., 1975. Adaptation in natural and artificial system. University of Michigan Press.
36. Moy, J., 1991. OSPF version 2. RFC 1247, 1991.
37. Chlamtac, I., A. Faragó and T. Zhang, 1996. Lightpath (wavelength) routing in large WDM networks," IEEE J. Select. Areas Commun., pp: 909-913.
38. Huitema, C., 1999. Routing in the Internet, 2nd Ed. Prentice hall, Upper Saddle River, N.J.
39. Michalewicz, Z., 1996. Genetic algorithms + data structures = evolution programs. 3rd rev. and extended Ed. Berlin; New York: Springer-Verlag, pp: 387.
40. Ramaswami, R. and K.N. Sivarajan, 1994. Optimal routing and wavelength assignment in all-optical networks.