



## AFAR: adaptive fuzzy ant-based routing for communication networks\*

Seyed Javad MIRABEDINI<sup>†1</sup>, Mohammad TESHNEHLAB<sup>2</sup>, M. H. SHENASA<sup>2</sup>,

Ali MOVAGHAR<sup>3</sup>, Amir Masoud RAHMANI<sup>1</sup>

<sup>1</sup>Engineering Department, Science and Research Branch, Islamic Azad University, Tehran, Iran)

<sup>2</sup>Control Department, K. N. Toosi Technology University, Tehran, Iran)

<sup>3</sup>Computer Department, Sharif University, Tehran, Iran)

<sup>†</sup>E-mail: jvd2205@yahoo.com

Received Feb. 20, 2008; revision accepted June 10, 2008; CrossCheck deposited Nov. 10, 2008

**Abstract:** We propose a novel approach called adaptive fuzzy ant-based routing (AFAR), where a group of intelligent agents (or ants) builds paths between a pair of nodes, exploring the network concurrently and exchanging obtained information to update the routing tables. Routing decisions can be made by the fuzzy logic technique based on local information about the current network state and the knowledge constructed by a previous set of behaviors of other agents. The fuzzy logic technique allows multiple constraints such as path delay and path utilization to be considered in a simple and intuitive way. Simulation tests show that AFAR outperforms OSPF, AntNet and ASR, three of the currently most important state-of-the-art algorithms, in terms of end-to-end delay, packet delivery, and packet drop ratio. AFAR is a promising alternative for routing of data in next generation networks.

**Key words:** Adaptive fuzzy routing algorithm, Swarm intelligence, Routing table, Communication network, Packet delay, Throughput

doi:10.1631/jzus.A0820118

Document code: A

CLC number: TN915.11

### INTRODUCTION

Modern communication networks are becoming extensively diverse and heterogeneous. In a decentralized network each node makes all of its own decisions. But decentralized algorithms have also oscillations and stability problems (Tannenbaum, 2003). Current routing algorithms are not adequate to tackle the increasing complexity of such networks. Ant colony boasts a number of advantages due to the use of mobile agents and stigmergy: scalability, fault tolerance, adaptation, speed, modularity, autonomy, and parallelism (di Caro and Vasilakos, 2000; Barabási and Bonabeau, 2003). But one of the biggest difficulties with ant colony in network routing area is that multiple constraints often make the routing problem

intractable. This is one area where fuzzy control may be beneficial. A full discussion concerning these algorithms can be found in (Sarif *et al.*, 2004).

In this paper, we propose a novel approach called adaptive fuzzy ant-based routing (AFAR) using a fuzzy logic technique to solve the network routing problem, which allows multiple constraints to be considered in a simple and intuitive way. Fuzzy control is a control technique based on the principles of fuzzy set theory, and the control systems are designed to mimic human control better than classical control systems by incorporating expert knowledge and experience in the control process. This is achieved by using linguistic variables in the control system to enable the designer to include control rules that naturally follow human thought. A survey on recent advances of fuzzy logic in telecommunication networks can be found in (Zhang and Phillis, 2001).

\* Project supported by the Iranian Telecommunication Research Center

## OVERVIEW OF ANT-BASED ROUTING

Development of adaptive routing algorithms for telecommunication networks is an area of active study. Ant colony optimization, in which information gathered by simple autonomous mobile agents is shared and exploited for problem solving, has been applied to routing in telecommunication networks (Dorigo and di Caro, 1999; Birattari *et al.*, 2002; Katangur *et al.*, 2004; Ducatelle *et al.*, 2006) with good results. The success achieved using such methods is in large part due to the introduction of randomness in the search procedure, permitting to escape from local minima and achieve a more globally favorable solution. Ant-based routing algorithms derive from recent understandings of basic principles underlying the operation of biological swarms, such as ants or honeybees. These swarms, often containing thousands of elements, routinely perform extraordinarily complex tasks of global optimization and resource allocation using only local information. These properties make ant colony very attractive for network routing (Sim and Sun, 2002), routing and load-balancing (Sim and Sun, 2003), routing in sensor networks (Singh *et al.*, 2008), routing in telecommunication network (Akon *et al.*, 2004), quality of service routing (Zhang and Liu, 2001), genetic ant routing (Cheng and Hou, 2003), etc.

## FUZZY LOGIC FOR AFAR

In this paper, our novel approach AFAR is constructed with the inspiration of a communication model observed in ant colonies combined with the capabilities of the fuzzy logic technique. The AFAR algorithm first determines the crisp path ratings for all eligible paths between the source and destination nodes from the viewpoint of fuzzy inference. The path with the highest rating is then chosen to route the traffic flow. The path congestion rate in this paper represents the degree of the path usability in the sense of the multiple criteria required. Whenever traffic flow is routed to a chosen path, a packet is dropped when it arrives at a full buffer. The fuzzy inputs are chosen as the ‘path delay’ and the ‘path utilization’. The path delay expresses the total delay of an agent to reach to destination from its source.

The path utilization determines the amount of a path usage. The fuzzy output is the rate of path congestion. The fuzzy rule base is shown in Table 1.

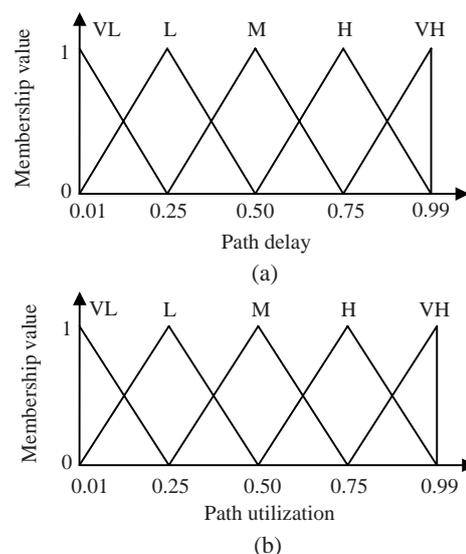
**Table 1 Fuzzy rule base for AFAR algorithm**

Delay	Congestion rate				
	VL	L	M	H	VH
VL	LL	LM	LH	ML	MM
L	LM	LH	ML	MM	MH
M	LH	ML	MM	MH	HL
H	ML	MM	MH	HL	HM
VH	MM	MH	HL	HM	HH

VL: very low; L: low; M: medium; H: high; VH: very high

There are 25 rules defined for this fuzzy system. For example, two of the rules are as follows: R1: If delay is VL and utilization is VL then Congest rate is LL; R25: If delay is VH and utilization is VH then Congest rate is HH.

The membership functions for the fuzzy input variables path delay  $\bar{d}$ , path utilization  $\bar{u}$ , are shown in Figs.1a and 1b, respectively. The universes of discourse for the fuzzy variables are all normalized between (0, 1). The membership functions for the fuzzy sets of inputs are chosen to be triangular because this type of membership function has good features such as easiness in computation, clarity, and noise tolerance. Both input variables  $\bar{d}$  and  $\bar{u}$  have five membership functions titled as VL, L, M, H, and VH, which stand for “very low”, “low”, “medium”, “high”, and “very high”, respectively.



**Fig.1 Membership functions for path delay (a) and for path utilization (b)**

There is also a fuzzy set for output variable as shown in Fig.2. All of the membership functions for the fuzzy sets of output are chosen to be triangular for its easiness in computation, clarity, and noise tolerance.

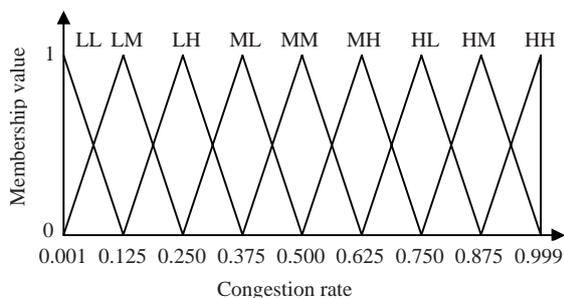


Fig.2 Membership functions for output (congestion rate)

The output variable  $\bar{Y}$  has nine membership functions titled as LL, LM, LH, ML, MM, MH, HL, HM, HH to indicate “low low”, “low medium”, “low high”, “medium low”, “medium medium”, “medium high”, “high low”, “high medium”, and “high high”, respectively. The fuzzy operator used for the AND method in “if-then rules” such as “If A is a AND B is b then C is c” is “multiplication”. The defuzzification is the process of conversion of fuzzy output set into a single number (Mirabedini and Teshnehlab, 2004; 2007a; 2007b; Mirabedini et al., 2007). In our approach, the delay of each route ( $d_i$ ) and the number of packets in each buffer ( $u_i$ ) are stored in a stack of forward agent. After reaching to the destination the backward agent inherits this stack and uses it to update the routing tables of all visited nodes. We describe the details of this mechanism in the next section.

**AFAR algorithm**

To sum up, the adaptive fuzzy ant-based routing (AFAR) algorithm is outlined as follows: In AFAR, routing is determined through complex interactions of network exploration agents, called ants. These agents are divided into two classes, the forward ants and the backward ants. The idea behind this subdivision of agents is to allow the backward ants to utilize the useful information gathered by the forward ants on their trip from source to destination.

The entries of the routing table are probabilities, and as such, they sum to one for each row of the network. These probabilities serve a dual purpose.

The exploration agents of the network, the ants, use them to randomly decide the next hop to a destination. However the actual network traffic uses them deterministically, choosing as the next hop the route with the highest probability (Dorigo and di Caro, 1999). The sequence of routing actions is simple and intuitive. Fig.3 provides a graphical explanation of the algorithm described below.

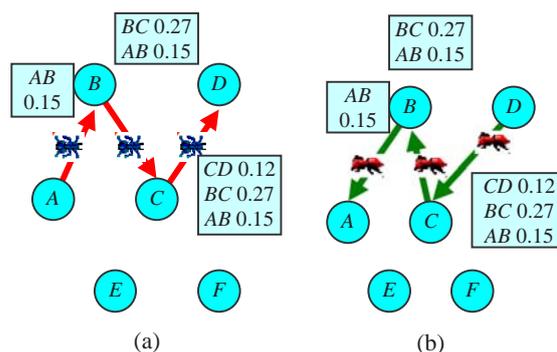


Fig.3 (a) Forward ant movement; (b) Backward ant movement

In order to apply our methodology for networks routing, we replaced the routing table of each network node by table of pheromones, which we call pheromone table. The entries of this table are the pheromones represented by probabilities  $\tau_{ij}$  ( $i=1, 2, \dots, L; j=1, 2, \dots, N$ ), where  $L$  is the number of neighbors,  $N$  is the number of nodes in the underlying network, and  $\tau_{ij}$  represents the amount of goodness from neighbor  $i$  to destination  $j$  via current node.

These pheromones are used by ants to allow them to randomly explore the network and possibly find new and better routes. Then once the routes are discovered, the next-hop pheromones are updated to reflect the new discoveries. All routing table entries conform to the constraint represented as

$$\sum_{n \in N_k} \tau_{dn} = 1, d \in [1, N], N_k = \{\text{neighbor of } k\}, \quad (1)$$

where  $\tau_{dn}$  expresses the goodness (desirability), under the current network-wide routing policy, of choosing  $n$  as next node when the destination node is  $d$ . Throughout this paper we consider that the topology of the network is modeled by a directed weighted graph  $G = \{N, A\}$  with a node set  $N$  and an arc set  $A$ . At each node  $n$  of  $N$ , there are two

parameters: an array  $D_n$  stores the last estimated delay  $D_{j,d}^n(t)$  and an array  $U_n$  stores the last estimated utilization  $U_{j,d}^n(t)$  experienced by the forward ants from node  $n$  to each destination node  $d$  via neighboring node  $j$ . Table of pheromones and the arrays  $D_n$  and  $U_n$  can be seen as memories local to nodes capturing different aspects of the network dynamics. The arrays  $D_n$  and  $U_n$  maintain absolute time and utilization estimates to all the nodes, while the table of pheromones gives relative probabilistic goodness rates for each source-destination pair under the current routing policy implemented over all the network.

In the AFAR algorithm, we defined two kinds of ants: forward ant and backward ant, which are referred to as  $Ant_F$  and  $Ant_B$ , respectively. The  $Ant_F$  is generated by the source node  $s$  in some interval times and its mission is to move from node  $s$  to a determined destination  $d$ . After reaching to the destination  $d$ , it generates the  $Ant_B$  and dies. The  $Ant_B$  returns to destination  $s$  following the same route as taken by  $Ant_F$ , with the aim of utilizing the information gathered by  $Ant_F$ , as well as updates routing tables of the visited nodes. In specific intervals, every network node  $n$ , launches a forward ant  $Ant_F$ , with a randomly chosen destination node  $d$ . Each  $Ant_F$ , chooses the next neighbor node using the information saved in the routing table. The  $Ant_F$ , when traveling to its destination, stores track of its trip from node  $n$  to node  $d$  as well as the time elapsed since its launching time. Whenever a cycle is found, that is, if the ant is had to come back to a previously traversed node, all of the information about the cycle is removed. When reaching to the destination node  $d$ , the  $Ant_F$  produces an  $Ant_B$ , as well as transmits to it all its memory. The  $Ant_B$  uses the same path as that of its relevant  $Ant_F$ , but in the reverse direction. The  $Ant_B$  uses the higher priority queue, because its function is to quickly update the routing tables by the information obtained by the  $Ant_F$ . To do this, the values of both path delay  $\bar{d}$  and path utilization  $\bar{u}$ , for each eligible path are computed by the following steps:

(1) The total delay  $D$  which is defined as the sum of all delays  $d_i$  of intermediate link  $i$  traversed by  $Ant_F$  is taken and used to make a weighting measure  $\lambda_{d_i}$  for each route  $i$  as

$$D = \sum_{i=1}^s d_i, \tag{2}$$

$$\lambda_{d_i} = d_i / D, \quad i=1, 2, \dots, s, \tag{3}$$

where  $s$  is the total number of links traversed by  $Ant_F$ .

(2) The estimated path delay  $\bar{d}$  is calculated by multiplying the delay  $d_i$  of each intermediate link  $i$  by its corresponding weight factor  $\lambda_{d_i}$ , as shown here:

$$\bar{d} = \sum_{i=1}^s \lambda_{d_i} d_i. \tag{4}$$

(3) The utilization  $u_i$  of each buffer on the path is calculated as

$$u_i = q_i / Q, \quad i=1, 2, \dots, s, \tag{5}$$

where  $q_i$  is the number of packets in buffer and  $Q$  is a maximum capacity of buffer for each node.

(4) The sum of these utilization measures is computed and used to make a weighting measure  $\lambda_{u_i}$  for each buffer  $i$  as

$$U = \sum_{i=1}^s u_i, \tag{6}$$

$$\lambda_{u_i} = u_i / U, \quad i=1, 2, \dots, s. \tag{7}$$

(5) Finally, the estimated path utilization  $\bar{u}$  is calculated by the multiplication of the utilization  $u_i$  and its corresponding weight factor  $\lambda_{u_i}$ ,

$$\bar{u} = \sum_{i=1}^s \lambda_{u_i} u_i. \tag{8}$$

When  $Ant_B$  reaches the current node  $n$  from node  $d$  via neighboring node  $j$ , it updates the estimated delay of  $D_{j,d}^n(t)$  and the estimated utilization of  $U_{j,d}^n(t)$  according to

$$D_{j,d}^n(t) = (1-\eta) D_{j,d}^n(t) + \eta \bar{d}_{j,d}^n, \tag{9}$$

$$U_{j,d}^n(t) = (1-\eta) U_{j,d}^n(t) + \eta \bar{u}_{j,d}^n, \tag{10}$$

where  $\eta \in (0, 1]$  is the damping parameter,  $\bar{d}_{j,d}^n$  is the estimated path delay between source node  $n$  to destination  $d$  via neighboring node  $j$ ,  $\bar{u}_{j,d}^n$  is the estimated path utilization between source node  $n$  to destination  $d$  via neighboring node  $j$ .

Using a small value of the damping parameter ( $\eta=0.15$ ) makes smoothed trajectories, which represents transient features more closely resembling those given by the simulation model. Calculating the pair values of  $D_{j,d}^n(t)$  and  $U_{j,d}^n(t)$  as crisp inputs, we determine the congestion rate  $r$  for each eligible path via fuzzification [based on the membership functions represented in Figs.1~2], fuzzy inference [based on the rule base shown in Table 1 and the ‘‘Mamdani’’ implication] and defuzzification [based on the ‘‘centers mean’’ method]. The estimation  $r$  means the amount of congestion rate to go from current node  $n$  to destination node  $d$  via neighboring node  $j$ , expressed as

$$r = \sum_{l=1}^M \bar{Y}^l (\mu_{A(d)}^l, \mu_{A(u)}^l) / \sum_{l=1}^M (\mu_{A(d)}^l, \mu_{A(u)}^l), \quad (11)$$

where  $i$  is the node that an ant is going from,  $j$  is the node where an ant wants to move,  $M$  is the number of fuzzy rule bases used ( $M=25$ ),  $\bar{Y}^l$  is the mean value of each membership function in the fuzzy set,  $\mu_{A(d)}^l$  is the amount of membership functions for delay, and  $\mu_{A(u)}^l$  is the amount of membership functions for path utilization.

Then, the output of the fuzzy system called rate  $r$ , is applied to the FARA algorithm, which can be used as a criterion for updating routing tables of each visited node. A  $\tau_{df}$  pheromone associated with node  $f$ , when it wants to update the data corresponding to node  $d$ , is increased according to

$$\tau_{df} = \tau_{df} + (1-r)(1-\tau_{df}). \quad (12)$$

The other neighboring nodes ( $j \neq f$ )  $\tau_{dj}$  pheromones associated with node  $k$  are diminished to satisfy Eq.(1), through

$$\tau_{dj} = \tau_{dj} - (1-r)\tau_{dj} \quad \forall j \in N_k, j \neq f. \quad (13)$$

### Improvements proposed by AFAR

One possible problem with this AFAR algorithm is that the distribution of probabilities eventually ‘‘would freeze’’ with a probability value, near to one, while the other values remain insignificant (Barán and Sosa, 2001). To avoid this problem we defined in our simulation system a noise factor of  $\gamma$ ,  $\gamma \in (0,1)$ , so that at every time slice an ant has the probability  $\gamma$  of choosing a purely random path, and probability  $1-\gamma$  of choosing its path according to the pheromone tables on each node. In our experiments we set  $\gamma$  to be 0.05. Furthermore, for having a better initialization of the pheromone table for each node, a greater pheromone value is assigned to the neighbor node  $j$  when it becomes destination  $d$ . Then the pheromone values are initialized in the pheromone table by

$$\tau_{j,d}^n = \begin{cases} \frac{1}{L_n} + \frac{3(L_n - 1)}{4L_n^2}, & j = d, \\ \frac{1}{L_n} - \frac{3}{4L_n^2}, & j \neq d, \end{cases} \quad (14)$$

where  $d \in Neighbor(n)$ .

If the destination  $d$  is not a neighboring node, then a uniform distribution is initially assumed as

$$\tau_{j,d}^n = 1/L_n, \quad d \notin Neighbor(n). \quad (15)$$

Hence, it should be noticed that Eqs.(14) and (15) fulfill the constraint of Eq.(1).

### COMMUNICATION NETWORK MODEL

We accomplished our experiments on communication networks with irregular topology lacking mechanisms for congestion and admission control. This choice has some characteristics. First, we desire to examine the behaviors of our algorithm and of its competitors in states which minimize the number of interacting components. In fact, any congestion or admission control algorithm has an important effect on the network performance and it is difficult to evaluate the efficiency of the control algorithm by itself and of its interactions with the routing algorithm.

## Simulation

We used network simulator 2 (ns-2) as a standard simulation package and extended it to implement our adaptive fuzzy ant-based algorithm with OSPF and two state-of-the-art ant-based algorithms called AntNet (Dorigo and di Caro, 1998) and ASR (Lü *et al.*, 2004). The aim of the simulator is to closely mirror the essential features of the concurrent and distributed behavior of a generic communication network without sacrificing efficiency and flexibility in code development. Two network models used for performance evaluation of these routing algorithms are as follows:

(1) The NSFNET with 14 nodes. It is assumed that the bandwidths of the links are all 1.5 Mb/s and the link delays are in milliseconds.

(2) The NTTNET network composed of 55 nodes. It is assumed that the bandwidths of the links are all 6 Mb/s and the link delays are in milliseconds.

## Traffic patterns

In our experiments, the packet size for both data and routing packets were the same and fixed at 512 bytes. Each packet was emitted from each node to a random destination following Poisson traffic distribution. At each simulation time, incoming packets were pushed to the buffer corresponding to the chosen outgoing link. Packets were queued and there was no congestion control other than the routing algorithm itself. Each node removed the packet in front of its queue, examined the destination of this packet and used its routing table to send the packet to one of its neighboring nodes.

In our simulation the following abbreviations will be used for the routing algorithms: OSPF=open shortest path first, AntNet=the original version of ant routing algorithm, ASR=adaptive swarm-based routing algorithm, and our novel approach AFAR=adaptive fuzzy ant-based routing algorithm.

During the experiments, all of the network situations were considered the same for the routing algorithms. The strategy considered during the simulation for better evaluation of algorithms was to change some link bandwidths between nodes in both NSFNET and NTTNET network topologies.

## RESULTS AND DISCUSSION

The metrics used for performance evaluations of proposed method (AFAR) with the competing algorithms are as follows:

(1) End-to-end delay: delay incurred by a packet being transmitted between a source and destination node.

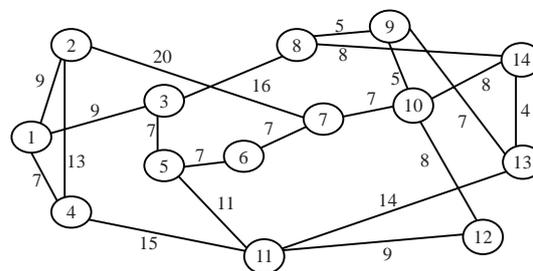
(2) Throughput: fraction of packets sent by a source node that arrives at the destination node.

(3) Dropped packets: the data packets that are dropped during the routing process because the buffer of the node is full or the life time of a packet is expired.

(4) Overhead: the number of packets that is used to maintain or control the network. Note that the number of control packets in ant-based routing algorithms is not much more than that in the conventional routing methods (i.e., OSPF), because updating the routing tables is done by ants in interval times and there is no obligation to have global updating mechanisms such as the flooding used in OSPF.

## Results with NSFNET network topology

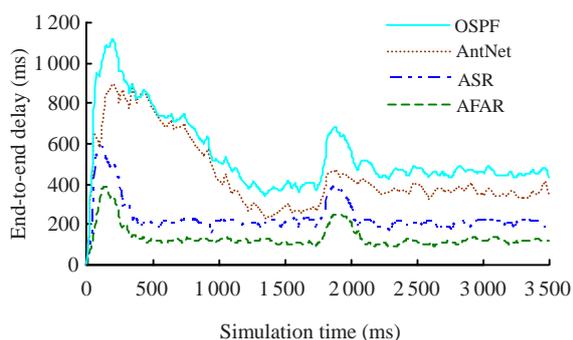
Simulation results for NSFNET (Fig.4) in state of link failures are discussed as follows. At the beginning of the simulation, as shown in Fig.5, it was observed that the AFAR algorithm learned quicker than ASR, AntNet and OSPF algorithms so that at around 400 ms, the AFAR could adjust routing tables and reached the stable state with an average delay around 120 ms, whilst the ASR and AntNet attained to stability with average delays around 200 ms and 280 ms, respectively. When the link changes



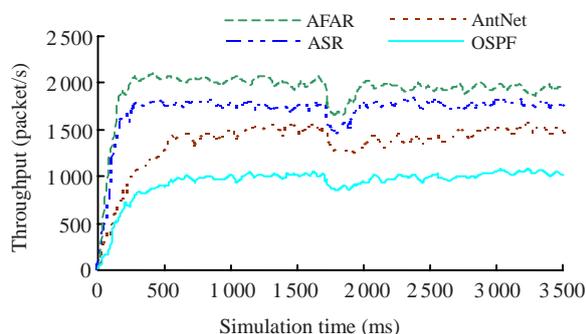
**Fig.4** NSFNET network topology. The bandwidths of all links are 1.5 Mb/s, and the links delays are in ms

occurred [the new bandwidths 0.5 Mb/s, 2.5 Mb/s are set to links (8, 9), (8, 14), respectively] in 1800 ms through the rest of the simulation, the delay of transmitted packets suddenly increased and a new load balance had to be explored (Fig.5).

During the state of changing links, the AFAR algorithm was not severely affected, demonstrating their robustness with relation to ASR and AntNet at the instant of the failure (Fig.6).



**Fig.5 Instantaneous packet delay in the simulation. NSFNET links (8, 9), (8, 14) fail from 1800 ms to 3500 ms**



**Fig.6 Instantaneous throughput in the simulation. NSFNET links (8, 9), (8, 14) fail from 1800 ms to 3500 ms**

With a glance at Fig.6, at the beginning of the simulation it is obvious that the AFAR algorithm learned the environment quicker and transmitted more packets than ASR and AntNet algorithms. In particular, it is observed in Fig.6 that the AFAR algorithm has the best performance in conveying the packets from their sources to their destinations due to its capability to update routing tables by the intelligent adaptive fuzzy method.

In addition to the above experiment for NSFNET, we also performed 15 runs for this network topology. Poisson traffic distribution was used with three different traffic loads: low, medium and

high. The total time in each simulation run was considered 3500 ms. Each run used the same traffic patterns for each of the three routing approaches with no failure during the simulation. The framework outputs event tagged to simple text files. These files were next analyzed after importing to a spreadsheet. The results of the final analysis were illustrated in Table 2. The values provided were mean values across the 15 simulations. In these experiments for the NSFNET network, AFAR offers the best mean throughput, the shortest mean packet delay, and the lowest packet loss ratio against the other approaches.

**Table 2 Experimental results averaged on 15 simulation runs in NSFNET network topology**

Standard criteria	OSPF	AntNet	ASR	AFAR
Average end-to-end delay (ms)	703.04	449.06	231.33	136.70
Average throughput (packet/s)	942.26	1349.62	1701.50	1929.20
Packet drop ratio (%)	55.13	35.73	18.97	8.14
Overhead (%)	6.39	7.51	7.51	7.51

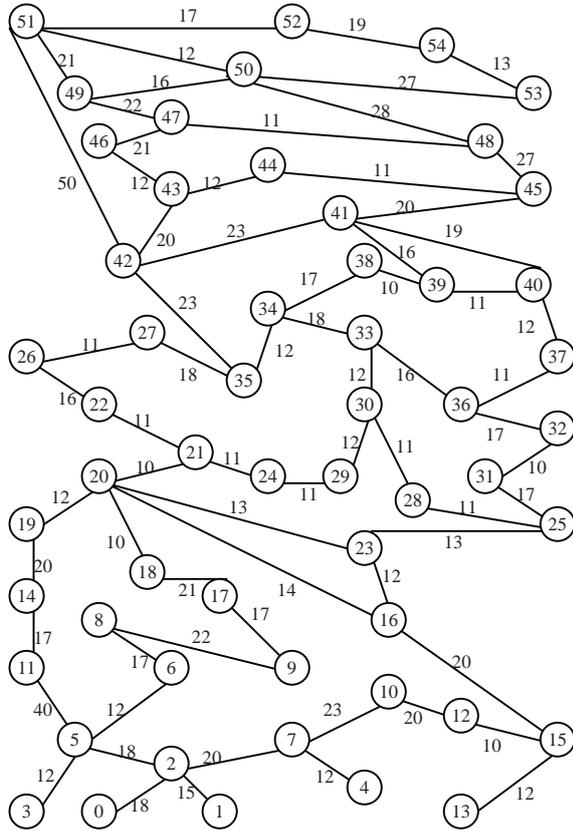
### Results with NTTNET network topology

Experimental results for a transient link failure in NTTNET (Fig.7) were discussed as following: The maximum transient delay of packet transmission for AFAR in the beginning of simulation was about 1800 ms, but these values for ASR and AntNet are close to 2400 ms and 3100 ms, respectively (Fig.8). From 4500 ms to 6700 ms of simulation, the link (33, 34) failed and its bandwidth has been changed suddenly from 6 Mb/s to 1.5 Mb/s. In term of packet delay, all algorithms were proportionally affected during the failure. But AFAR is not severely influenced, and it overcomes ASR and AntNet.

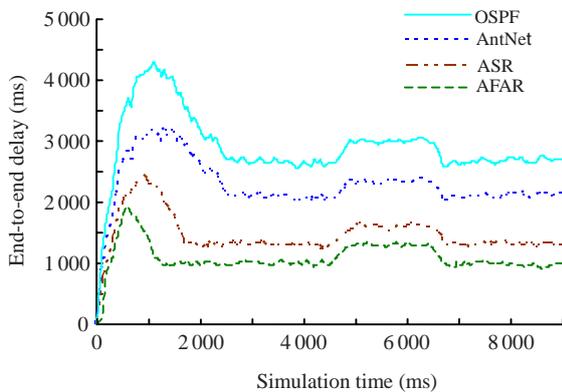
In particular, it is observed in Fig.9 that AFAR had the best performance in conveying the packets from their sources to their destinations in less end-to-end delay as well as more throughput due to its smart adaptive fuzzy control mechanism.

Besides the above experiment for NTTNET, we also accomplished 15 simulation runs for this network topology. Poisson traffic distribution was used with three different traffic loads: low, medium and high. The total time in each simulation run was considered 9000 ms. Each run used the same traffic patterns for each of the competing routing algorithms

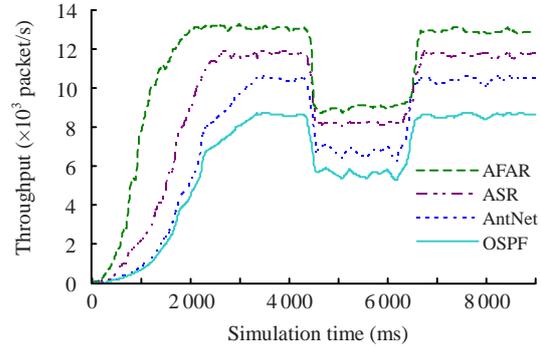
with no failure during the simulation execution. The output results were saved in simple text files. These files were next analyzed after importing to a spreadsheet.



**Fig.7** NTTNET network topology comprising 55 nodes. The bandwidths of the links are all 6 Mb/s, and the links delays are in ms



**Fig.8** Instantaneous packet delay in the simulation. NTTNET link (33, 34) fails during 4500 ms to 6700 ms



**Fig.9** Instantaneous throughput in the simulation. NTTNET link (33, 34) fails during 4500 ms to 6700 ms

The results of the final analysis were illustrated in Table 3. The values provided were mean values across the 15 simulations. In these experiments for NTTNET network, as shown in Table 3, the fact that a fewer percentage of packets are lost under the AFAR, means that not as many packets are approaching full buffers under this scheme. This illustrates the ability of AFAR to handle an increased traffic load better than the other three algorithms.

**Table 3** Experimental results averaged on 15 simulation runs in NTTNET network topology

Standard criteria	OSPF	AntNet	ASR	AFAR
Average end-to-end delay (ms)	291.54	227.93	145.90	109.86
Average throughput (packet/s)	609.42	731.76	886.04	1068.69
Packet drop ratio (%)	55.97	29.06	14.89	5.06
Overhead (%)	7.10	9.25	9.25	9.25

After the analysis of simulation results for all scenarios for the two tested networks, the following general conclusions can be inferred:

(1) In all our experiments, AFAR had a shorter transient state, a better throughput and a shorter packet delay than ASR and AntNet, indicating the effective role of the applied fuzzy technique in the suggested agent-based routing algorithm.

(2) AFAR is more robust than ASR and AntNet, in the case of link failures, because its instantaneous throughput was better than those of its competitors.

Overall, the AFAR algorithm outperforms its competitors with regard to all of the measures

collected. The results of the above experiments express the capability of AFAR as a routing mechanism to direct packets to their destinations more efficiently in case of network events such as link failures. Consequently, AFAR balances the traffic loads among different routes causing better utilization of network resources as well as traffic congestion avoidance.

## CONCLUSION

We proposed a novel routing algorithm, AFAR, based on ant colony and enhanced by fuzzy logic for network routing. Then, we developed a framework for simulation of all competing algorithms (AFAR, ASR, AntNet, and OSPF) to run on two experimental networks. The results shown in the graphs and tables indicate that the AFAR algorithm does a better job at dispersing traffic in a more uniform manner throughout the network. It also handles an increased traffic load as well as decreased transmission delay by utilizing network resources more efficiently. The advantages of such an intelligent algorithm include increased flexibility in the constraints that can be considered together in making the routing decision efficiently and likewise the simplicity in taking into account multiple constraints. In the near future the next generation networks will have capabilities including soft-switches, which allow such an intelligent agent-based routing algorithm to update the routing tables autonomously, and then they can be substituted with the conventional routing algorithms such as OSPF. Our next direction of this research is to develop adaptive fuzzy ant-based methodology in the complex world of communication networks including power-aware routing optimization as well as mobile ad hoc networks.

## ACKNOWLEDGEMENT

The authors would like to express their cordial thanks to the research group of K. N. Toosi Technology University for their valuable collaborations.

## References

- Akon, M.M., Goswami, D., Jyoti, S.A., 2004. Routing in Telecommunication Network with Controlled Ant Population. Proc. 1st IEEE Consumer Communications and Networking Conf., p.665-667.
- Barabási, A., Bonabeau, E., 2003. Scale-free networks. *Sci. Amer.*, **288**(5):50-59.
- Barán, B., Sosa, R., 2001. AntNet routing algorithm for data networks based on mobile agents. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, **12**:75-84.
- Birattari, M., di Caro, G.A., Dorigo, M., 2002. Toward the formal foundation of ant programming. *LNCS*, **2463**:188-201.
- Cheng, X., Hou, Y.B., 2003. A Study of Genetic Ant Routing Algorithm. Proc. Int. Conf. on Machine Learning and Cybernetics, p.2041-2045.
- di Caro, G.A., Vasilakos, T., 2000. Ant-SELA: Ant-agents and Stochastic Automata Learn Adaptive Routing Tables for QoS Routing in ATM Networks. Ant Colonies to Artificial Ants: Second Int. Workshop on Ant Colony Optimization, Brussels, Belgium, p.101-104.
- Dorigo, M., di Caro, G., 1998. AntNet: distributed stigmergetic control for communications networks. *J. Artif. Intell. Res.*, **9**:317-365.
- Dorigo, M., di Caro, G., 1999. Ant Colony Optimization: A New Meta-heuristic. Proc. Congress on Evolutionary Computation, p.1470-1477.
- Ducatelle, F., di Caro, G., Gambardella, L.M., 2006. An analysis of the different components of the AntHocNet routing algorithm. *LNCS*, **4150**:37-48.
- Katangur, A.K., Akkaladevi, S., Yi, P., Fraser, M.D., 2004. Applying Ant Colony Optimization to Routing in Optical Multistage Interconnection Networks with Limited Crosstalk. Proc. 18th Int. Parallel and Distributed Processing Symp., p.163-170. [doi:10.1109/IPDPS.2004.1303156]
- Lü, Y., Zhao, G.Z., Su, F.J., Li, X.R., 2004. Adaptive swarm-based routing in communication networks. *J. Zhejiang Univ. Sci.*, **5**(7):867-872. [doi:10.1631/jzus.2004.0867]
- Mirabedini, S.J., Teshnehlab, M., 2004. AntNeuroFuzzy: Optimal Solution for Traveling Salesman Problem Using Ant Colony and Neuro-fuzzy Systems. Proc. ICTIT Int. Conf., p.305-312.
- Mirabedini, S.J., Teshnehlab, M., 2007a. Performance evaluation of fuzzy ant based routing method in connectionless networks. *LNCS*, **4488**:960-965.
- Mirabedini, S.J., Teshnehlab, M., 2007b. FuzzyAntNet: a novel multi-agent routing algorithm for communications networks. *GESJ: Comput. Sci. Telecommun.*, **12**(1): 45-49.

- Mirabedini, S.J., Teshnehlab, M., Rahmani, A.M., 2007. FLAR: An Adaptive Fuzzy Routing Algorithm for Communications Networks Using Mobile Ants. Proc. Int. Conf. on Convergence Information Technology, p.1308-1315. [doi:10.1109/ICCIT.2007.26]
- Sarif, B.A.B., Abd-El-Barr, M., Sait, S.M., Al-Saiari, U., 2004. Fuzzified Ant Colony Optimization Algorithm for Efficient Combinatorial Circuit Synthesis. Proc. IEEE Congress on Evolutionary Computation, p.1317-1324.
- Sim, K.M., Sun, W.H., 2002. Multiple Ant-Colony Optimization for Network Routing. Proc. 1st Int. Symp. on Cyber Worlds, p.277-281. [doi:10.1109/CW.2002.1180890]
- Sim, K.M., Sun, W.H., 2003. Ant colony optimization for routing and load-balancing: survey and new directions. *IEEE Trans. on Syst. Man Cybern., Part A*, **33**(5):560-572. [doi:10.1109/TSMCA.2003.817391]
- Singh, G., Das, S., Gosavi, S., Pujar, S., 2008. Ant Colony Algorithms for Steiner Trees: An Application to Routing in Sensor Networks. In: Sugumaran, V. (Ed.), Intelligent Information Technologies: Concepts, Methodologies, Tools and Applications. Idea Group Publishing, USA, p.1551-1575.
- Tannenbaum, A.S., 2003. Computer Networks (4th Ed.). Prentice Hall, New Jersey.
- Zhang, R.T., Phillis, Y., 2001. Admission control and scheduling in simple series parallel networks using fuzzy logic. *IEEE Trans. on Fuzzy Syst.*, **9**(2):307-314. [doi:10.1109/91.919251]
- Zhang, S., Liu, Z., 2001. A QoS Routing Algorithm Based on Ant Algorithm. Proc. IEEE Int. Conf. on Communications, **5**:1581-1585.