Vol. 44 No. 3 IPSJ Journal Mar. 2003

Regular Paper

A Genetic Algorithm Based QoS Routing Method for Multimedia Communications Over High-Speed Networks

LEONARD BAROLLI,[†] AKIO KOYAMA,[††] TAKUO SUGANUMA[†††]
and NORIO SHIRATORI[†††]

In order to support multimedia communication over high speed networks, it is necessary to develop routing algorithms which use more than one QoS parameter. This is because new services such as video on demand and remote meeting systems require better QoS. However, the problem of QoS routing is difficult and to find a feasible route with two independent path constraints is NP-complete. Therefore, QoS routing algorithms for high speed networks must be adaptive, flexible, and intelligent for efficient network management. In a previous work, we proposed a Genetic Algorithm (GA) based routing method. This method used only the delay time as a routing parameter. In this paper, we improve the previous work in three aspects. The new QoS routing method uses two QoS parameters for routing. Also, we implemented a new tree generating algorithm which generates and reduces the tree automatically. Furthermore, we carried out the simulations for different kinds of networks. The simulation results show that the proposed method has a better performance than a conventional GA based routing algorithm and can find better routes by using two QoS parameters.

1. Introduction

The high-speed networks are expected to support a wide range of multimedia applications. The requirement for timely delivery of multimedia data raises new challenges for the next generation broadband networks. One of the key issues is the Quality of Service (QoS) routing. It selects network routes with sufficient resources for the requested QoS parameters.

So far, many routing algorithms have been proposed. The routing strategies can be classified into three classes: source, distributed and hierarchical routing. Source routing algorithms are conceptually simple, but they suffer from scalability problem. Distributed routing algorithms are more scalable, but loops may occur, which make the routing to fail. Hierarchical routing has been used to cope with the scalability problems of source routing in large internetworks. The hierarchical routing retains many advantages of source routing and has some advantages of distributed routing because the routing computation is shared by many nodes. But, the network state is aggregate additional and gives some imprecision, which has a significant negative impact on QoS routing 1)~3).

High-speed transmission rates bring forward their specific issues influencing the network design. To cope with high-speed networks the routing algorithms should give a fast decision and must be adaptive, flexible, and intelligent for efficient network management 4)~6).

In Ref. 4), a Genetic Load Balancing Routing (GLBR) method is proposed. The effectiveness of the GLBR is shown compared with conventional Shortest Path First (SPF) and RIP algorithms. The GLBR method has a better behavior than SPF and RIP algorithms. But, the genetic operations are complicated. We proposed an Adaptive Routing method based on Genetic Algorithm (GA) (ARGA) 5),6). Performance evaluation via simulations shows that ARGA method has a faster routing decision compared with GLBR method. However, the ARGA method uses only the delay time as a parameter for routing.

In order to support multimedia communication over high speed networks, it is necessary to develop routing algorithms which use for routing more than one QoS parameter such as throughput, delay, and loss probability. This is because new services such as video on demand and remote meeting systems require better QoS. However, the problem of QoS routing is difficult, because the distributed applications have very diverse QoS constraints on delay, loss ratio, bandwidth, and multiple constraints often make the routing problem intractable. For example, finding a feasible route with two independent path constraints is NP-complete 1).

[†] Faculty of Engineering, Saitama Institute of Technology (SIT)
[††] Faculty of Engineering, Yamagata University
[†††] Research Institute of Electrical Communication, Tohoku University

The performance of a QoS routing algorithm can be seriously degraded if the state information is outdated. For this reasons, fast and efficient QoS algorithms are required for high speed networks.

In this paper, we propose a new GA based routing algorithm for multimedia communication called ARGAQ. The ARGAQ compared with ARGA algorithm has the following improvements. The ARGAQ uses two QoS parameters for routing. Also, we implemented a new tree generating algorithm which generates and reduces the tree automatically. With ARGA algorithm, we carried out the simulations only for a network with 20 nodes. However in this paper, we carry out many simulations for different kinds of networks.

The paper is organized as follows. In Section 2, we give GA cycle. In Section 3, we introduce the previous work. The proposed QoS routing method is treated in Section 4. The simulation results are discussed in Section 5. Finally, conclusions are given in Section 6.

2. GA Cycle

The GA cycle is shown in **Fig. 1**. At the beginning, an initial population of potential solutions is created as a starting point for the search. In the next stage, the performance (fitness) of each individual is evaluated with respect to the constraints imposed by the problem. Based on each individual's fitness, a selection mechanism chooses "parents" for the crossover and mutation operators. The crossover operator takes two chromosomes and swaps part of their genetic information to produce new chromosomes. The mutation operator introduces new genetic structures in the population by randomly modifying some of genes, helping the search algorithm to escape from local optimum. The offspring produced by the genetic manipulation process are the next population to be evaluated. GA can replace either a whole population or just its less fit members. The creation-evaluation-selection-manipulation cycle repeats until a satisfactory solution to the problem is found, or some other termination criteria are met [7].

3. Previous Work

In this section, we will explain ARGA and GLBR methods. In these methods, the most important factor to achieve efficient genetic operations is gene coding, because it has influence on the efficiency of genetic operations.

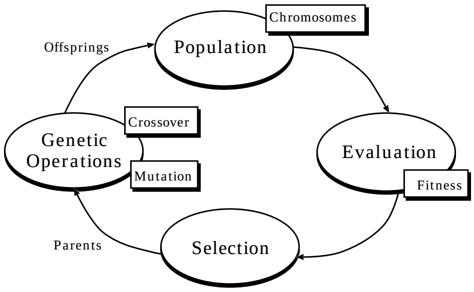

Fig. 1 GA cycle.

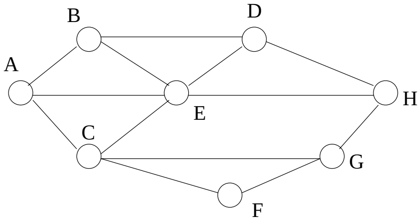

Fig. 2 A network example with 8 nodes.

In the GLBR method[4], the genes are put in a chromosome in the same order the nodes form the communication route, so the chromosomes have different size. If genetic operations are chosen randomly, the new offsprings of a population may be unsuitable individual populations. As a result, a communication route between two adjacent nodes may not exist and some complicated genetic operations should be carried out in order to find a new communication route. Also, because the individuals of a population have different sizes, the crossover operations are complicated.

In order to simplify the genetic operations of GLBR method, in the ARGA method, the network is expressed by a tree network and the genes are expressed by the tree junctions. By this coding method, the length of each chromosome is the same and the genetic operations are carried out in the tree junctions, so the searched routes always exist. Therefore, there is no need to check the validity of the searched routes as in GLBR method.

To explain this procedure, we use a small network with 8 nodes as shown in **Fig. 2**. Node A is the source node and node H is the destination node. All routes are expressed by the network tree model shown in **Fig. 3**. The shaded areas show the same routes from node C to H. In order to decrease the chromosome gene number, the network tree model of Fig. 3 is reduced as

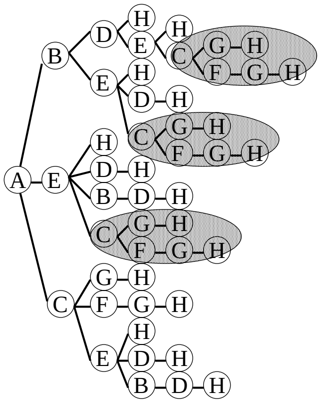

Fig. 3 Network tree model.

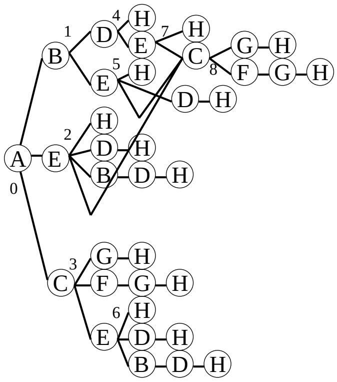

Fig. 4 Reduced network tree model.

0	1	2	3	4	5	6	7
A	B	D	E	C	F	G	H

Fig. 5 GLBR gene coding.

0	1	2	3	4	5	6	7	8
BEC	D E	H D B C	GFE	H E	HDC	HDB	H C	G F

Fig. 6 ARGA gene coding.

shown in **Fig. 4**. In the reduced network tree model, each tree junction is considered as a gene and the path is represented by the chromosome.

By using this gene coding method, the selected route always exists and the algorithm can avoid the routing loops. In **Fig. 5** and **Fig. 6** are shown the chromosomes of route A-B-D-E-C-F-G-H, for GLBR method and ARGA method, respectively. In ARGA method, the genes in a chromosome have two states "active" and "inactive". A gene is called active if the junction is in the route, otherwise the gene is in "inactive" state. The genetic operations are carried out in active genes.

4. Proposed Method

In this section, we will explain the tree generation algorithm, tree reduction algorithm, how the ARGAQ uses two QoS parameters for routing, and ARGAQ method operation.

4.1 Tree Generation Algorithm

First, the algorithm transforms the network in a tree network, which we call the basic tree network. Next, the basic tree network is reduced in the part where are the same routes. This is called the reduced tree network. The tree junctions of the reduced tree network are used as genes of ARGAQ method.

4.1.1 Basic Tree Network Generation

To generate the basic tree network, we represent the network model in a matrix form. When a link exists between two nodes the value of matrix element will be one, otherwise will be zero. The flowchart has two parts. The first part generates the paths from Source Node (RN) to Destination Node (DN), therefore the node check procedure is carried out in forward direction. The second part of the algorithm generates the branches of the tree network, thus the algorithm searches in backward direction. It should be noted that in a transmission route the same node name appears only once. This is done in order to avoid the routing loops. Also, when a route is searched, if there are links to nodes that have already been checked they will be ignored.

The flowchart of basic tree network algorithm is shown in **Fig. 7**. In order to illustrate this procedure, the network shown in Fig. 2 is represented in a matrix form as shown in Eq. (1). When there is a link between node A and node B, the values of M[A, B] and M[B, A] will be 1. Otherwise, the values will be 0.

$$M = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array} \begin{array}{c} \begin{array}{cccccccc} A & B & C & D & E & F & G & H \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array} \qquad (1)$$

Using this matrix the basic tree network is generated as follows.

- Select the node A as the tree root.
- In order to find the links from A, the search is started in row A from the column A.

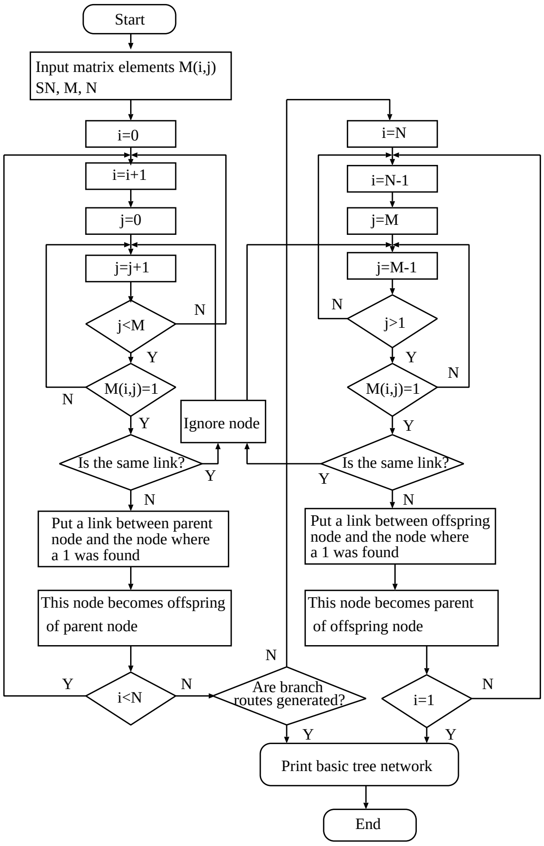

Fig. 7 Basic tree network flowchart.

- Because the search is done in order from the column A, the link to column B is found.
- The node B becomes the offspring of node A.
- In order to find the links from B, the search continues in row B starting again from column A.
- A link connected to node A is found, but node A has been already checked, so it is ignored.
- Next, a value 1 is found in the column D. This is an available link, because node D has not been checked yet.
- The node D becomes the offspring of node B and the search continues from node D.
- By searching in the same way the route "A-B-D-E-C-F-G-H" is found.
- In order to generate the branches of the tree network the search is started in backward direction from row G. The node H is the destination node, so the row H is not checked.
- Here, the node G which is the node before the last node H is checked whether this node has other links or not. For this reason, the algorithm goes back to G and the search continues.
- The search of G row already has finished until H column and there is not any column for checking. Therefore the search for row G finishes.
- The node F which is the node before the last node G is checked whether this node has other links or not. For this reason, the algorithm goes back to G and the search continues.
- The search of row F has been finished until column G. Therefore, the search restarts from column H.
- There are not links to be examined from column H, therefore the search for row F finishes.
- In the same way, the search continues from node C which is the node before the last node F.
- The column G is not examined yet. Therefore, the node G becomes offspring of node C and the search of row G starts from column A.
- Since the first available link of row G is found on column F, the node F becomes the offspring of node G and the search of column F continues.
- Since there are not available links from row F and this route cannot go up to node H, the node F is removed and it is not an offspring of node G.
- The search of row G is restarted from column G.
- An available link exists to H, so a new branch is found.

By repeating these procedure, all matrix elements are examined and the basic tree network is generated as shown in **Fig. 8**. In this research, we consider only the unicast routing. Therefore, the basic tree network has only one leaf (H).

4.1.2 Reduced Tree Network Algorithm

The reduced tree network is constructed by modifying the basic tree network. This procedure considers the comparison between nodes. As information for comparison are used the name of nodes (they are shown by alphabetic names) and the number of offspring nodes. The node name is written with a letter and a number (see Fig. 8). The number is used to distinguish the nodes with the same alphabetic

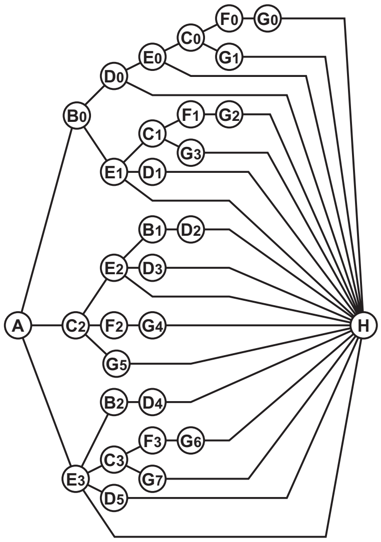

Fig. 8 Generated basic tree network.

name. The comparison is carried out in two directions: from the top to bottom and from the right to the left. When the nodes have the same name and offspring nodes the reduction process is carried out. Otherwise, the nodes are left unchanged and the algorithm moves to the next comparison. The algorithm stops when all nodes are compared and the nodes which have the same name and offsprings are reduced.

The flowchart of reduced tree network algorithm is shown in **Fig. 9**. In order to illustrate this procedure, we consider the reduction of generated basic tree network shown in Fig. 8. The reduced tree network is generated as follows.

- The search starts from the parents of node H.
- The sequence is checked in order from the top to the bottom of Fig. 8. For example, G0, G1, E0, D0, and so on.
- First, two parents of node H which are G0 and G1 are selected.
- The name of nodes and the number of offsprings are compared. In the case of nodes G0 and G1, the name of nodes is the same and they have the same offsprings.
- For this reason, G1 is replaced with G0 and removed from the tree.
- Next, G0 and E0 are compared. In this case, the node names are different, so they are left unchanged and go to the next comparison.
- When the comparison between G0 and other parents of H has been finished, the algorithm assigns the name G for all nodes which were replaced with G0.
- After this operation, the reduction of node G which is the parent of node H is finished.
- Since the name G represents all nodes which were replaced with G0, all parents of G are also the parents of G0. In this case, the node G0 represent all nodes from G1 to G7.
- Next the reduction continues with the parents of node G0 in the right-left direction.
- By continuing the reduction procedure for parents of node G0, nodes F1, F2, F3 are replaced with F0 and nodes C1 and C3 with C0.
- When the parent nodes are not reduced, then the algorithm continues with next comparison.
- When the comparison of all parent nodes has finished, the reduction of the basic tree network is finished and the reduced tree

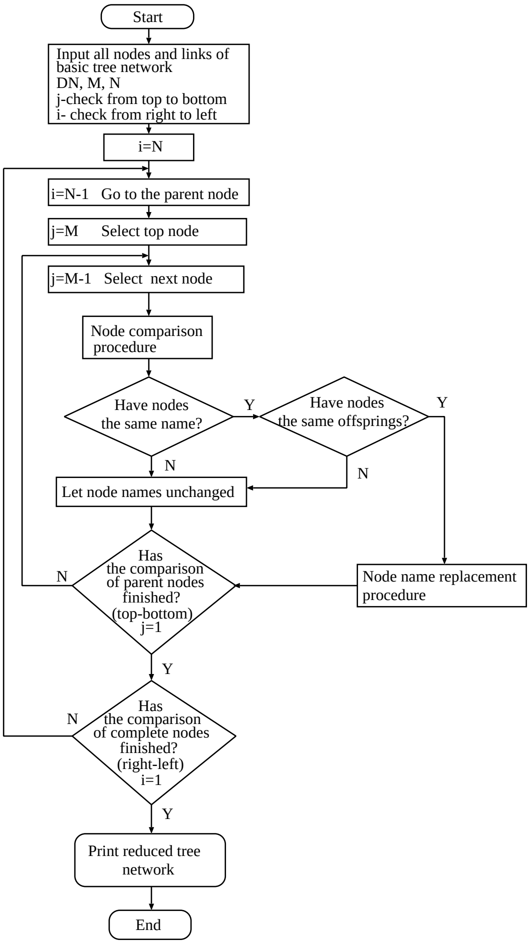

Fig. 9 Reduced tree network flowchart.

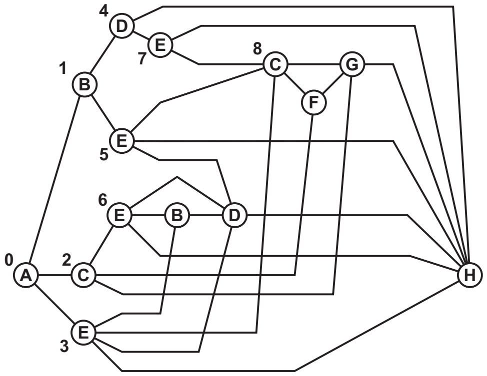

Fig. 10 Generation of reduced tree network.

network is generated.

After generation of the reduced tree network, the tree junctions are coded as genes and a number is assigned to each tree junction, which corresponds to the chromosome locus. The reduced tree network generated from Fig. 8 is shown in **Fig. 10**.

4.1.3 Two QoS Routing Parameters

In ARGA and GLBR methods, the best route was decided considering only the delay time parameter. In ARGAQ method two parameters: the Delay Time (DT) and Transmission Success Rate (TSR) are considered as QoS parameters. The DT means the time it takes a packet to go from one node to another one. The TSR shows the rate of correctly transmitted packets (without loss). The packet loss may be caused by the overflow in the queue or from some network troubles.

Let consider a network as shown in **Fig. 11**. The node A is the source node and node B is the destination node. Let node A sends to node B 10 packets. The total TSR value for Fig. 11 (a) and Fig. 11 (b) is calculated by Eq. (2) and Eq. (3), respectively.

$$10 \times 0.9 \times 0.9 \times 0.9 \times 0.9 = 6.561 \qquad (2)$$

$$10 \times 1.0 \times 1.0 \times 0.6 \times 1.0 = 6.000 \qquad (3)$$

The best route in this case is that of Fig. 11 (a), because the total TSR is higher compared with that of Fig. 11 (b).

Let consider another example, when the values of DT and TSR are considered as shown in **Fig. 12**. The value of T parameter is decided as follows.

$$T = \frac{\sum_{i=1}^{n} DT_i}{\prod_{i=1}^{n} TSR_i} \qquad (4)$$

where n is the number of links in a path.

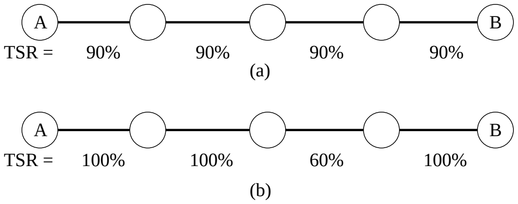

Fig. 11 An example of TSR calculation.

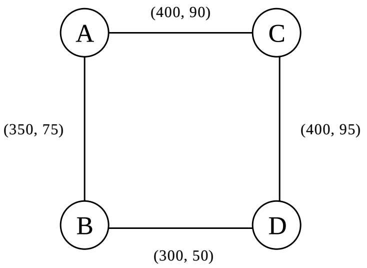

Fig. 12 A network example for QoS routing.

When node A wants to communicate with node D, there are two possible routes: "A-B-D" and "A-C-D". The T value for these routes is calculated by Eq. (5) and Eq. (6), respectively.

$$T_{A-B-D} = \frac{350 + 300}{75 \times 50} = \frac{650}{3750} = 0.1733 \quad (5)$$

$$T_{A-C-D} = \frac{400 + 400}{90 \times 95} = \frac{800}{8550} = 0.0468 \quad (6)$$

The delay time of "A-B-D" route is lower than "A-C-D" route, but the T value of "A-C-D" route is lower than "A-B-D", so "A-C-D" route is the better one. This shows that a good candidate route can be found when two QoS parameters are used for routing.

4.2 ARGAQ Method Operation

The ARGAQ method is a source-based routing mechanism and uses two QoS parameters for routing. When a node wants to transmit information to a destination node, this node becomes the source node. The ARGAQ method transforms the network in a tree network with the source node as the root of tree. After that, the tree network is reduced in the parts where are the same routes. By using the tree model, the routing loops can be avoid, therefore the algorithm doesn't lose searching time in the routing loops. Also, by reducing the tree network, the chromosome length is shorten so the genetic operations become simple.

After the reduction of the tree network, the tree junctions are coded as genes. The genes in a chromosome have the information of the adjacent nodes. Because, the individual and chromosome are the same, the route is represented by the chromosome and the population is a collection of routes.

After the gene coding, the ARGAQ method starts the genetic operations. First, an initial population is selected. In the selected population, the ranking selection model is used to select two individuals in order to carry out the genetic operations. The ranking model ranks each individual by their fitness. The rank is decided based on the fitness and the probability is decided based on the rank. The individual fitness is based on T value. When T value is small, the individual fitness is high.

The genetic operations are the crossover and mutation. The ARGAQ method uses the single point crossover, because simple operations are needed to get a fast response. In the mutation operation, the genes are chosen randomly in the range from zero up to mutation probability $p\_mutation \leq \frac{1}{\ell}$, where l is the chromosome length.

After the crossover and mutation, the elitist model is used. Based on the elitist model the individual which has the highest fitness value in a population is left intact in the next generation. Therefore, the best value is always kept and the routing algorithm can converge very fast. The offsprings produced by the genetic operations are the next population to be evaluated. The genetic operations are repeated until the initialized generation size is achieved or a route with a minimum T value is found.

The route selection in ARGAQ method is based on T value, which is the ratio of DT with TSR. T is used as a fitness function to evaluate the selected individuals (routes). By minimizing the T value, the DT value is minimized and the TSR value is maximized. This means that a packet from node A to node H is transmitted with a small delay and a high transmission success rate.

We intent to use the proposed method for small scale networks. For large scale networks, we have implemented a distributed routing architecture based on cooperative agents[8]. In this architecture, the ARGAQ method will be used for intra-domain routing.

5. Simulation Results

In this paper, we carried out many simulations for different kinds of networks with different number of nodes, routes and branches as shown in **Table 1**. We implemented a new routing algorithm based on GLBR method with two QoS parameters and called this method GLBRQ method. By simulations, we compare the results of ARGAQ method with GLBRQ and ARGA methods.

Table 1 Number of nodes, routes and branches.

Nodes	20	30	35
Routes	725	11375	23076
Branches	33	85	246

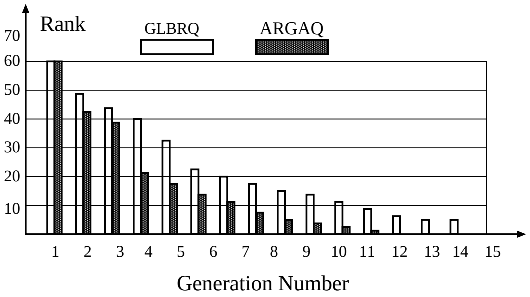

Fig. 13 Performance behavior of ARGAQ and GLBRQ methods.

First, we set in a random way the DT and TSR in each link of the network. Next, we calculate the value T, which is the ratio of DT with TSR. This value is used to measure the individual fitness. For simulations, we consider that a congestion state happens in a communication link, and we investigate the speed at which each algorithm finds a new route. The genetic operations are repeated until a route with a small T value is found or the initialized generation size is achieved. For the sake of comparison, we use the same parameters and the population size is selected the same.

Performance behavior of ARGAQ and GLBRQ methods for a network with 20 nodes is shown in **Fig. 13**. This figure shows the rank versus the generation number. The rank is decided based on the value of fitness function T. When the rank is low the fitness value is low. This means that the selected route has a low delay and a high transmission rate. The average rank value of ARGAQ method is lower than average rank value of GLBRQ method for the same generation number. This shows that GLBRQ method needs more genetic operations to find a feasible route. Therefore, the search efficiency of ARGAQ method is better than GLBRQ method.

In **Table 2** are shown the simulation results for the time needed for one generation. The

Table 2 Time needed for one generation (ms).

Nodes	Method	Average	Max	Min
20	GLBRQ	75.98	350	*
	ARGAQ	4.80	20	*
30	GLBRQ	82.57	410	*
	ARGAQ	6.41	30	*
35	GLBRQ	89.61	400	*
	ARGAQ	12.99	50	*

Table 3 Performance for different parameters.

Nodes	Method	Rank	Gen	Fail	Ref
20	GLBRQ	5.50	33.50	6	54.32
	ARGAQ	5.62	8.00	0	26.30
30	GLBRQ	8.80	69.94	14	123.12
	ARGAQ	6.44	53.30	8	100.94
35	GLBRQ	6.12	55.52	6	103.84
	ARGAQ	5.38	28.72	0	65.62

Table 4 Comparison between ARGAQ and ARGA methods.

Method	TA	DA	TSRA	GSA	GOTA
ARGAQ	4.47	10.52	9.36	9.00	85.78
ARGA	—	4.66	70.60	8.33	69.04

ARGAQ method is faster than GLBRQ method for all kinds of networks. This is because the ARGAQ method has a better gene coding than GLBRQ method.

In **Table 3**, the Rank, Gen, Fail and Ref have the following meaning: Rank — the average rank to find a new route; Gen — the average number of generations to find a new route; Fail — the rate that a new route was not found (%); Ref — the average number of individuals refereed in one simulation. Considering the results in Table 3, the ARGAQ can find a new route by using less generations than GLBRQ method. This means that the search efficiency of ARGAQ method is better than GLBRQ method. For the network with 30 nodes, the failure for GLBRQ method was about 14%. In the case of networks with 30 and 35 nodes, the algorithms sometime did not find a new route, but the failure rate for the network with 30 nodes is about 2 times more than the network with 35 nodes. This shows that by increasing the network scale the ARGAQ method shows better behavior than GLBRQ method.

In **Table 4**, we show the simulation results of ARGAQ and ARGA methods. The TA means the average rank value of T parameter, DA means the average rank value of delay, TSRA means the average rank value of TSR parameter, GSA means the average value of generation number, and GOTA means the average value of genetic processing time. The genetic operations processing time of ARGA method is better than ARGAQ method. However, the difference is very small (see parameter GOTA). In the case of ARGAQ method, when two QoS parameters are used, both DA and TSRA values are optimized. However, in the case of ARGA method only one QoS parameter is used. Thus, only DA value is optimized, the TSRA value is large. Therefore, the selected route is better from the QoS point of view when we use two QoS parameters than using only one QoS parameter.

6. Conclusions

In this paper, we proposed a new QoS routing approach for multimedia applications using GA. The performance evaluation via simulations shows that the proposed method has better behavior than GLBRQ method. From the simulation results, we conclude:

- ARGAQ method has a faster response time compared with GLBRQ method;
- ARGAQ method has simple genetic operations;
- ARGAQ method can support two QoS parameters;
- ARGAQ method can find better QoS routes than ARGA method.

In the future, we would like to extend our study by implementing the proposed method in a parallel GA in order to find routes which satisfy more than two QoS parameters.

References

1) Chen, S. and Nahrstedt, K.: An Overview of Quality of Service Routing for Next-Generation High-Speed Networks: Problems and Solutions, *IEEE Network*, Special Issue on Transmission and Distribution of Digital Video, Vol.12, No.6, pp.64–79 (1998).

2) Baransel, C., Dobosiewicz, W. and Gburzynski, P.: Routing in Multihop Packet Switching Networks: Gb/s Challenge, *IEEE Network*, Vol.9, No.3, pp.38–60 (1995).

3) Lee, W.C., Hluchyj, M.G. and Humblet, P.A.: Routing Subject to Quality of Services Constraints in Integrated Communications Networks, *IEEE Network*, Vol.9, No.4, pp.46–55 (1995).

4) Munetomo, M., Takai, Y. and Sato, Y.: An Adaptive Routing Algorithm with Load Balancing by a Genetic Algorithm, *Trans. Information Processing Society of Japan*, Vol.39,

No.2, pp.219–227 (1998).
5) Barolli, L., Koyama, A., Motegi, S. and Yokoyama, S.: Performance Evaluation of a Genetic Algorithm based Routing Method for High-speed Networks, *Trans. IEE Japan*, Vol.119-C, No.5, pp.624-631 (1999).
6) Barolli, L., Koyama, A., Yamada, T. and Yokoyama, S.: An Intelligent Policing-Routing Mechanism Based on Fuzzy Logic and Genetic Algorithms and Its Performance Evaluation, *IPSJ Journal*, Vol.41, No.11, pp.3046-3059 (2000).
7) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc. (1989).
8) Barolli, L., Koyama, A., Yamada, T., Yokoyama, S., Suganuma, T. and Shiratori, N.: An Intelligent Routing and CAC Framework for Large Scale Networks Based on Cooperative Agents, *Computer Communications*, Vol.25, No.16, pp.1429-1442 (2002).

(Received May 20, 2002)
(Accepted December 3, 2002)

Leonard Barolli was born in Bilisht, Albania. He received B.E. and Ph.D. degrees from Tirana University and Yamagata University in 1989 and 1997, respectively. From April 1997 to March 1999, he was a JSPS Post Doctor Fellow Researcher at the Department of Electrical and Information Engineering, Yamagata University. From April 1999 to March 2002, he worked as a Research Associate at the Department of Public Policy and Social Studies, Yamagata University. From April 2002, he is an Assistant Professor at the Department of Computer Science, Saitama Institute of Technology (SIT). Dr. Barolli has published about 70 papers in referred Journals and International Conference proceedings. He is an Editor of the IPSJ Journal and has served as an Editor for some Special Issues. Dr. Barolli has been a PC Member of many International Conferences and is a PC Co-Chair of AINA2003. His research interests include network traffic control, fuzzy control, genetic algorithms and agent-based systems. He is a member of SOFT, IPSJ, and IEEE.

Akio Koyama was born in Yonezawa. He received B.E. and Ph.D. degrees from Yamagata University in 1987 and 1998, respectively. From April 1981 to March 1999, he was working as a technical staff at the Faculty of Engineering, Yamagata University. From April 1999 to March 2002, he was an Assistant Professor at the Department of Computer Software, the University of Aizu. From April 2002, he is working as an Associate Professor at the Department of Informatics, Yamagata University. His current research interests include network agent systems, LAN protocols, distance learning systems, and network congestion control. He is a member of IEEE Computer Society, IPSJ and IEICE.

Takuo Suganuma was born in Nagano prefecture. He received B.E., M.S. and Ph.D. degrees from Chiba Institute of Technology in 1992, 1994 and 1997, respectively. From April 1997, he is working as a Research Associate at the Research Institute of Electrical Communication (RIEC), Tohoku University. His research interests include agent-based computing and design methodology for distributed systems. He is a member of IPSJ, IEICE and IEEE.

Norio Shiratori, after receiving his doctorate degree at Tohoku University, he joined the Research Institute of Electrical Communication (RIEC) and is now a professor. He has been engaged in research on distributed processing systems and flexible intelligent networks. He received the IPSJ Memorial Prize-Winning Paper Award in 1985, Telecommunications Advancement Foundation Incorporation Award in 1991, the Best Paper Award of ICOIN-9 in 1994, the IPSJ Best Paper Award in 1997, etc. He has been named a Fellow of the IEEE for his contributions to the field of computer communication networks. He is a member of IPSJ, IEICE and IEEE.