Journal of Network and Computer Applications 39 (2014) 140–151

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

ANFIS and agent based bandwidth and delay aware anycast routing in mobile ad hoc networks

V.R. Budyal [a,*], S.S. Manvi [b]

[a] Department of Electronics and Communication Engineering, Basaveshwar Engineering College, Bagalkot 587102, India
[b] Department of Electronics and Communication Engineering, Reva Institute of Technology and Management, Bangalore 560064, India

ARTICLE INFO

Article history:
Received 6 October 2012
Received in revised form
20 March 2013
Accepted 5 June 2013
Available online 19 June 2013

Keywords:
Anycast routing
MANETs
Adaptive neuro-fuzzy inference system
Quality of service
Software agents

ABSTRACT

Anycast is a point to point flow of packets for obtaining services or sending data to one of a multitude of destinations that share one address. To meet needs of real time and multimedia applications, anycast routing in Mobile Ad hoc Networks (MANETs) must provide faster service with better Quality of Service (QoS). This paper proposes an Adaptive Neuro-Fuzzy Inference System (ANFIS) based multiple QoS constrained anycast routing in MANETs by using a set of static and mobile agents. Three types of agents are used in the scheme: static anycast manager agent, static optimization agent, and mobile anycast route creation agent. The scheme operates in the following steps. (1) Optimization agent at the client optimizes membership functions for bandwidth, link delay and packet loss rate to develop Fuzzy Inference System (FIS) by using ANFIS. (2) Anycast route creation agents are employed by the client to explore multiple paths from source (client) to all anycast members (servers) through intermediate nodes. These agents gather intermediate node's information such as available bandwidth, link delay, residual battery power, and stability of anycast servers. The information is passed on to the client. (3) Anycast manager agent at the client performs finding QoS factor by using optimized FIS for every path, and selects QoS anycast path based on QoS and server stability factor, and (4) Anycast route creation agent is also employed for maintaining the QoS path in the event of node/link failures. The simulation results demonstrate reduction in end-to-end delay and control overhead, improvement in packet delivery ratio and path success ratio, as compared to shortcut tree based anycast routing (SATR) in MANETs.

1. Introduction

Mobile ad hoc network (MANET) consists of hand held mobile nodes like personal digital assistants (PDA), laptop computers, cell phones, etc. They communicate through single-hop or multihop paths in a peer-to-peer fashion by using wireless media. The nodes of the MANET operate as end hosts as well as routers (Azzedine et al., 2011; Sunil and Ashwani, 2010). Due to mobility of the nodes, routing path is affected by addition and deletion of nodes. Hence, the topology of the network may change rapidly and unexpectedly. Many different protocols have been proposed to solve the routing problems in ad hoc networks which are roughly classified as unicast, multicast, broadcast and anycast. In unicast (one-to-one), packet is delivered to a particular destination (Hue et al., 2010). In multicast (one-to-many), packet is transmitted to all members of particular group (Rajashekhar and Sunilkumar, 2012). In broadcast (one-to-all), packet is sent to all network hosts. Anycast (one-to-one-of-many) allows source (client) to choose single destination (server) from a set of destination nodes (Dow et al., 2006).

The set of destinations is identified by unique anycast address and provide the same services. Searching for services on networks, often depends on the broadcast or multicast mechanism to acquire the information, which usually results in large overhead. It will be a serious problem in ad-hoc wireless networks, where the bandwidth is limited and each node moves arbitrarily. Anycasting scheme in ad hoc wireless networks can simplify access management in distributed service, improve the robustness and performance of an ad hoc network when mobility and link disconnections are frequent, and reduces the communication overhead. The source node does not need to know about picking a single server and is determined by routing scheme (Shi et al., 2010; Wu-Hsiao et al., 2007).

The server in anycast routing may be chosen by minimum hops, delay or other metrics. Anycasting along the minimum hops path may result in inefficient use of network resources, because it forwards packets along already congested shortest path, and also may not satisfy the Quality of Service (QoS) requirements for multimedia and real time application services. Hence, main objective of

* Corresponding author. Tel.: +91 8105148994.
E-mail addresses: vijayasri_rb@rediffmail.com, vrbudyal@yahoo.co.in (V.R. Budyal), sunil.manvi@revainstitution.org (S.S. Manvi).

1084-8045/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2013.06.003

end-to-end QoS based anycast routing is to find the path that satisfy QoS constraint from client to any one server. To support multimedia applications such as video conference, disaster relief, etc., in MANETs multi-constraint QoS need to be satisfied (such as a bandwidth, delay, delay jitter, packet loss rate and cost) (Jian et al., 2010; Taoshen and Zhihui, 2009). Very few works are done in QoS anycast routing in MANETs as per our literature survey.

The multi-constrained QoS routing is NP-hard and heuristic algorithms are proposed to find solution for the problem . But these algorithms are too complex and cannot obtain best global solution. QoS may be more accurately determined by using fuzzy logic instead of static values. Fuzzy Inference System (FIS) accepts more number of uncertain and imprecise data as inputs and thereby achieves flexibility, robustness, and low cost solution (Mie, 2010; El-Hajj et al., 2009). But, FIS uses human-determined membership functions (MFs) that are fixed. Therefore, they are rarely optimal in terms of reproducing the desired outputs. Tuning membership functions of parameters is a time consuming task. Neural networks overcome most of the complex problems to adapt dynamically to the system operating conditions, and to make correct decisions, if the signals are uncertain. But the integration of neural network into the fuzzy logic system makes it possible to learn from prior obtained data sets (Abraham, 2005).

This paper proposes an approach which integrates both neural and fuzzy techniques to select a server from a number of group members belonging to anycast group by considering QoS constraint route and server with higher stability in MANETs. This section presents some of the related works, software agent concept and our contributions.

1.1. Related works

Some of the related works on anycast routing in MANETs, IPv6, and wireless sensor networks are as follows. Anycast routing is an important mechanism for service discovery and load balancing (Ge and Li, 2011; Ulas and Leandros, 2004; Fernanda and Peter, 2010) in wired and wireless networks.

Density based anycast routing (DBA) considers number of available anycast group members information for routing decision is presented in Vincent et al. (2008). Anycast packets are routed along the steepest gradient at each node. The steepest gradient at each node is determined by evaluating the potential values of their direct neighbours. Packets are forwarded to the neighbour with highest potential value until there are no neighbors left with a higher potential value than its own to guarantees that the steepest gradient is ascending.

k-anycast (KA) discussed in Bing and Jie (2010), deliver a packet to any threshold *k* members of a set of servers instead of one among the nearest group servers. *k*-anycast members are selected from a set of servers by three different schemes. Three schemes explain about how to select *k* servers. The first scheme selects the *k*-servers out of group servers, by flooding the request to the network depending on the number of responses it receives. The second and third scheme forms multiple components such that each component has atleast *k*-members and less than *k*-members, respectively.

In Pei-Jung et al. (2008) clustering and virtual backbone techniques (VBAF) are used to establish anycast tree and forward gate is used to decrease the overhead of responses received from duplicate service packets. A virtual backbone structure is used to set up stable routing paths and clustering scheme is used to reduce the length of the routing paths. Anycast tree is established according to the leaf node on the virtual backbone. When a client node sends out a service request message to the anycast tree, the client is responded by its nearest or best server.

Density of nodes through count of routes (RCBA) is discussed in Martin et al. (2009) to route the packets to anycast group member. Minimum count of hops when forwarding packet and count of routes are the two metrics used to find the best path to the server. The work given in Shyr-Kuen and Pi-Chung (2012) constructs an anycast tree from a cluster-based virtual backbone in a MANET. It finds a path from client to nearest server using anycast tree. For each branch node in the anycast-tree, a routing table is established. The nodes in the routing table are then used to generate a shorter path.

The clustering and virtual backbone techniques to establish anycast tree (SDBA) is discussed in Shyr-Kuen et al. (2008). A virtual backbone structure is used to set up stable routing paths and the clustering scheme is used to reduce the length of routing paths. The Ad Hoc On-demand Distance Vector (AAODV) routing protocol extended to support anycast routing is presented in Jidong et al. (2004).

The work given in Martin and Takuro describes the probability of connected route to anycast member as a function of dynamicity and density of the network. Anycast routing scheme chooses the shortest path routing as well as considers node degree density (NDAR) of hosts in the network through count of routes to the anycast group member. Weight value mechanism is adopted to select an optimal anycast member (OAM) in Wang (2010). Through entropy, the average available bandwidth and the average moving velocity are used to calculate anycast members weight value which can indicate the performance of the routing from the IPv6 ingress gateway to the anycast member.

The combination of ant based routing and clustering models are discussed in Jianping et al. (2009) to solve multi-constrained anycast routing. Anycast QoS routing algorithm based on Genetic Algorithm is studied in Shi and Shen (2010). It adopts the idea of dissimilarity to make population diversity, and the theory of simulated annealing to adjust the fitness function so as to inhibit the premature convergence.

A multi-path multi-gateway wireless mesh network anycast routing protocol based on ant colony optimization is presented in Song et al. (2010). For gateway selection, distributed computing and heuristic searching of ant colony algorithm are used. Multi-sink load balanced reliable forwarding for video delivery in a multi-sinked sensor network for target tracking is proposed in Sinan et al. (2012). To provide load balancing among the sinks, it proposes a sink selection mechanism based on fuzzy logic for the frame forwarding which evaluates the traffic density in the direction of each sink by combining two dynamic criteria which are the number of contenders and the buffer occupancy levels in the neighborhood with the static distance criterion.

Load distribution strategy by adopting distributed resource discovery and dynamic request-redirection mechanisms by using anycast, considering traffic load and network proximity is discussed in Mukaddim and Rajakumar (2009) for content delivery network servers. A distributed algorithm for sink selection in wireless sensor network is discussed in Trivino et al. (2011). Sensor nodes determine goodness of being the next hop for every online transmission. The estimation is supported by fuzzy logic based system which takes into the account the connectivity of the source, the connectivity of the candidate and the candidate's residual energy.

Tables 1 and 2 present the summary and comparison of some of the above mentioned anycast routing in MANETs.

As per literature survey, limitation of the related works are as follows: (1) lack of combination of anycast path and server stability mechanisms in MANETs, (2) multi-constrained QoS anycast routing in MANETs is not supported, (3) lack of robust and reliable route discovery and maintenance, and (4) FIS based routing solutions consider fixed membership functions and does

Table 1
Features comparison of anycast routing in MANETs.

Protocol	Operation	Advantages	Disadvantages
DBA	Computes potential field of group member to select proximity or density based routing	Very robust to route failures	Needs periodic refresh of potential values of the neighbours, more anycast members-more overhead
KA	Predicts TTL considering number of responses it receives to search servers	Reduces the routing control messages and network delay	Bottle neck on component header
VBAF	Usage of clustering and virtual backbone techniques to construct anycast tree	Support any K services and attain the backup, reduces control overhead	Optimization of parameter K-services
RCBA	Density of the server is decided by number of routes and hops	Advantage of shortest path routing and density of the host in the network	Each node needs periodic refresh about number of hops and count of routes to anycast group member, control overhead is more
SATR	Usage of clusters and virtual backbone techniques to construct anycast tree	Increases forwarding efficiency of the transmission	High routing table maintenance overhead
AAODV	Extension to AODV for anycast routing	Service discovery	Route maintenance is diffcult
NDAR	No. of routes connecting anycast member is computed by prob.of distribution fn. and connected links	Better recovery under link/node failure	Less robust
QARA	Adapts ANFIS to decide QoS fulfillment	More robust to link or node failure, less control overhead	More route discovery time because of learning process

Table 2
Summary of anycast routing works.

Protocol	S	A	SS	OMF	MQoS	RDT	K
DBA	x	x	x	–	x	Less	x
KA	x	x	x	–	x	Less	Yes
VBAF	x	x	x	–	x	Less	Yes
RCBA	x	x	x	–	x	Less	x
SATR	x	x	x	–	x	Less	Yes
AAODV	x	x	–	–	x	Less	x
NDAR	x	x	x	–	x	Less	x
QARA	Yes	Yes	Yes	Yes	Yes	More	x

S, soft computing approach; A, usage of agents; SS, server stability; OMF, optimized membership functions; MQoS, multi−constrained QoS; RDT, route discovery time; K, k−services; x, not present; –, not applicable.

not perform fine tuning of FIS. Hence, we propose a anycast routing protocol to choose a path with QoS guarantees and to select a server with capacity guarantee.

1.2. *Software agents*

Agents are the autonomous programs activated on an agent platform of a host. Agents use their own knowledge base to achieve the specified goals without disturbing the activities of the host. They have two special properties: mandatory and orthogonal, which make them different from the standard programs. Mandatory properties are autonomy, reactive, proactive and temporally continuous. The orthogonal properties are communicative, mobile, learning and believable (Anh and Karmouch, 1998).

Mobile agent is an itinerant agent which contains program, data, execution state information, migrates from one host to another host in a heterogeneous network, and executes at a remote host until it completes a given task. By nature, mobile agents are flexible modular entities that can be created, deployed and deleted in real time. The mobile code should be platform independent, so that it can execute at any remote host in a heterogeneous network environment. Inter-agent communication can be achieved by message passing, remote procedure call (RPC) or common knowledge base (blackboard) (Chess et al., 1995).

A mobile agent platform comprises agents, agent server, interpreter and transport mechanisms. The agent server is responsible for receiving mobile agents and sending it for execution by the local interpreter. Agents can be written in Java, Tcl, Perl and XML languages. Agent interpreter depends on the type of agent script/language used. An agent platform supports following services: agent creation, agent execution, agent migration, transport for mobile agents, agent security and persistence. Some of the Java-based agent platforms are Aglets, Grasshopper, Concordia, Voyager and Odyssey (Lange and Oshima, 1999). The advantages of adopting mobile agents are as follows:

1. Mobile agents can reduce network traffic compared to the traditional client-server approaches and maintain load balancing, thus increase performance of network nodes specially in MANETs.
2. Mobile agents can interact, collaborate, and communicate with environment. Mobile agents can perform important tests, which could be used to generate multiple paths to the individual servers through a network.
3. Mobile agents can execute in asynchronous and autonomous fashion. This autonomy along with platform and system independence make them ideal for maintaining and repairing the anycast QoS path, whenever node/link fails.
4. The mobile agent can encapsulate the protocol code. When protocol is upgraded, only the mobile agent has to be altered. Anycast manager agent can upgrade the protocol by adding some parameters and code based on required services.
5. Mobile nodes running on battery power in MANETs do not have enough power to run complex routing protocols. An alternative is to use mobile agents to perform routing operations and thus reduce complexity and network traffic. Therefore, saving important battery life of mobile nodes in MANETs.

Security threats due to the mobile agent's existence may affect in following ways: agent to platform, agent to agent and platform to agent. Possible types of misbehavior exhibited by agents and agent platforms under security threats are as follows: masquerading, denial of service, unauthorised access, repudiation, eavesdropping, alteration, and copy and replay. Security of mobile agents can be provided through two ways: (1) security to agent from agent platform (host) and (2) security to host (agent platform) from agent. Mechanisms used to ensure security in the above-listed situations are (Borselius, 2002; Jansen, 2002) software-based fault isolation, safe code interpretation, signed code, state appraisal, path histories, proof carrying code, partial result encapsulation, mutual itinerary recording, itinerary recording with replication and voting, execution tracing, environmental key generation and computing with encrypted functions.

Agent-based schemes comprising static or mobile agents offer several advantages as compared to traditional approaches: overcome latency; reduce network traffic; encapsulate protocols; flexibility; adaptability; software re-usability and maintainability; and facilitate the creation of customised dynamic software architectures (Manvi and Venkataram, 2004).

1.3. Our contributions

This paper considers a problem of optimizing FIS to meet QoS requirement of user by employing ANFIS with set of software agents as well as attempts to establish robust route discovery and maintenance procedures. Set of agents used in the model are static Anycast Manager Agent (AMA), static Optimization Agent (OA), and mobile Anycast Route Creation Agent (ARCA). AMA triggers, communicates and coordinates with OA and ARCA to establish QoS anycast path to the best server.

The proposed scheme operates in the following steps. (1) OA at client, optimizes membership functions of bandwidth, link delay, packet loss rate for Fuzzy Inference System (FIS) development by using Adaptive Neuro-Fuzzy Inference System (ANFIS) according to the user requirement. (2) ARCA is engaged to explore multiple paths from client to anycast servers through intermediate nodes. These also, gather intermediate nodes information such as available bandwidth, link delay, residual battery power, anycast servers stability and is made available at client. Server stability factor is derived from residual battery power and mobility metric. (3) AMA at client determines QoS factor from the collected information by using optimized FIS for each path and selects a QoS anycast path based on QoS factor and stability factor, and (4) ARCA is employed to maintain the QoS path in the event of node/link failures.

Our contributions in comparison to existing works are as follows. (1) To tailor the membership functions of QoS parameters for optimization of FIS. (2) Intelligent mobile agents usage to discover multiple paths from client to servers. (3) Computation of QoS factor for each of the path by using optimized FIS and server stability factor based on mobility and residual battery power. (4) Identification of multi-constrained QoS path to anycast server. (5) Robust path maintenance mechanism under link/node failure, and (6) Comparing the performance of the proposed scheme with shortcut tree based anycast (SATR) routing in MANETs (Shyr-Kuen and Pi-Chung, 2012).

The reason to consider SATR scheme for comparison with the proposed work is that SATR adapts node clustering, virtual backbone and control gate mechanisms to construct shortcut anycast tree for fast delivery of the packets to a server, reducing end-to-end delay and proposed protocol also needs the reduction in the end-to-end delay of packets to support multimedia applications.

Rest of the paper is organized as follows. QoS anycast routing in MANETs using ANFIS is given in Section 2. Simulation model for proposed scheme is presented in Section 3. Result analysis is given in Section 4. Finally, Section 5 concludes the work.

2. Agent driven QoS anycast routing in MANETs based on ANFIS

This section describes the network environment, computational models, anycast routing agency, and QoS anycast routing scheme.

2.1. Network environment

We consider a MANET scenario in which number of nodes are separated by distance (between consecutive nodes) as shown in the Fig. 1. Each of the node moves with a different mobility in different direction and computes available bandwidth, residual battery power, packet loss rate, and link delay periodically. Each node has a finite transmission range. Anycast server group

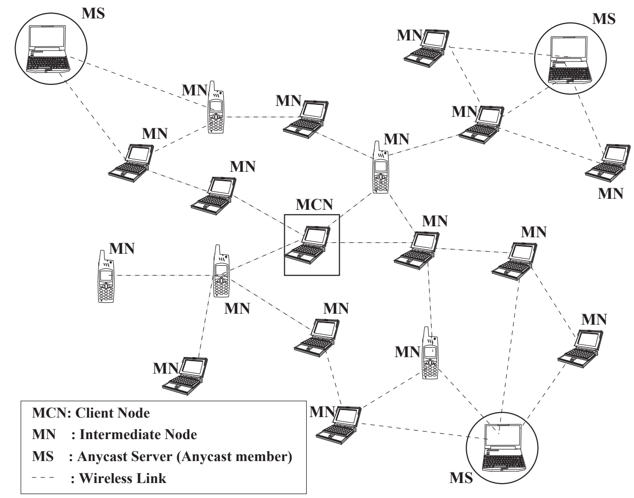

Fig. 1. Anycast environment.

members may be located in any part of a given geographical area and respond to the same address known as anycast address. Network consists of number of anycast groups. The number of members in anycast group may vary from one group to another. A client is informed of a list of replicated anycast servers address and is asked to select appropriate one and does not care which server is selected.

The model assumes that an agent platform is available in every node and consists of an agency in which static and mobile agents reside. However, if an agent platform is unavailable, the agents communicate by traditional message exchange mechanisms. We assume that agents have protection from hosts on which they execute. Similarly, hosts have protection from agents that can communicate on available platform. The secured platform consists of protection from denial of execution, masquerading, eavesdropping, etc. Both static and mobile agents are deployed on each of the nodes. All nodes are equipped with Global Positioning System (GPS) receiver for obtaining location and time. However, if some nodes do not have GPS facility, they can use localization algorithms.

2.2. List of notations

The list of notations and acronyms used in our work description are listed in Table 3.

2.3. Computational models

This section describes the computational models for calculation of QoS parameters like, bandwidth, link delay, packet loss rate, residual battery power, and stability of server. Also, Adaptive Neuro-Fuzzy Inference System to tune the membership functions is explained.

2.3.1. Computation of QoS parameter values

Client is required to know the QoS parameter values (residual bandwidth, link delay, and packet loss rate) of all intermediate nodes to compute QoS factor by using optimized FIS. The inputs to FIS are the status of the intermediate nodes such as bandwidth, link delay, and residual battery power. Hence, these are computed periodically at every node. Servers compute the server stability periodically and is derived from the parameters like mobility of the node and residual battery power.

Table 3
Notations and acronyms.

Notation	Description
BW	Residual bandwidth
DL	Link delay
PR	Packet loss rate
I_i	Idle time period
T_B	Observation interval time to calculate bandwidth
BW_{total}	Total channel bandwidth
T_D	Total interval to calculate link delay
T_P	Total interval to calculate packet loss rate
T_M	Total interval to calculate server stability
ψ	QoS value
Γ	QoS factor
υ	Mobility of node
ζ	Residual battery power of node
δ	Server stability factor
γ	Drain rate
η, σ, ρ, and ϵ	Consequent parameters of fuzzy rules
a_i, b_i, and c_i	Antecedent parameters of fuzzy rules
B_{less}, B_{more}	Linguistic labels for bandwidth
D_{less}, D_{more}	Linguistic labels for link delay
P_{less}, P_{more}	Linguistic labels for packet loss rate
μ	Membership function value (0–1) of QoS parameters
O_x^y	Output of xth node in yth layer of ANFIS model
w	Firing strength of fuzzy rule
w'	Normalized firing strength output
RMSE	Root mean square error
KB	Knowledge base
AMA	Anycast manager agent
OA	Optimization agent
ARCA	Anycast route creation agent

Residual bandwidth: It is estimated by considering channel status of radio, i.e., the idle periods of a shared wireless media. Each node listens to the channel status to determine idle period during observation interval time 'T_B'. Idle time period 'I_i' is computed by increasing the count from end of previous busy time to the start of next busy time of the channel. The total idle time consists of several idle slots, say 'n'. Total idle time is the summation of all 'n' idle times. Residual bandwidth 'BW' for total channel bandwidth of 'BW_{total}' at a node is given by the following equation:

$$BW = \frac{\sum_{i=1}^{n} I_i}{T_B} * BW_{total} \tag{1}$$

Link delay: The link delay 'DL' at each node is composed of input queuing delay, processing delay, propagation delay, and retransmission delay. Let '$P_{i,a}$' be the time at which the packet 'i' has arrived at the node, and '$P_{i,c}$' be the time when data packet is acknowledged. The link delay for packet is the difference of arrival time and acknowledged time. Average link delay for 'm' packets in a certain period 'T_D' is given by the following equation:

$$DL = \frac{\sum_{i=1}^{m} (P_{i,c} - P_{i,a})}{T_D} \tag{2}$$

Packet loss rate: It is due to buffer over flow and retransmission. Packet loss rate 'PR' is the sum of number of packets lost, 'N_o', due to overflow, and number of retransmission packets, 'N_r', for a time period 'T_P', and is given by the following equation:

$$PR = \frac{N_o + N_r}{T_P} \tag{3}$$

2.3.2. Server stability model

Each server maintains a stability database information which helps client to select a best server. Server computes the following parameters: mobility (υ), residual battery power (ζ) and server stability factor (δ).

Server mobility: Mobility of server 'i' (υ_i) is obtained by finding position of the server at different time for observation time interval 'T_M' and is given by Eq. (4). 'T_M' is discretized and position of server is found at each discrete interval.

$$\upsilon_i = \frac{\sum_{t=1}^{T_M} \sqrt{|x_t - x_{(t-1)}|^2 + |y_t - y_{(t-1)}|^2}}{T_M} \tag{4}$$

where (x_t, y_t), $(x_{(t-1)}, y_{(t-1)})$ are (x,y) co-ordinates of a server at time t and $(t-1)$, respectively.

Residual battery power: Life time (ζ_i) of battery of server 'i' is calculated by considering residual energy 'E_i' and drain rate 'γ' of server and is expressed by the following equation:

$$\zeta_i = \frac{E_i}{\beta * \gamma_{old} + (1-\beta)\gamma_{new}} \tag{5}$$

where γ_{old}, γ_{new} are previously and newly calculated drain rate values. Drain rate is defined as the energy consumed every second. 'E_i' is residual battery power. Co-efficient β represents constant value between 0 and 1.

Server stability factor: It is value computed at each server. The 'ith' server stability factor, 'δ_i', is given by the following equation:

$$\delta_i = m * \upsilon_i + n * \zeta_i \tag{6}$$

where υ_i and ζ_i are mobility and residual battery power of a server i, and m and n are constants where, $m + n = 1$ and $m, n \neq 0$.

2.3.3. Optimizing membership functions

In this section, we tune FIS considering user necessity by using ANFIS. ANFIS is an integrated system of artificial neural network and fuzzy inference system, which combines the learning capabilities of neural network and reasoning capabilities of fuzzy logic. ANFIS is used to shape parameters of the membership functions of fuzzy antecedent parameters, as well as linear consequent parameters of fuzzy rules for Takagi–Sugeno FIS. This adjustment allows the fuzzy system to learn from data that is modelled.

Our ANFIS structure is based on three inputs, four rules, one output, and first order Takagi–Sugeno fuzzy model. Sugeno method is used due to its computational efficiency and is more suitable for developing a systematic approach to generate FIS from given input-output data set. The three inputs are bandwidth (BW), delay (DL) and packet loss rate (PR) and has a single output as QoS value (Ψ). BW, DL, PR are non linear parameters. Figure 2 depicts

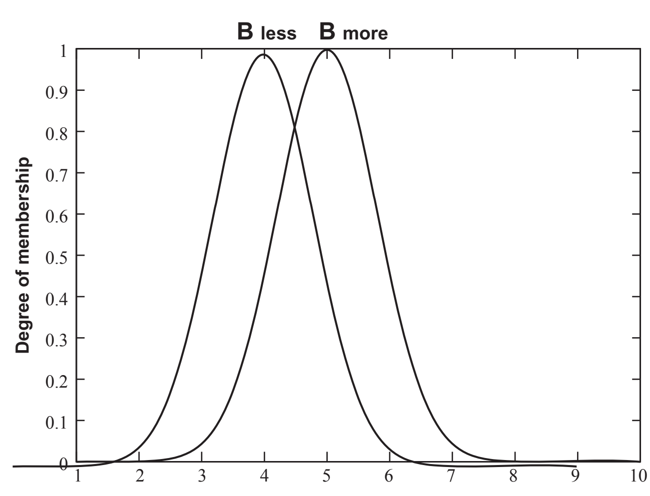

Fig. 2. Membership functions for bandwidth.

the membership functions for the Bandwidth. The link delay and packet loss rate are also represented in the same way. The symbols B_{less}, B_{more} for *BW*, D_{less}, D_{more} for *DL*, and P_{less}, P_{more} for *PR*, represent linguistic labels for dividing the membership functions (MFs).

The generalized bell-shaped MFs is defined by Eq. (7) for *BW*.

$$\mu_{B_{less}}(BW) = \frac{1}{1 + \left(\frac{BW - c_i}{a_i}\right)^{2b_i}} \quad (7)$$

where a_i , b_i and c_i are the parameters of membership function governing the centre, width and slope of the bell-shaped membership function. The parameters a_i , b_i and c_i are referred to as antecedent parameters of if–then rules and are non-linear. As the values of these parameters change, the bell-shaped function varies accordingly, thus this change will give various forms of membership functions for fuzzy set as required according to the data set. In a similar way, *DL* and *PR* are also expressed by using bell-shape MFs.

ANFIS incorporates fuzzy "if–then" rules involving premise and consequent parts of Sugeno type FIS. For the first order, rule set with four fuzzy "if–then" rules are as follows.

Rule 1: If *BW* is B_{less} and *DL* is D_{less} and *PR* is P_{less} then $f_1 = \eta_1 BW + \sigma_1 DL + \rho_1 PR + \epsilon_1$

Rule 2: If *BW* is B_{more} and *DL* is D_{less} and *PR* is P_{less} then $f_2 = \eta_2 BW + \sigma_2 DL + \rho_2 PR + \epsilon_2$

Rule 3: If *BW* is B_{more} and *DL* is D_{less} and *PR* is P_{more} then $f_3 = \eta_3 BW + \sigma_3 DL + \rho_3 PR + \epsilon_3$

Rule 4: If *BW* is B_{more} and *DL* is D_{more} and *PR* is P_{less} then $f_4 = \eta_4 BW + \sigma_4 DL + \rho_4 PR + \epsilon_4$

where η_i, σ_i, ρ_i, and ϵ_i for i=1 to 4 are linear consequent parameters to be settled, and f_x is output for x=1 to 4.

ANFIS is a multi-layered network, in which each layer performs a particular task. It consists of five layers (Layers 1 to 5) namely, fuzzification, rules, normalization, defuzzification and output layers. Each layer consists of number of nodes performing the similar functions. Circle indicates a fixed node, whereas a square indicates an adaptive node. The output signals from the nodes of the previous layer are the input signals for the current layer as shown in Fig. 3.

Layer 1: Antecedent parts of fuzzy rules are represented by nodes in this Layer 1. Layer 1 is fuzzification layer and every node is an adaptive node. Fuzzification layer determines degree of membership functions of inputs *BW*, *DL*, and *PR*. Outputs of this layer for different nodes are given by the following equation:

$$O^1_{B_{less}} = \mu_{B_{less}}(BW); O^1_{B_{more}} = \mu_{B_{more}}(BW)$$
$$O^1_{D_{less}} = \mu_{D_{less}}(DL); O^1_{D_{more}} = \mu_{D_{more}}(DL)$$
$$O^1_{P_{less}} = \mu_{P_{less}}(PR); O^1_{P_{more}} = \mu_{P_{more}}(PR) \quad (8)$$

For example $O^1_{B_{less}}$ represent output of node B_{less} of layer 1. $\mu_{B_{less}}(BW)$ represent input membership function value on a defined bell shaped curve. The generalized bell shape membership function is chosen, because of its continuous and differential property which is very suitable to apply learning algorithm in parameter tuning phase. For certain values of *BW*, *DL* and *PR*, degree of membership is shown in Fig. 4 and the values are $\mu_{B_{less}}(BW) = 0.6$, $\mu_{B_{more}}(BW) = 0.4$, $\mu_{D_{less}}(DL) = 0.3$, $\mu_{D_{more}}(DL) = 1$, $\mu_{P_{less}}(PR) = 0.8$, $\mu_{P_{more}}(PR) = 0.5$.

Layer 2: Layer 2 is the rule layer and is labelled as R_x. Every node in this layer are fixed. The output of this layer is w_x, which represents the firing strength of each rule. Firing strength is obtained by fuzzy logic operator 'intersection' which gives the product of the input membership grades as given in the following equation:

$$O^2_x = w_x = \mu_{B_i}(BW)\mu_{D_i}(DL)\mu_{P_i}(PR); \quad \forall i = less \text{ or more} \quad (9)$$

where O^2_x is output of node x in the layer 2, ∀ x=1,…,4.

Layer 3: Layer 3 is the normalization layer represented as *N*. Every node in this layer is a fixed node. Each node in this layer receives inputs from all nodes in the rule layer, and calculates the normalized firing strength of a given rule. The normalized firing strength of the *x*th node is the ratio of the *x*th rule firing strength to the sum of all rules firing strengths as represented by the following equation:

$$O^3_x = w'_x = \frac{w_x}{\sum_{x=1}^{4} w_x} \quad (10)$$

where x=1,…,4.

Layer 4: Layer 4 is the defuzzification layer and is represented by f_x. Every node in this layer is an adaptive node. A defuzzification node calculates the weighted consequent value of a given rule and is simply the product of the normalized firing strength and the first order polynomial (for a first-order Sugeno model). Thus, the output of this layer are given in the following equation:

$$O^4_x = w'_x f_x = w'_x(\eta_i BW + \sigma_i DL + \rho_i PR + \epsilon_i) \quad (11)$$

where w'_x is a normalized firing strength from layer 3, x=1,…,4, and η_i, σ_i, ρ_i, and ϵ_i are consequent parameters of if–then rules.

Layer 5: It is output layer consisting of single fixed node and is represented as ∑. The output of node is the summation of all

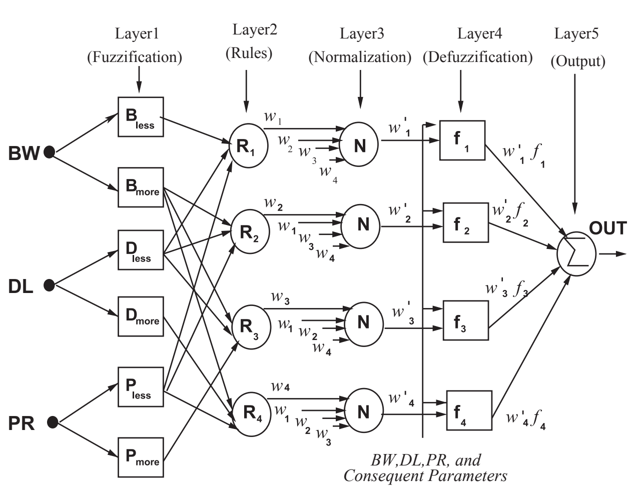

Fig. 3. The structure of ANFIS model.

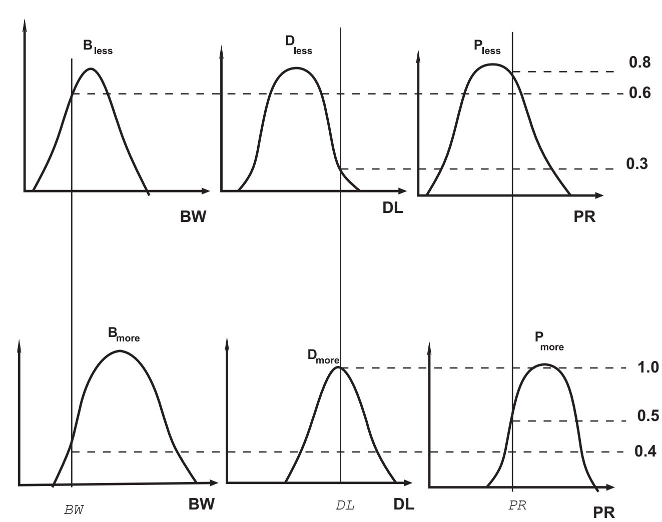

Fig. 4. Membership functions of the inputs.

incoming signals. Hence, the overall output of the model is given by the following equation:

$$O^5 = \sum_{x=1}^{4} w'_x f_x = \frac{\sum_{x=1}^{4} w_x f_x}{\sum_{x=1}^{4} w_x} \quad (12)$$

ANFIS has two adaptive layers the first and the fourth. In the first layer modifiable antecedent parameters a_i, b_i, and c_i are related to the input membership functions. In the fourth layer, consequent parameter set η_i, σ_i, ρ_i, ϵ_i are modifiable.

The available data set was divided in to two subsets randomly. These data set were training data set and checking data set. Initially a_i, b_i, and c_i are untrained and are assumed. The task of the learning algorithm for the ANFIS architecture is to train all consequent and antecedent parameters according to the data set. Adjusting modifiable parameters is a two step process known as hybrid learning algorithm. In the forward pass, rule consequent parameters are identified by least square estimator and are adjusted. While antecedent parameters remain fixed. In the backward pass, the errors are propagated backward and premise parameters are adjusted by gradient method (Jyh-Shing and Roger, 1993).

2.4. QoS anycast routing agency (QARA)

In this section, we present QoS routing agency employed in the proposed scheme. Fig. 5 shows anycast agency and agent interactions. The agency is located in each node and consists of knowledge base (KB), static Anycast Manager Agent (AMA), Optimization Agent (OA), and mobile Anycast Route Creation Agent (ARCA). AMA creates OA and ARCA to perform optimization of FIS and to discover multiple paths to individual anycast servers.

Knowledge base (KB): It contains information of itself and of intermediate nodes and servers. The information of itself as a client are as follows: QoS requirement of user, anycast servers address, multiple path ids from client to individual servers, forward anycast routing table (QoS satisfied path ids to reach servers), status of node (connected/not connected to the QoS anycast path). The information of the intermediate nodes comprises of gathered available bandwidth, link delay and residual battery power. It contains stability factor at server. KB is read and updated by AMA, OA and ARCA.

Optimization agent (*OA*): It is a static agent and is triggered only at the client. OA optimizes the membership functions of bandwidth, delay and packet loss rate (refer Section 2.3.3) for the development of FIS based on the client requirement, OA provides the optimized FIS information to AMA. OA is disposed after optimizing FIS.

Anycast route creation agent (*ARCA*): It is a mobile agent employed to search the multiple paths from client to individual anycast servers through intermediate nodes by cloning method. ARCA carries client address, anycast server address, hop count and sequence number and the path traversed to reach all the servers through the intermediate nodes. Upon reaching servers, it traces back the same path it has traversed to reach client. It provides intermediate state information (*BW*, *DL*, *PR*) and server stability factor to AMA of client. ARCA is responsible for disseminating the QoS factor computed by AMA to all the nodes on the path to perform the local patch-up in case of failures. ARCA is triggered periodically whenever node/link fails to recover the path.

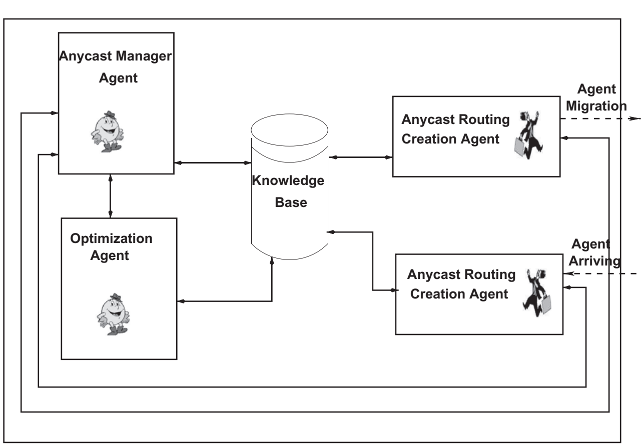

Fig. 5. QoS anycast routing agency.

Table 4
Client AMA information.

Anycast ID	128.98.101.34				
Path	**C - 3 - 9 - S14**				
Node ID	*BW*(*Mbps*)	*DL*(*ms*)	*PR*(%)	Ψ	Γ
3	5	122	54	38	94
9	4	100	32	56	
Path	**C - 4 - 10 - S14**				
Node ID	*BW*(*Mbps*)	*DL*(*ms*)	*PR*(%)	Ψ	Γ
4	8	112	38	42	92
10	6	132	58	50	
⋮	⋮				
Server ID	δ				
S11	0.9				
S12	0.8				
S13	0.6				
S14	0.7				

Anycast manager agent (*AMA*): It is static agent and initiates OA and ARCA and knowledge base. It controls and coordinates activities of QoS anycast routing agency. AMA at all the node calculates available bandwidth, link delay and residual battery power periodically and server stability factor (refer subsections 2.3.1 and 2.3.2). AMA performs some different tasks, at client, intermediate and server nodes. (1) AMA at client: (i) holds multiple paths from client to the individual anycast servers, available *BW*, *DL*, *PR* of intermediate nodes and stability 'δ' of servers, (ii) holds the optimized FIS information as shown in Table 4, where path is considered from client 'C' to server 'S14'. (iii) performs FIS procedure to evaluate QoS value 'Ψ' of each node on the path and evaluates QoS factor 'Γ' for each path, (iv) selects an anycast server based on 'Γ' of the paths and server stability factor 'δ', and (v) disseminates respective QoS factor of that path through ARCA to each of the nodes on the path. (2) AMA at intermediate node: maintains forward anycast routing table at the intermediate nodes. (3) AMA at server: it periodically computes and holds server stability factor. AMA triggers ARCA to maintain the path when node/link fails.

2.5. QoS anycast routing scheme

The proposed scheme works in following phases: (1) optimization of membership functions, (2) mobile agent based multiple path search to individual servers, (3) selection of QoS anycast path, and (4) QoS anycast route maintenance.

2.5.1. Regulating fuzzy parameters

AMA of source node receives the user application requirement and triggers OA to optimize membership functions, bandwidth, link delay and packet loss rate (refer subsection 2.3.3) according to the QoS requirement. OA updates the optimized information in AMA and is disposed off.

2.5.2. Exploring multiple paths to individual servers

Whenever a client wishes to send data to anycast server, AMA of client dispatches ARCA to initiate multiple path discovery to each of the individual servers. ARCA carries client ID, anycast server ID, addresses of the traversed path, hop count and sequence number to reach its neighbours. When neighbours receive ARCA, they check for the sequence number of a packet generated by the source. If it is duplicate, ARCA is disposed. Else, ARCA communicates and collaborates with AMA of intermediate nodes to know, if there exists a path to reach anycast server. If a path exists, ARCA does not flood further, but collects the forwarding anycast route table to reach server, and travels back to client. This intelligence of ARCA makes different from that of message passing.

If path does not exist in intermediate nodes to reach a server, then ARCA is cloned. Cloned ARCA travels further to reach individual servers. The agent cloning is a technique of creating an agent similar to that of parent, where cloned agent contains the code and information of the parent agent. The cloning is done at multiple levels; cloned agent contains identification of parents at all its levels. A child agent can communicate either to any one of its parents who are within the range or to any of its parents at a given level. In our work, cloned agent has parent residing at previous node. Upon reaching server, ARCA traces back the same traversed path collecting the available bandwidth, link delay and residual battery power at each of the intermediate nodes between client and servers and server stability factor. This information is provided to AMA of client.

2.5.3. Selection of QoS anycast path

AMA of client estimates QoS value by using tuned FIS (refer subsection 2.3.3). The inputs to the FIS are available bandwidth, link delay, and packet loss rate of intermediate node. And the output of FIS is QoS value. AMA performs three steps in FIS; fuzzification of inputs, rule base inference mechanism and defuzzification (Timothy, 2009). After evaluating the QoS values (Ψ) of all the nodes on the path by adopting optimized FIS, AMA computes QoS factor 'Γ_i' for path 'i' given by the following equation:

$$\Gamma_i = \sum_{j=1}^{n} \Psi_{ij}; \quad \forall i = 1 \ldots m \tag{13}$$

where Ψ_{ij} is QoS value of node j on the path i. n number of intermediate nodes on the paths and m total number of paths from client to servers. AMA selects a QoS path '$P(C_s)$' given by Eq. (14) and checks the server stability of QoS path. If server stability factor is greater than the threshold, selects the path. Else, AMA selects the next QoS path with server stability higher than threshold.

$$P(C_s) = \left(\min_{r_i \in m} \Gamma_i \right) \tag{14}$$

If there are several number of paths having the minimum Γ, then AMA compares server stability δ connected to paths. A path with maximum δ is selected to route QoS anycast packets.

2.5.4. QoS anycast route maintenance

When forwarding a packet, AMA of each intermediate node transmitting the packet is responsible for confirming that the packet has been received by its next node along the anycast transmission path. If confirmation from an upstream node is not received by a downstream node after limited number of retransmissions for the packet, AMA of downstream node will assume that link/node has broken. AMA of downstream node searches for the local patch-up path to reach any of the anycast servers with higher QoS factor. Else, AMA of downstream node triggers ARCA to forward '$Anycast_{error}$' packet to reach client. If a client AMA has any other QoS anycast route in its route cache, it will send packets using the new anycast route. Otherwise, it initiates a new route discovery to discover a new QoS anycast route.

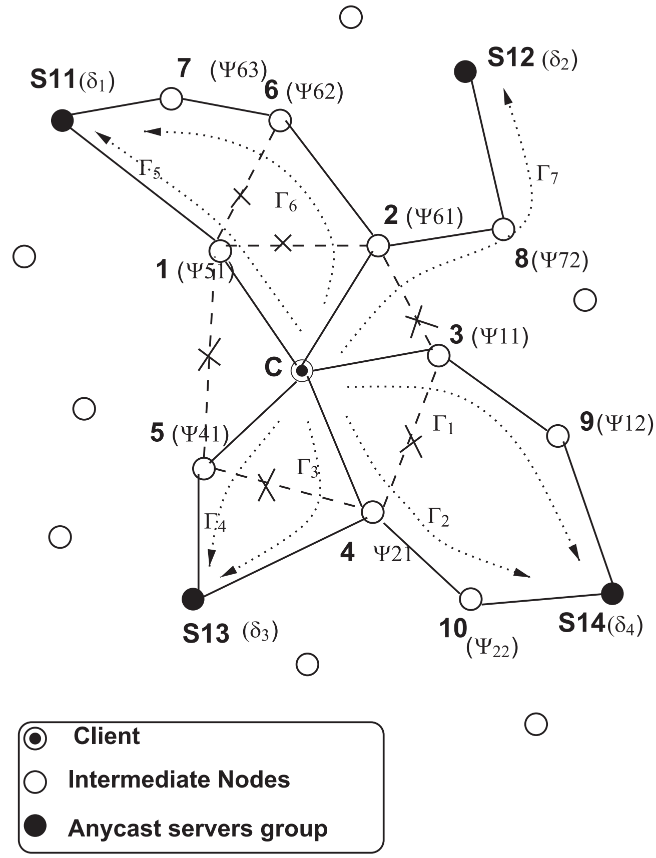

Fig. 6. QoS anycast path selection.

Consider an example to illustrate the concept as shown in Fig. 6. Client 'C' requests for the QoS path to anycast server. There are four servers S11, S12, S13, and S14 connected to the same anycast address. Multiple paths traced by ARCA from client to the individual servers is shown by solid line. Dotted line indicates duplicate ARCAs and are disposed hence these links are not connected to the path. The multiple paths discovered by ARCA are Path 1: C-3-9-S14; Path 2: C-4-10-S14; Path 3: C-4-S13; Path 4: C-5-S13; Path 5: C-2-8-S12; Path 6: C-1-S11; and Path 7: C-2-6-7-S11. Ψ_{ij} is QoS value for a node j on a path i, Γ_1 to Γ_7 are the QoS factors computed for paths 1–7, and δ_1, δ_2, δ_3, and δ_4 are server stability factors.

Let the values be $\Gamma_1 = 72$, $\Gamma_2 = 56$, $\Gamma_3 = 82$, $\Gamma_4 = 83$, $\Gamma_5 = 56$, $\Gamma_6 = 90$, $\Gamma_7 = 48$, $\delta_1 = 0.9$, $\delta_2 = 0.8$, $\delta_3 = 0.6$, $\delta_4 = 0.7$. AMA selects a path 7 which has minimum Γ, and server stability for the path selected is 0.8 which is more than server threshold 'Ser_{th}'=0.5. When there are more than two paths with a same QoS factor. If $\Gamma_6 = 90$, $\Gamma_7 = 90$ for path 6 and 7, respectively, then servers stability S11(δ_1) = 0.9 and S12(δ_2) = 0.8 are compared and server 1 with path 6 is selected to route the anycast packets.

Some of the limitations of our ANFIS based QoS anycast routing in MANETs are as follows. (1) FIS tuned for one specific QoS requirement cannot be used for different QoS requirement without re-tuning. (2) Scheme does not consider server traffic during routing selection, and (3) AMA is heavy weight to perform various functions to select a QoS path to anycast server and route maintenance.

3. Simulation

The proposed QoS Anycast Routing Agency (QARA) is simulated along with Shortcut Anycast Tree Routing (SATR) in various network scenario by using C programming language to verify the performance and operation effectiveness of the scheme. ANFIS model was developed by using MATLAB 7.0 Fuzzy Logic Toolbox to optimize the membership functions. Then the outputs are used in the C program. The simulation is carried out on Pentium IV using 'C' programming language with a confidence interval of 95%.

In this section, we describe the simulation model and the simulation procedure. Simulation model for the MANET scenario consists of six models: (1) network model, (2) propagation model, (3) mobility model, (4) channel model, (5) traffic model, and (6) ANFIS model.

3.1. Simulation model

Network model: MANET consists of a collection of 'N_{max}' mobile nodes placed randomly in an area '$l \times b$' m^2. The nodes move within the area. The coverage area around each node has a bandwidth 'BW_{total}' that is shared among its neighbours. All the nodes between the links are bidirectional. The number of servers are 'S' and clients are 'C'.

Propagation model: Free space propagation model is used with transmission range for each node as 'T_r' for a single-hop distance. Received power at the receiver is dependent on gain of transmitter 'G_t' and receiver 'G_r' antenna, distance between the two nodes 'D_n' and system loss 'L' and transmitted power 'P_w' which are appropriately assumed. Received signal strength is measured in terms of signal-to-noise ratio (SNR). A packet will be accepted if it is received with SNR higher than the fixed SNR threshold 'SNR_{th}' at the receiver. Packet propagation delays (per hop) are generated proportional to distance between the nodes, i.e., we consider 'm' second per meter. Link delay varies between 0 and 'L_d' and residual power varies between 0 and 'R_p'.

Mobility model: Mobility model uses random way point. The mobility of the nodes varies uniformly from 'M_{min}' to 'M_{max}'. Maximum number of nodes allowed for the movement is 'M_n' within the area. Each host is initially placed at a random position. As the simulation progresses, each host pauses at its current location for a period 'P_t', the pause time, and then randomly chooses a new location to move. Each host continues this behavior, alternately pausing and moving to a new location, for the duration of the simulation time 'S_t'. If a node tries to go out of the boundary, its direction is reversed (Bouncing ball model).

Channel model: Each wireless link is associated with a channel noise that consists of white noise (additive white Gaussian noise) and other channel interferences that defines the link quality. To access the channel, ad hoc nodes use CSMA/CA (802.11b) MAC layer standard protocol to avoid possible collisions and subsequent packet drops. We set queue length in MAC layer to be infinite to avoid packet dropping due to buffer overflow.

Traffic model: Constant bit rate traffic was generated by using fixed size packet, 'PKT_{size}' bytes long. The packets are transmitted with 'BW_{trans}'. For the traffic model, various anycast group sizes are used to assess the performance of route establishment overhead, packet delivery ratio, average end-to-end delay, path success ratio, average number of hops.

ANFIS model: ANFIS MATLAB Toolbox requires several steps as follows. The training data contains 'T_d' human expert conditions while the checking data contains 'C_d' random cases. Generate "2 2 2" ANFIS structure. Check the training data against the bell membership function shape. Number of epochs 'E_n' is varied. Four input parameters were controlled, namely 'BW', 'DL' and 'PR' and output 'QoS'. After training, the shapes of the membership functions were slightly modified depending on the knowledge provided by experts and the input/output data pair. The lowest Root Mean Square Error (RMSE) indicates that the model is optimal.

3.2. Simulation procedure

Simulation procedure for the proposed scheme is as follows:

1. Generate ad hoc network and traffic across the network.
2. Apply ANFIS scheme to optimize membership functions.
3. Identify multiple paths to reach individual servers and gather intermediate information and server stability factor.
4. Apply optimized FIS to evaluate QoS factor.
5. Discover QoS anycast path to any of the server from the multiple paths considering even the server stability.
6. Invoke route maintenance procedure.
7. Compute the performance parameters of the system.

The following performance metrics are used for evaluating the proposed scheme.

1. *Packet delivery ratio*: It is the ratio of the number of data packets actually delivered to the destinations to the number of data packets received and is expressed in percentage.
2. *Path success ratio*: It is the ratio of the number of connection request discovered to the servers to the number of QoS connection requests. It is expressed in percentage.
3. *Average end-to-end delay*: It is defined as the average time taken to transmit predefined number of packets from client to anycast server. It is expressed in seconds.
4. *Overall control overhead*: It is defined as the ratio of the total number of control packets or agents to the total number of packets generated to perform anycast communication.
5. *Average number of hops*: The traveled distance is assessed as the average number of hops that QoS anycast packets undertake until they reach server.

The following simulation input parameters were used for simulation. $N_{max}=100$, $l \times b$ $m^2=1000$ m^2, $BW_{total}=10$ Mbps, $T_r=250$ m, $G_t=1, G_r=1$, $D_n=0$ to 225 m, $L=1$, $P_w=100$–500 mw, $SNR_{th}=12$ dB, $m=0.001$, M_{min} to $M_{max}=0$–12 m/s, $P_t=10$ s, $S_t=200$ s, $BW_{trans}=2$ Mbps, $PKT_{size}=512$ bytes, $Ser_{th}=0.5$, $S=5$–25, $C=2$–8, $B_t=2$ Mbps, $L_d=0$–20 ms, $R_p=0$–200 mw, $E_n=100$, $T_d=100$, $C_d=100$, $hop_{count}=10$, $T_B=T_D=T_P=T_M=10$ s. The bell shaped parameters for the residual bandwidth BW with different linguistic terms are for B_{less}, a, b, and c are 0.25, 1.5 and 2.5, respectively; for B_{more}, a, b, c are 2.5, 4, 7.5, respectively. The bell shaped parameters for the link delay DL with different linguistic terms are for D_{less}, a, b, and c are 2, 5, and 5, respectively; for D_{more}, a, b, and c are 2, 5, and 14, respectively. The bell shaped parameters for the residual power PR with different linguistic values are for P_{less}, a, b, and c are 20, 40, and 50, respectively, for P_{more}, a, b, and c are 20, 40, and 150, respectively.

4. Results

This section presents the results obtained during simulation. We compare results of proposed work with Shortcut Anycast Tree Routing (SATR) in MANETs.

4.1. Packet delivery ratio (PDR)

The effect of PDR over variation in number of nodes, mobility of the nodes, and number of servers are studied. The scalability of the system is tested by finding the PDR with increase in number of nodes from 40 to 100 with fixed mobility of the node as 5 m/s as shown in Fig. 7. In both QARA and SATR, the PDR increases consistently with increase in number of nodes and increases with increase in number of servers from 5 to 15.

SATR has a slightly low packet delivery ratio than QARA due to high drop rates. In QARA, AMA uses only those paths which satisfy the QoS and also server is selected in such a way that its server factor is more than the 'Ser_{th}' threshold value. Hence breakage of paths as well as failure of nodes is less. And also, there exist alternate multipaths present in the AMA to the server and hence this makes QARA significantly better than SATR and increases PDR as the number of nodes increase. PDR increases as number of servers increase because client can get connected to any other server when the connected server has mobility.

The effect of variation in mobility of the nodes on PDR for number of servers 5, 10 and 15 is shown in Fig. 8. Comparing PDR of QARA with SATR, there is a notable difference as the mobility increases. As the speed of the node increases new paths discovery is performed in SATR which causes loss of packets. Whereas in case of QARA, mobile agent ARCA rebuilds the transmission path and make the packets to be transferred to maximum extension whenever either node/link fails or nodes move out of range. So, QARA has higher PDR than the SATR.

Figure 9 shows the performance of PDR to the different mobility speed 5 m/s, 8 m/s, and 10 m/s with the variation in number of clients. PDR is more in QARA, compared to SATR and decreases in both the schemes as the mobility of the nodes increase. AMA selects the QoS path depending on the optimized FIS decision. FIS output is a function of the QoS parameters like bandwidth, link delay, and packet loss rate which facilitates AMA to select the optimal QoS path leading to increase in PDR in QARA. SATR uses the shortest path from the constructed tree and the path selected may not satisfy the QoS requirement leading to loss of packets, hence PDR is less.

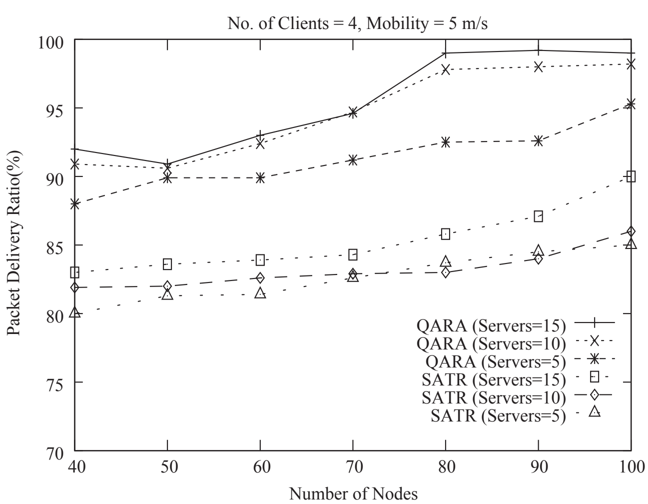

Fig. 7. Packet delivery ratio vs. number of nodes.

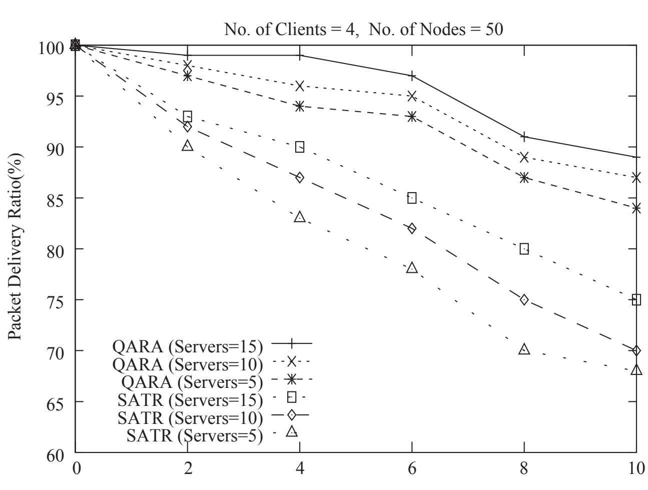

Fig. 8. Packet delivery ratio vs. mobility speed (m/s).

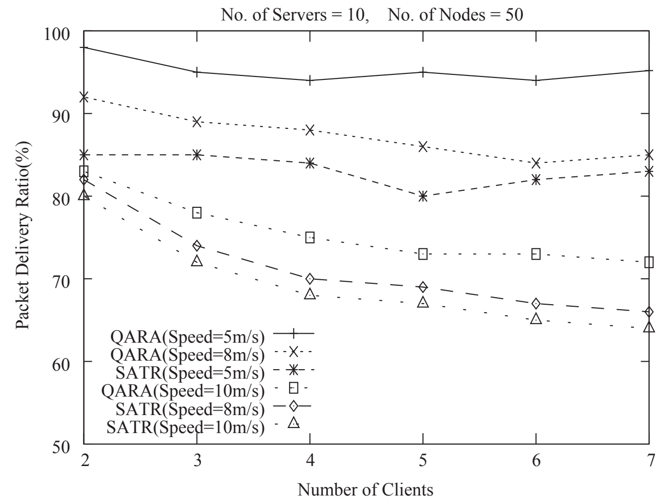

Fig. 9. Packet delivery ratio vs. number of clients.

4.2. Path success ratio

The comparison of path success ratio for different number of nodes under varying mobility scenarios is shown in Fig. 10. As the number of nodes increase, path success ratio increases constantly. The path success ratio is more in QARA as compared to SATR scheme. Path selection by AMA is based on QoS satisfied path, hence QARA has more path success ratio. For SATR scheme, client gets connected to any one of the server depending on the shortest path tree and do not consider QoS satisfaction for the path, therefore path success ratio is less.

4.3. Average end-to-end delay

End-to-end delay includes route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer time. End-to-end delay for varying number mobility of nodes is depicted in Fig. 11. As mobility speed of the nodes increase, end-to-end delay also increases. This delay is high for SATR scheme as compared to QARA. Some of the reasons for QARA to perform better than SATR scheme are as follows: (1) AMA selects the paths containing the intermediate nodes which fulfill the QoS requirements, hence breakage of path is less (2) server selected by AMA has stability more than the 'Ser_{th}' and hence client gets connected to the same server for long time. (3) ARCA are used to recover routes against link failures, node failures and mobility of nodes with local patching for failed links/nodes. Rediscovery of paths leads more number of retransmissions and hence the end-to-end delay increases significantly in SATR and remains almost constant in QARA as speed of the node increases.

End-to-end delay for varying number of clients with different mobility speed is depicted in Fig. 12. As the mobility values increases, end-to-end delay also increases with increase in number of clients because there are number of packets which are generated by each of the client to the server.

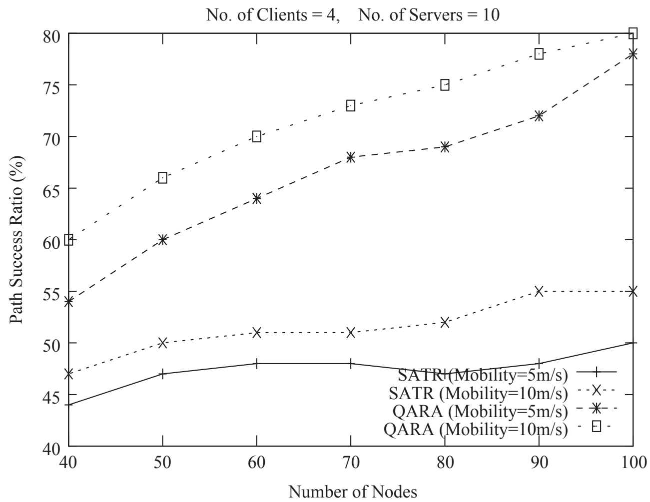

Fig. 10. Path success ratio vs. number of nodes.

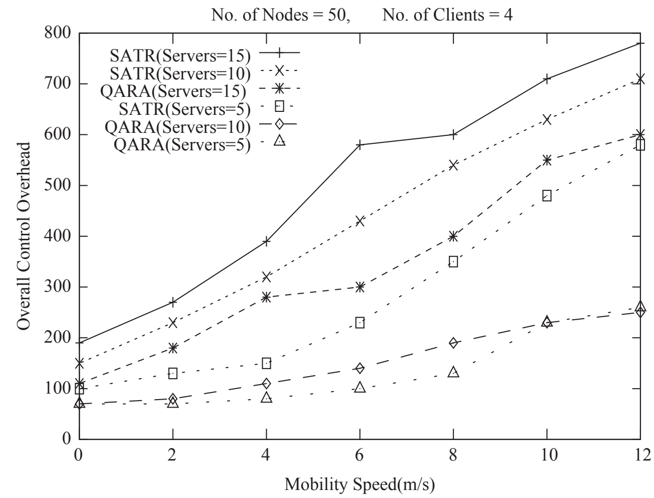

Fig. 13. Overall control overhead vs. mobility speed.

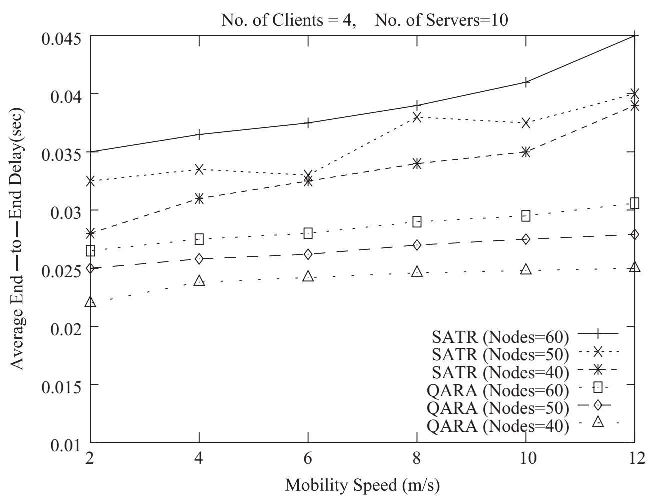

Fig. 11. Average end-to-end delay vs. mobility speed.

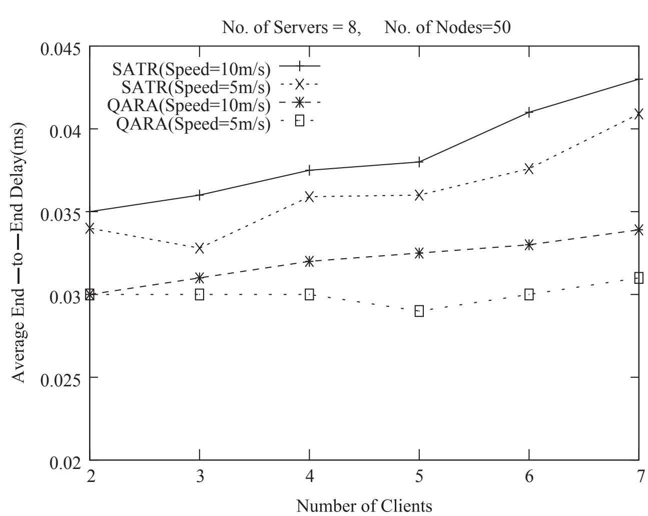

Fig. 12. Average end-to-end delay vs. mobility speed.

4.4. Overall control overhead

Figure 13 outlines the control overheads for varying the mobility values and number servers. Control overheads are more for increased number of servers and mobility. Because of the network connectivity, as the nodes mobility increases, control overheads are also increased. This is due to the fact that more number of mobile agents are generated for discovering the multipaths to the individual servers and to maintain the QoS anycast path. In SATR, anycast path is less distributed at low mobility speeds up to 4 m/s, but control packets rapidly increase after the speed more than 4 m/s due to new route discovery. In QARA, as mobility speed increases, control overheads almost remains the same as the multipaths are cached in AMA to select the anycast path instead of rediscovering. Hence control packets are well controlled in QARA than SATR.

4.5. Average number of hops

Average number of hops for different number of servers under varying mobility speed conditions of nodes is illustrated in Fig. 14. As the number of servers increase, average number of hops decreases gradually. Compared to SATR scheme, QARA scheme average number of hops is more. This is because AMA selects the QoS satisfied path for the QoS anycast routing out of the total collected multipaths. SATR is based on shortcut tree and reduces the average number of hops compared to QARA.

5. Conclusions

This paper proposed an intelligent model to find QoS anycast route in MANETs by using ANFIS. The scheme employed an agency consisting of static Anycast Manager Agent (AMA), Optimization Agent (OA), and mobile Anycast Route Creation Agent (ARCA). OA optimizes fuzzy inference system according to the requirement by using neural technique. ARCA discovers multipaths from client to individual servers. AMA at client chooses a QoS anycast path from a set of multipaths by employing optimized fuzzy technique and server with higher stability. The agent based architectures provides flexible, adaptable, asynchronous mechanisms, take autonomous decisions for distributed network management. The proposed work is simulated for various MANET network environments to validate its performance. From the simulations, we observed that the proposed scheme performs better than SATR scheme in terms end-to-end delay, packet delivery ratio, path success ratio and control overhead.

Some of the additional research issues that can be considered for future extension of the work are as follows: (1) extending the

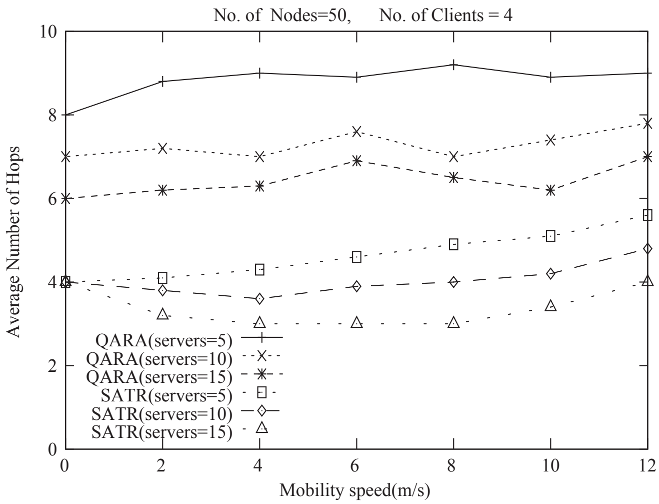

Fig. 14. Average number of hops vs. mobility speed.

work by considering more number of QoS parameters to decide QoS satisfaction of a node, (2) the work can be extended to construct multiple anycast route for different QoS requirements, (3) determining optimal membership functions for adaptive user requirement, and (4) we have further plans to make an exhaustive comparison of various techniques based on ants, fuzzy, probability, and neural network based approaches for anycast routing in MANETs.

References

Abraham A. Adaptation of fuzzy inference system using neural learning. Studies in fuzziness and soft computing, vol. 181. Berlin, Heidelberg: Springer-Verlag; 53–83.

Anh PhamV, Karmouch A. Mobile software agents: an overview. IEEE Communication Magazine 1998;36(7):26–7.

Azzedine Boukerche, Begumhan Turgut, Nevin Aydin, Mohammad ZA, Ladislau B, Damla Turgut. Routing protocols in ad hoc networks—a survey. International Journal of Computer Networks 2011;55(13):3032–80.

Bing Wu, Jie Wu. k-anycast routing schemes for mobile ad hoc networks. In: Proceedings of the IEEE 20th international conference on parallel and distributed processing (IPDPS-2006). Washington, USA; 2010. p. 129–38.

Borselius N. Mobile agent security. Journal of Electronics and Communication 2002;14(5):211–8.

Chess D, Benjamin G, Harrison C, Levine D, Paris C. Itinerant agents in mobile computing. IEEE Personal Communication 1995;2(5):35-49.

Dow CR, Hsuan R, Hwang SF. Design and implementation of anycast routing in mobile ad hoc networks. In: Proceedings of the IEEE 8th international conference on advanced communication technology, vol. 1. Phoenix Park: Korea; 2006. p. 419–24.

El-Hajj W, Al-Fuqaha A, Guizani M, Hsiao-Hwa Chen. Fuzzy on efficient network planning and routing in large-scale MANETs. IEEE Transactions on Vehicular Technology 2009;58(7):3796–801.

Fernanda Weiden, Peter Frost. Anycast as load balancing feature. In: Proceedings of ACM 24th international conference on large installation system administration (LISA-10). Berkely, USA; 2010. p. 1–6.

Ge Zhi-hui, Li Tao-shen. A novel gateway load-balance algorithm for wireless mesh network. Journal of China Universities of Posts and Telecommunications 2011;18(2):75–8.

Hue Xu, Xianen Wu, Sadiadpour Garcia-Luna. A unified analysis of routing in MANETs. IEEE Transactions on Communications 2010;58(3):911–22.

Jansen WA. Countermeasures for mobile agent security. International Journal of Computer Communication 2002;25(15):1392–401.

Jian Li, Hongyan Cui, Ru Gao, Jia Du, Jianya Chen. The application of an improved particle swam optimization for multi-constrained QoS routing. In: Proceedings of the IEEE international workshop on database technology and applications. Wuhan, China, 2010. p. 1–5.

Jianping Yu, Yaping Lin, Yong Wang. Ant based reliable multi-constrained anycast routing for sensor networks. International Journal of Distributed Sensor Networks 2009;5(1):31.

Jidong Wu, Oliver Stanze, Kilian Weniger, Martina Zitterbart. Prototype implementation of anycast-based service discovery for mobile ad hoc networks. Germany; 2004. p. 87–95. ⟨telematics.tm.kit.edu/english/staff230.php⟩.

Jyh-Shing, Roger Jang. ANFIS: adaptive network based fuzzy inference system. IEEE Transactions on Systems, Man, and Cybernetics 1993;23(3):665–85.

Lange DB, Oshima M. Seven good reasons for mobile agents. ACM International Journal of Communication 1999;43(3):88–9.

Manvi SS, Venkataram P. Applications of agent technology in communication: a review. International Journal of Computer and Communication 2004;27(15):1493–508.

Martin Macuha, Takuro Sato. Considering node degree in anycast routing in wireless ad hoc networks. ⟨www.jstage.jst.go.jp/article/tjsst/2/2/2267/pdf⟩.

Martin Macuha, Takuro Sato, Sheng-Chang Chen. Route-count based anycast routing in wireless ad hoc networks. In: Proceedings of the IEEE 70th conference on vehicular technology fall. Tokyo, Japan; 2009. p. 1–5.

Mie Mie Thaw. Fuzzy based multi-constrained QoS distance vector routing in MANETs. In: Proceedings of the IEEE 2nd international conference on computer and automation engineering, vol. 3, Singapore; 2010. p. 429–33.

Mukaddim Pathan, Rajakumar Buyya. Resource discovery and request-redirection for dynamic load sharing in multi-provider peering content delivery networks. International Journal of Network and Computer Applications 2009;32(4):976–90.

Pei-Jung Lin, Chyi-Ren Dow, Sheng-Chang Chen. An efficient anycast scheme for discovering K services in mobile ad hoc networks. In: Proceedings of the ACM 5th symposium on performance evaluation of wireless ad hoc sensor and ubiquitous networks. New York, USA; 2008. p. 33–7.

Rajashekhar Biradar, Sunilkumar Manvi. Review of multicast routing mechanisms in mobile ad hoc networks. International Journal of Network and Computer Applications 2012;35(6):221–39.

Shi Rui, Shen Shi-Lei. Improved anycast QoS routing algorithm based on genetic algorithm. In: Proceedings of the international conference on computer application and system modeling (ICCASM 2010). Kaifeng, China; 2010. p. 167–70.

Shi Rui, Shen Shi-Lei, Bai Chen-Xi, HeXin. Improved anycast QoS routing algorithm based on genetic algorithm. In: Proceedings of the IEEE international conference on computer application and system modeling (ICCASM 2010), vol. 14. Taiyuan, China; 2010. p. 167–70.

Shyr-Kuen Chen, Pi-Chung Wang. Shortcut anycast tree routing in MANETs. In: Proceedings of the 26th IEEE international conference on advanced information networking and applications. Washington, USA; 2012. p. 635–40.

Shyr-Kuen Cheng, Pi-Chung Wang, Tung-Shou Chen. Service discovery based on anycasting in mobile ad-hoc networks. In: Proceedings of the ACM 2nd international workshop on agent oriented software engineering challenges for ubiquitous and pervasive computing. Sorrento, Italy; 2008. p. 9–14.

Sinan Isik, Mehmet Yunus Donmez, Cem Ersoy. Multi-sink load balanced forwarding with a multi-criteria fuzzy sink selection for video sensor networks. Journal of Computer Networks 2012;56(2):615–27.

Song Ling, Cao Jie, Yang Xue-jun. Multi-path anycast routing based on ant colony optimization in multi-gateway WMN. In: Proceedings of the IEEE 5th international conference on computer science and education. Hefei, China; 2010. p. 167–70.

Sunil Taneja, Ashwani Kush. A survey of routing protocols in mobile ad hoc network. International Journal of Innovation, Management, and Technology 2010;1(3):279–85.

Taoshen Li, Zhihui Ge. Multiple QoS constraint anycast routing algorithm based on adaptive genetic algorithm. In: Proceedings of the ACM third international conference on genetic and evolutionary computing. USA; 2009. p. 89–92.

Timothy J Ross. Fuzzy logic with engineering applications. second edition India: Wiley; 2009.

Trivino Cabrera, Uste AJ, Cintrano D. Fuzzy logic based and distributed gateway selection for wireless sensor networks. In: Proceedings of the Springer international conference on advances in intelligent and soft computing, vol. 89; Springer; 2011. p. 243–8.

Ulas C Kozat, Leandros Tassiulas. Service discovery in mobile ad hoc networks: an overall perspective on architectural choices and network layer support issues. Elsevier Ad Hoc Networks 2004;2(1):23–44.

Vincent Lenders, Martin May, Bernhard Plattner. Density-based anycast: a robust routing strategy for wireless ad hoc networks. IEEE/ACM Transaction on Networking 2008;16(4):852–63.

Wang Xiaonan. Design and implementation of anycast services in ad hoc networks connected to IPv6 networks. Journal of Networks 2010;5(4):403–10.

Wu-Hsiao Hsu, Ming-Chih Tung b, Li-Yuan Wu. An integrated end-to-end QoS anycast routing on DiffServ networks. International Journal of Computer Communications 2007;30(6):1406–18.