

Research on Innovating, Applying Multiple Paths Routing Technique Based on Fuzzy Logic and Genetic Algorithm for Routing Messages in Service - Oriented Routing

Nguyen Thanh Long^{1,*}, Nguyen Duc Thuy², Pham Huy Hoang³

¹Informatic Center of Hanoi Telecommunications, 75 Dinh Tien Hoang, Hoan Kiem, Hanoi, Viet Nam

²Post and Telecommunications Institute, 122 Hoang Quoc Viet, Nghia Tan, Cau Giay, Hanoi, Viet Nam

³Hanoi University of Science Technology, 1 Dai Co Viet, Hanoi, Viet Nam

Abstract

MANET (short for Mobile Ad-Hoc Network) consists of a set of mobile network nodes, network configuration changes very fast. In content based routing, data is transferred from source node to request nodes is not based on destination addresses. Therefore, it is very flexible and reliable, because source node does not need to know destination nodes. If We can find multiple paths that satisfies bandwidth requirement, split the original message into multiple smaller messages to transmit concurrently on these paths. On destination nodes, combine separated messages into the original message. Hence it can utilize better network resources, causes data transfer rate to be higher, load balancing, failover. Service Oriented Routing is inherited from the model of content based routing (CBR), combined with several advanced techniques such as Multicast, multiple path routing, Genetic algorithm to increase the data rate, and data encryption to ensure information security. Fuzzy logic is a logical field study evaluating the accuracy of the results based on the approximation of the components involved, make decisions based on many factors relative accuracy based on experimental or mathematical proof. This article presents some techniques to support multiple path routing from one network node to a set of nodes with guaranteed quality of service. By using these techniques can decrease the network load, congestion, use network resources efficiently.

Keywords: Ad hoc network, MANET, QOS, bandwidth guarantee, multiple paths, routing, service

Received on 30 December 2013, accepted on 10 September 2014, published on 17 February 2015

Copyright © 2015 Nguyen Thanh Long *et al.*, licensed to ICST. This is an open access article distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/sis.2.4.e2

1. Introduction

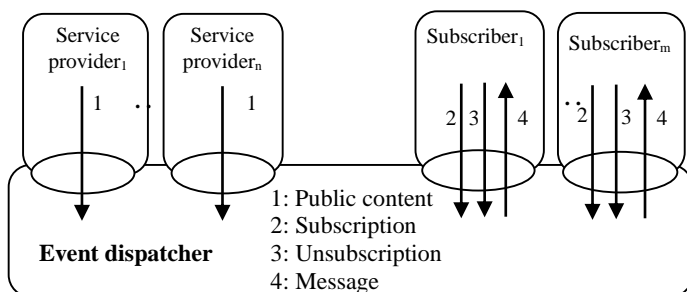


Figure 1. Service based routing model.

In the SBR routing, the sender does not indicate message's receiver by the unicast or multicast use. Instead, it simply pushes messages to the network. It defines the routing based on the messages it

carries. It determines the appropriate message class based on message content based on its key-value pairs or regular expressions. Therefore, in SBR routing the receiver determines the transmission of messages, not by the sender. Communication based on content services increases the independence, flexibility in the distributed architecture. In order to perform service based routing, we have to organize data into messages, there are some message types will be constructed. That are:

1. Subscription/ unsubscription message: In order to register/ unregister node's requested content.
2. Content message: for publishing node's content.

*Corresponding author. Email: longtpm@vnpt-hanoi.com.vn

3. Advertisement message: Use this message to advertise node's content, direct subscription from subscribers to match publishers, avoid flooding subscription all over the networks. When a node needs to advertise its content: (i) Make an advertisement message; (ii) It will broadcast this content advertisement message to the network. (iii) It's content advertisement will be cached with nodes of the network.

4. Update request and reply sender message pair: in order to update the routing table of content based routing protocol. By using minimum spanning trees to forward data from source node to a set of destination nodes in the service based routing. To maintain this multicast tree, the source node periodically has to send an update request message to destination nodes. When the destination node receives this request, it will reply with a reply sender message.

5. Route request and route reply message pair: in order to detect routes (network topology), We use a pair of messages: route request and route reply message. When We want to forward data from one source node to a set of destination nodes in the case of transmitting matched content to subscribers. We have to build a multicast tree with the root is the current node, leaf nodes are destination nodes. We have to make update request message, broadcast this message to the network. When this message reaches destination nodes or forwarding nodes that have up-to-date route information, a reply message is created to send to the source node. After specified time-out the source node has received enough route information from destination nodes to the source node. We use these received routes to build multiple paths by using some techniques below.

| | |
|------------------|-----------------|
| Source Address | Source Address |
| Request number | Request number |
| Type | Type |
| Time to live | Time to live |
| Timeout | Timeout |
| Route | Route |
| Free time slots | Route |
| Destination List | Free time slots |

Figure 2. Structures of RREQ and RREP.

2. Multiple Paths Routing

Routing process is the process to find some routes from a source node to one or more destination nodes. In order to build multiple path routing, in

this paper mention the clustering technique based on fuzzy logic. When many paths exist between a pair of source and destination nodes, use the genetic algorithm to find the optimal path.

2.1. Multiple path routing model

Assume network is defined by a weighted graph $G(T)=\{V(T), L(T)\}$, $V(T)$ and $L(T)$ are vertex and edge sets. Divide network into n clusters: $G = \{C_1, C_2, \dots, C_n\}$, in each cluster C_i use an inner routing protocol to find the set of routes: $RS = \{RS_{CH,1}, RS_{CH,2}, \dots, RS_{CH,n}\}$, $RS_{CH,i}$ is the route set between the cluster head CH and a node member i. So in order to find routes between two nodes (p, r) that belong to two clusters (G_i, G_j), use the formula:

$$RS_{pr} = RS_{pi} \times RS_{ij} \times RS_{jr} \quad (1)$$

Where x is the Descartes operator of two sets. So we have multiple paths between node p and r.

Assume the number of routes in $RS_{pi}, RS_{ij}, RS_{jr}$ are RC_1, RC_2, RC_3 respectively, the number of routes from p to r is:

$$RC_{pr} = RC_1 * RC_2 * RC_3 \quad (2)$$

2.2. Use fuzzy logic to cluster network

Fuzzy logic model

A fuzzy logic controller has some components:

- 1) Conversion functions: that convert input values into fuzzy values, fuzzy value is in the range [0, 1].
- 2) Use member functions to estimate fuzzy values. Calculate fuzzy outputs by some conversion formulas.
- 3) Inference functions: that use some inference rules to make decision outputs.
- 4) The conversion function converts membership functions' results into a fuzzy output:

(i) Case 1: there is one parameter:

$$\eta = \frac{\sum_{i=0}^n \sum_{j=1}^M x_i \mu_{F_j}(x_i)}{\sum_{i=0}^n \sum_{j=1}^M \mu_{F_j}(x_i)} \quad (3)$$

(ii) Case 2: there are more than one parameter:

$$\eta = \frac{\sum_{k=0}^m \sum_{i=1}^n \sum_{j=1}^M x_{ik} \mu_{F_j}(x_{ik})}{\sum_{k=0}^m \sum_{i=1}^n \sum_{j=1}^M \mu_{F_j}(x_{ik})} \quad (4)$$

Where M is number of membership functions F^j , x is fuzzy input, n is number of tests.

Choose cluster head of a cluster

Choose cluster head for a cluster based on two parameters:

- (i) Node remaining energy: E , has three member functions: L, M, H that are equal to three levels of energy: Low, Medium, High.
- (ii) Node mobility: M , has three member functions: C, A, F , that are equal to three levels of mobility: Close, Adequate, Far.

Make an inference rule in the table:

Table 1: Inference rules for selecting CH.

| | | | | |
|-------------|------------|----|----|----|
| $F_3(CH_N)$ | $F_1(E_N)$ | | | |
| | L | M | H | |
| $F_2(M_N)$ | C | RM | L | VL |
| | A | S | M | L |
| | F | VS | RM | M |

So get six levels for selecting CH: VS (very small), S (small), RM (rather medium), M (medium), L (large) and VL (very large).

Choose cluster members for a cluster

Select node members for a cluster based on two parameters:

- (i) Hop counts from node to CH: HC , there are three member functions: S, A, L , that are equal to three levels of hop count: Short, Average, Long.
- (ii) The bandwidth of the route from this node to CH: N, M, W , that are equal to three levels of bandwidth: Narrow, Medium, Wide.

Table 2. Inference rules for selecting nodes.

| | | | |
|-------------|-------------|----|----|
| $F_3(M_N)$ | $F_1(HC_N)$ | | |
| | S | A | L |
| $F_2(BW_N)$ | N | RM | VS |
| | M | S | RM |
| | W | VL | M |

So get six levels for selecting cluster members: VS (very small), S (small), RM (rather medium), M (medium), L (large) and VL (very large).

2.3. Choose a route by fuzzy logic

Assess route in each intermediate node by two parameters:

- (i) Remaining energy in node: E , this parameter has five member function VL, L, M, H, VH;
- (ii) Energy consumption for transferring an amount of data through this node: J . This parameter also has five member function VL, L, M, H, VH;

VL, L, M, H, VH are equal to five levels of energy: Very Low, Low, Medium, High, Very High. Assessing the membership value of each parameter by their member functions as in the diagram:

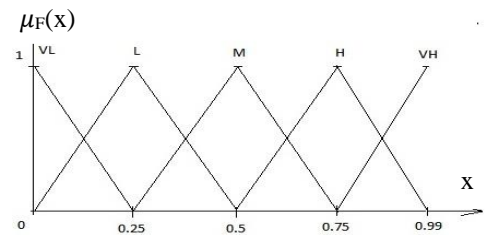


Figure 3. Member functions of each parameter.

Build inference rules as in the following table:

Table 3: Inference rule set for selecting routes.

| | | | | | | |
|-----|-----|----|----|----|----|----|
| S | J | | | | | |
| | VL | L | M | H | VH | |
| E | VL | HH | HM | HL | MH | MM |
| | L | HM | HL | MH | MM | ML |
| | M | HL | MH | MM | ML | LH |
| | H | MH | MM | ML | LH | LM |
| | VH | MM | ML | LH | LM | LL |

So We have 25 fuzzy rules to get 9 member functions to choose route on each node. Assess the fuzzy results by output member functions as in the following diagram:

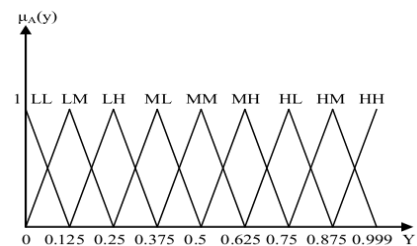


Figure 4. Assess fuzzy result (y).

From the route sets chosen by fuzzy logic to get some optimal routes by the GA algorithm as mentioned in the next section. Otherwise, it can build the optimal multicast tree in the case of transmitting data from one source node to some destination nodes simultaneously [10].

2.4. Find the optimal best routes by Genetic Algorithm

Establish the fitness function

Assume that there are k chromosomes, denote chromosome i by $R_i = \{g_1, g_2, \dots, g_n\}$, n is the number of genes in R_i .

The Fitness function is established on the basic of some QoS parameters S_{QoS} mentioned above:

$$S_{QoS} = \{p_1, p_2, \dots, p_m\} \quad (12)$$

Where p_i can be one of cost, bandwidth, energy, availability, ...

Define fitness of R_i by the function:

$$f_i = \prod_{k=1}^{n-1} \sum_{j=1}^m \alpha_j(p_j, c_{ik}) \quad (13)$$

α_j is a function that operates on two parameters p_j and a hop c_{ik} between gene g_k, g_{k+1} of chromosome R_i .

Establish Genetic Algorithm to find routes

The GA finds optimal QoS routes between a source (S) and a destination (D) that is simulated by some basic processes:

1. Collect all possible routes (chromosomes) between S and D.
2. Assign weights for all parameters of these chromosomes.
3. At each generation, choose the best chromosomes based on a fitness function for reproduction, mutation, crossover operations to get offsprings:
 - a. Select two fittest routes based on fitness function: $R_1 = \{r_{11}, r_{12}, \dots, r_{1n}\}$, $R_2 = \{r_{11}, r_{12}, \dots, r_{1n}\}$
 - b. Execute crossover on R_1 and R_2 to generate a new route: $R_3 = \{r_{31}, r_{32}, \dots, r_{3n}\}$.
 - c. Execute mutation on R_3 to prune the low energy nodes.
4. Get the optimal QoS route R.

Some techniques to execute crossover operations:

1. Firstly the condition for crossover operation that is two chromosomes have at least one common gene.
2. $R_3 = R_1 \theta R_2$, denote θ is crossover operation. Have to check if R_3 has a route loop in its path. If there exists route

loop, remove this route and choose the next common gene of these routes.

3. If R_3 is found, then apply mutation operator for it. Finally, add it to the next population for the next processing.

Technique to execute mutation operations:

1

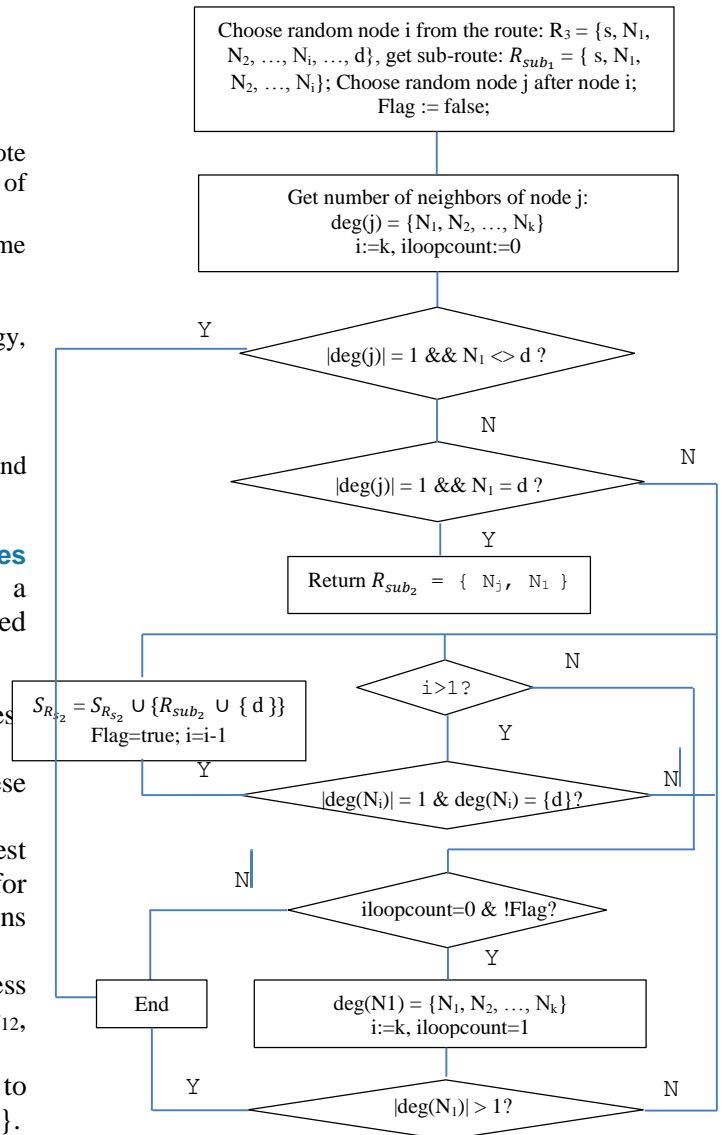


Figure 5. Algorithm for choosing a sub-route.

1. At first choose the first sub-route from this route $R_3 = \{s, N_1, N_2, \dots, N_i, \dots, d\}$, $R_{sub_1} = \{s, N_1, N_2, \dots, N_i\}$, node i is chosen randomly. Delete a set of nodes after i , choose the next sub-route by following algorithm:
2. Assume the remaining nodes of main route is: $\{N_j, N_{j+1}, \dots, d\}$, begin from the node j , define $deg(j)$ is the set of neighbors of j : $deg(j) = \{N_1, N_2, \dots, N_k\}$.

After this algorithm is done, get the $S_{R_{s_2}}$ is the set of some next sub-routes. Get the optimal route by the least hop count metric.

To simulate this algorithm by the diagram:

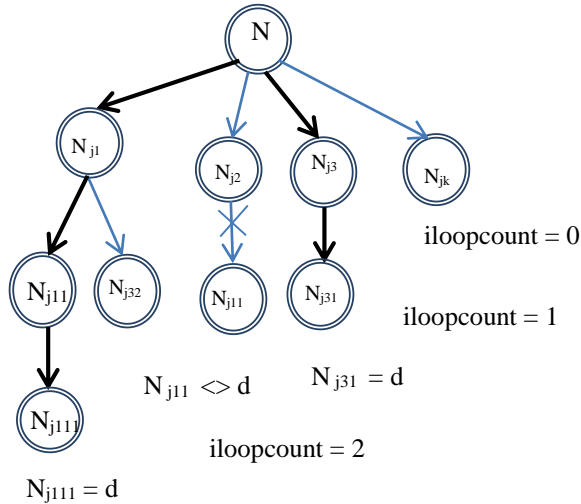


Figure 6. Mutation operator to choose the route.

In this diagram, it is very easy to realize that: $\text{Node}_{j31} \equiv \text{Node}_{j111} \equiv d$, and $S_{R_{s_2}} = \{(s, \text{Node}_{j1}, \text{Node}_{j11}, d), (s, \text{Node}_{j3}, d)\}$.

2.5. Build optimal multicast tree by GA algorithm

Build optimal multicast tree by route finding GA

For each found multicast tree by executing one of the methods in [10], find the optimal route for it by the above algorithm in section 3.4. Assume $L(T)$ is the list of found multicast trees. Following is an algorithm to find multiple trees and transmit data:

- For each multicast tree T_i in the list tree $L(T)$.
- For each route in this multicast tree T_i .
- Find optimal R_{ik} in T_i by the above genetic algorithm.
- Execute (c) simultaneously for multiple routes to increase speed to get optimal T_i in $L(T)$.
- Divide a large block of data into several smaller blocks to transmit by multiple trees have found to multiple destinations.

Build optimal multicast tree by GA Encoding multicast tree

Before applying GA to find an optimal multicast tree, it has to encode multicast trees into chromosomes. Use two arrays for each tree: (i) the first array S stores nodes' ID in the order by executing pre-order algorithm P of tree traversal depth first search algorithm; (ii) the second array T stores the parent's ID of each node. This algorithm P is executed recursively as described below:

- Assume that R is the root of the tree;
- The function is named `Scan` with two parameters: (i) a node A that will be visited; (ii) a reference parameter $iCount$, assign $iCount$ by 1, that stores the sequence number of current node A . So `Scan` is:


```

            Procedure Scan (byval R as Node, byref iCount as Integer)
            d) Begin
            e) Visit R by storing R in its order in the array S:
                a. S[iCount] = R.ID;
                b. If iCount > 1 then
                    i. T[iCount] = R.parent.ID;
                c. End if
                d. Increase iCount by one:
                    iCount += 1;
            f) Foreach (Node child in R.children)
            g) Execute Scan(child, ref iCount);
            h) End;
```

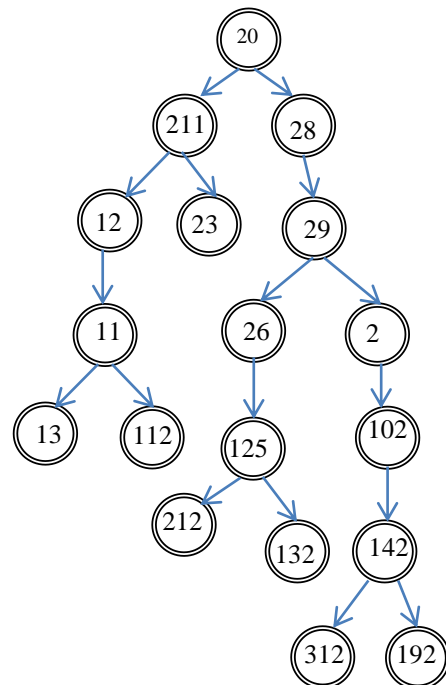


Figure 7. An example of multicast tree.

Applying function `Scan` for this tree: to get two arrays with contents:

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| O | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| r | | | | | | | | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| d | | | | | | | | | | | | | | | | | | |
| e | | | | | | | | | | | | | | | | | | |
| r | | | | | | | | | | | | | | | | | | |
| S | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 1 | 3 | 1 |
| | 0 | 1 | 2 | 1 | 3 | 1 | 3 | 8 | 9 | 6 | 2 | 1 | 3 | | 0 | 4 | 1 | 9 |
| | | | | | 2 | | | | | 5 | 2 | 2 | | | 2 | 2 | 2 | 2 |
| T | 0 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | 1 | 1 |
| | 0 | 1 | 2 | 1 | 1 | 1 | 0 | 8 | 9 | 6 | 2 | 2 | 9 | | 0 | 4 | 4 | |
| | | | | | | | | | | 5 | 5 | | | | 2 | 2 | 2 | |

Decoding chromosomes into multicast tree:

The pair of chromosomes S and T after applying GA will be decoded into multicast tree T by the algorithm:

- For $k=2$ to S.length Do
- $L(T) = L(T) \cup (S(k), S(T(k)))$;
- After the loop, get L(T) is the set of links of the tree T.

Establish the fitness function

Fitness function assesses each chromosome on the basic of delay, cost and residual energy of all links and nodes of the multicast tree:

$$F = \sum_{P_i \in P} \alpha(P_i, C_i) \quad (13)$$

Where: P is the set of chromosome's parameters, P_i, C_i are parameter and its weight for calculating the fitness level of each chromosome by function α . For example, $F = (\frac{C_1}{Delay_T}) * (C_2 Residual_Energy_T) * (\frac{C_3}{Cost_T})$, where C_1, C_2, C_3 are three constant values are chosen by user for easy estimating. In each round, estimate each chromosome in the population by this fitness function to choose the two best chromosomes for applying genetic operators consisting of crossover and mutation.

Genetic operators

Similar to genetic operators for algorithm to find routes.

- Crossover operator: $\{T_{new1}, T_{new2}\} = T_1 \Theta T_2$, in which $T_1 = \{T_{1first}, A, T_{1second}\}$, $T_2 = \{T_{2first}, A, T_{2second}\}$, A is the common node

of two chromosomes. So two result chromosomes can be:

- $T_{new1} = \{T_{1first}, A, T_{2second}\}$, $T_{new2} = \{T_{2first}, A, T_{1second}\}$,
- Check T_{new1} and T_{new2} for eliminating any route cycle and having the same destination list.
- If they don't satisfy these conditions, then find the next common A of two chromosomes and execute (b).
- Looping until finding two satisfied parent chromosomes (P) and get two child chromosomes or scan all common genes of P.

Mutation operator

Use mutation operator to eliminate low residual energy nodes of chromosomes. Assume that each pair of nodes of tree T also has at least one connection.

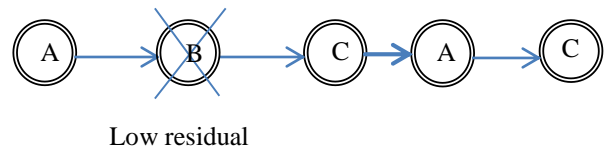


Figure 8. An example of applying the mutation operator.

Evaluation algorithm complexity

The fitness function has complexity $O(m*n)$, where m is the number of chromosomes and n is the average number of genes on each chromosome. Crossover operator has complexity $O(n*\log(n))$ and mutation operator has complexity $O(n)$. Hence this algorithm has complexity $O(n*(m+\log(n)))$.

3. Simulation and evaluation

3.1. Simulate the process to find the optimal route by GA algorithm

Some criteria to execute the GA algorithm:

- Population size: this is about 20 to 30 chromosomes;
- Crossover and mutation probabilities: Crossover probability P_c is about from 0.2 to 0.9, mutation probability P_m is about from 0.05 to 0.2.
- Chromosome size: this is about 20;

Following is a diagram that represents the simulation result of GA to find the optimal route with a number of routes varies from 4 to 20 routes,

the route size varies from 3 to 20. The number of simulations is 1000 times:

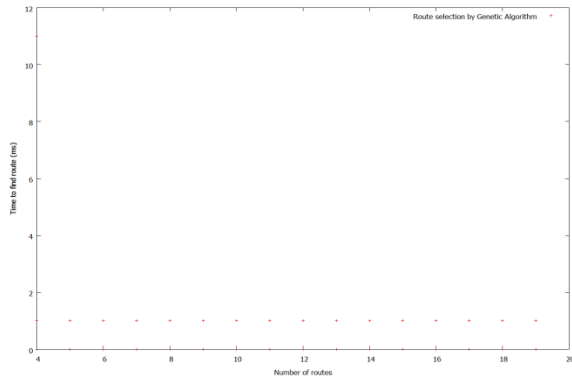


Figure 9. Diagram for representing simulation results of GA to find optimal route.

From this diagram, it is very easy to realize that time required to find the optimal route doesn't increase when number of routes increase. In particular, with 1000 times of simulations, but in the diagram there are the number of points that is less than 1000, so the algorithm execution is very stable in time. The points are marked by red color.

3.2. Simulate the process to find an optimal multicast tree by GA algorithm

Some criteria to execute the GA algorithm:

- a) Population size: this is about 2 to 100 chromosomes;
- b) Crossover and mutation probabilities: Crossover probability P_c is about from 0.2 to 0.9, mutation probability P_m is about from 0.05 to 0.2.
- c) Chromosome size: this is about 20;

Following is a diagram that represents the simulation result of GA to find the optimal multicast tree with a number of trees varies from 2 to 100 routes, the route size varies from 3 to 20. The number of simulations is 100 times:

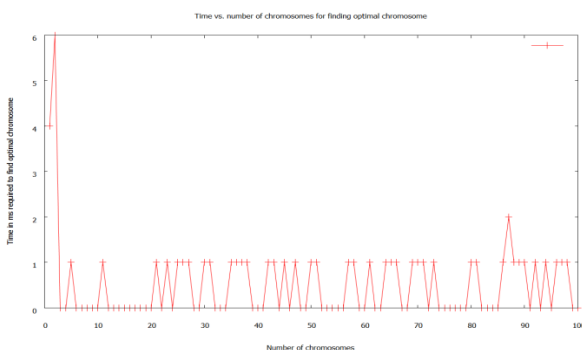


Figure 10. Diagram for representing simulation results of GA to find an optimal multicast tree

From this diagram, it is very easy to realize that time required to find the optimal multicast tree doesn't increase fast when the number of multicast trees increases. In particular, the time required for algorithm execution is sometimes decreased when the number of multicast trees is increased. In general, this time is very stable. The points are marked by red color.

3.3. Simulation of the fuzzy logic controller to choose cluster members

In this simulation, each node is assessed based on three metrics: bandwidth, hop count from this node to cluster node, energy remained. Simulation executes 2000 times with number of nodes from 20 to 50. These metrics of each node are randomly generated. The route from each member node to the cluster head consists of a number (from 1 to 10) of paths for increasing bandwidth and reducing latency. So the fuzzy logic controller assesses each node by using Eq. (12). The results of the simulation are represented in the Fig. 11.

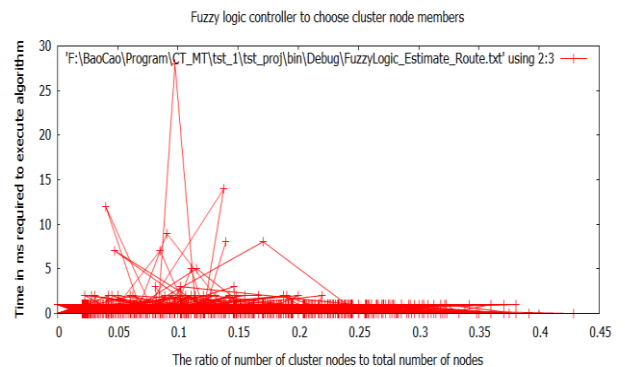


Figure 11. The simulation results of the fuzzy logic controller to choose cluster members

From the Fig. 11, it is easy to realize that the time required to execute the algorithm converges.

4. Conclusions

In order to increase route bandwidth and availability, the paper has mentioned fuzzy logic for clustering and genetic algorithm for finding optimal routes. Apply GA algorithms to find optimal route or multicast tree from the set of routes or multicast trees respectively. It requires the time to execute that is executable and this process is carried out at the same time with the process to detect routes. So when the process to detect route finishes, a little time later the GA process also finishes.

References

- [1] Fengyun Cao, Jaswinder Pal Singh. "Efficient Event Routing in Content-based Publish-Subscribe Service Networks" In Proc. IEEE Infocom 2004.
- [2] R. Kalaiarasi, Getsy S. Sara, S. Neelavathy Pari and D. Sridharan. Department of Electronics and Communication Engineering, MIT Campus Anna University Chennai, India. "PERFORMANCE ANALYSIS OF CONTENTION WINDOW CHEATING MISBEHAVIORS IN MOBILE AD HOC NETWORKS" International journal of computer science & information Technology (IJCSIT) Vol.2, No.5, October 2010.
- [3] Antonio Carzaniga and Alexander Wolf. "Forwarding in a Content-Based Network," In Proc. SIGCOMM 2003.
- [4] Antonio Carzaniga, Matthew J. Rutherford, and Alexander L. Wolf. Department of Computer Science. University of Colorado Boulder, Colorado 80309-0430 USA "A Routing Scheme for Content-Based Networking," In Proc. IEEE Infocom 2004.
- [5] Jianping Li, li@cni.ku-tokyo.ac.jp, Graduate School of Frontier Sciences, The University of Tokyo, Japan, Yasushi Wakahara, wakahara@nc.u-tokyo.ac.jp, Information Technology Center, The University of Tokyo, Japan "Time Slot Assignment for Maximum Bandwidth in a Mobile Ad Hoc Network" JOURNAL OF COMMUNICATIONS, VOL. 2, NO. 6, NOVEMBER 2007.
- [6] Huayi Wu, Xiaohua Jia, Computer School, Wuhan University, Luoyu Road 129, Wuhan 430079, China, Department of Computer Science, City University of Hong Kong, Hong Kong, "QoS multicast routing by using multiple paths / trees in wireless ad hoc networks, "Research supported by a grant FFCSA 2006 Elsevier BV.
- [7] Aisha-Hassan A. Hashim , Mohammad M. Qabajeh, Othman Khalifa and Liana Qabajeh, Department of Electrical and Computer Engineering, IIUM, Malaysia, "Review of Multicast QoS Routing Protocols for Mobile Ad Hoc Networks" IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.12, December 2008.
- [8] J. Abdullah, Dept of Communication Engineering, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia, "Effect of Maximum Node Velocity on GA-Based QOS Routing Protocol (QOSRGA) for Mobile Ad Hoc Network".
- [9] Yun-Sheng Yen, Yi-Kung Chan, Han-Chieh Chao, Jong Hyuk Park, Department of Electrical Engineering, National Dong Hwa University, Taiwan, ROC, "A genetic algorithm for energy-efficient based multicast routing on MANETs".
- [10] Nguyen Thanh Long, Nguyen DucThuy, Pham Huy Hoang, "Research on Innovating, Evaluating and Applying Multicast Routing Technique for Routing messages in Service-oriented Routing", Springer, ISBN: 978-1-936968-65-7, Volume Number 109, 2012.
- [11] Anjum A. Mohammed, Information Technology Department, College of Computer and Information Sciences, King Saud University, KSA, "Optimal Routing In Ad-Hoc Network Using Genetic Algorithm".