Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Two-stage evolutionary algorithm for dynamic multicast routing in mesh network

Li ZHU[†1,2], Zhi-shu LI[1], Liang-yin CHEN[1], Yan-hong CHENG[1]
([1]School of Computer Science, Sichuan University, Chengdu 610065, China)
([2]Network Center, Chengdu Sport University, Chengdu 610041, China)
[†]E-mail: zhulicdsu@126.com

Received Sept. 25, 2007; revision accepted Jan. 7, 2008; published online May 10, 2008

Abstract: In order to share multimedia transmissions in mesh networks and optimize the utilization of network resources, this paper presents a Two-stage Evolutionary Algorithm (TEA), i.e., unicast routing evolution and multicast path composition, for dynamic multicast routing. The TEA uses a novel link-duplicate-degree encoding, which can encode a multicast path in the link-duplicate-degree and decode the path as a link vector easily. A dynamic algorithm for adding nodes to or removing nodes from a multicast group and a repairing algorithm are also covered in this paper. As the TEA is based on global evaluation, the quality of the multicast path remains stabilized without degradation when multicast members change over time. Therefore, it is not necessary to rearrange the multicast path during the life cycle of the multicast sessions. Simulation results show that the TEA is efficient and convergent.

Key words: Dynamic multicast, Routing, Encoding, Quality of Service (QoS), Evolution, Genetic algorithm (GA)
doi:10.1631/jzus.A071519 **Document code:** A **CLC number:** TP18; TP393

INTRODUCTION

With the high development of network technology, multimedia applications have become dominant in mesh networks. The multimedia communication is sensitive to the quality of service (QoS) along transmission routes and occupies a mass of network bandwidth, which in turn brings big challenges to network resources and transmission approaches. Multicast focuses on sharing the network transmission and enhancing the utilization of network resources, and thus this is a key in minimizing the total cost of network transmissions.

Shortest path tree (SPT) (Dalal and Metcalfe, 1978) is an original algorithm for multicast routing, which minimizes the path cost from the source to each multicast member with time complexity $O(n^2)$. As shared paths are not considered, the network resources cannot be utilized in an optimized way in the SPT. The objective of the minimum Steiner tree is to minimize the total cost of the multicast path. Kompella (1993) proposed a heuristic Steiner tree algorithm which was improved by Erlangung (2003) with time complexity $O(mn^3)$. SPT and Steiner tree algorithms are suitable for static multicast, however, multicast mechanism operates actually on dynamic networks, and multicast members vary with time.

Waxman (1988) proposed a weighted greedy algorithm for dynamic multicast for the first time, which allows nodes joining in or off the multicast group during a connection. Since the quality of the multicast tree deteriorates over time when multicast members change frequently, Raghavan *et al.*(1999) proposed an algorithm with a quality factor (QF) to measure the quality of the dynamic multicast tree. When QF drops below a particular threshold, the algorithm rearranges the multicast tree. Chen and Xu (2006) proposed a balancing factor to measure and trigger the multicast tree rearrangement. However, rearrangement of the multicast tree may result in a large service disruption which may not be tolerated, especially by QoS multicast sessions. Therefore, having received a significant attention, the rearrangement needs further researches.

Wu *et al*.(2006) proposed a heuristic local optimum algorithm for QoS aware overlay multicast, adopting self-organized, distributed and tree-first strategies. Viswanath *et al*.(2006) used flooding to explore multicast routing for a mobile network. Zhou *et al*.(2007) proposed a dynamic zone-based multicast routing protocol. Wu, Viswanath and Zhou *et al*. solved the dynamic multicast problems in a distributed way without rearrangement for the existing multicast tree. However, since the algorithms are based on the local optimization, the quality of the multicast tree cannot be retained over time.

Genetic evolutionary algorithms were used in multicast routing in recent years. Fu and Li (2005) proposed a genetic algorithm (GA) for multicast routing with cost-delay-jitter QoS constraints. Liu *et al*.(2006) proposed a GA for multicast in wireless ad hoc networks. Garrozi and Araujo (2006) proposed a multi-objective optimization GA for multicast routing. Yue *et al*.(2007) proposed a chaotic GA, adding chaotic perturbation in GA to inhibit the premature multicast routing. Most GAs are encoded in a complicated tree style, e.g., Fu and Li (2005) and Vijay *et al*.(2005) adopted tree-based encodings, and Liu *et al*.(2006) used an indexed tree encoding. Garrozi and Araujo (2006) proposed a novel $M\cdot L$ link encoding, however, the proposed algorithm is not target for dynamic multicast. Sun *et al*.(2007) proposed a parallel quantum GA for QoS multicast routing, using quantum bits to express genes, but they did not provide any specific routing expression in quantum genes.

Focusing on the dynamic multicast and the quality of multicast paths, this paper presents a Two-stage Evolutionary Algorithm (TEA). First, we propose a simple link-based encoding method, link-duplicate-degree encoding, which can encode a multicast path in the link-duplicate-degree and decode the path as a link vector easily. Then based on the link-based encoding we propose a two-stage algorithm, including unicast routing evolution and multicast path composition, for multicast routing creation. We present a dynamic multicast maintenance algorithm for adding nodes to the multicast group or removing nodes from the group, and a repairing algorithm for the dynamic network. Finally, we study the performance of the TEA in simulation experiments and verify the efficiency of the algorithm.

Multicast routing model

Let graph $G(V, E, \omega)$ represent an interconnected network, and N and L denote the numbers of nodes and edges in G, respectively; $V=\{v_1, v_2, \ldots, v_N\}$ is a set of nodes, denoting the routers; $E=\{e_1, e_2, \ldots, e_L\}$ is a set of edges, denoting the links; $\omega=\{\omega_1, \omega_2, \ldots, \omega_L\}$ is the weight set of graph G which represents QoS constraints on nodes and links. Given a source node s ($s\in V$) and a set of multicast destinations M ($M\subseteq V$), we call 'multicast nodes' are given, and the multicast problem can be described as follows:

A multicast path T has to be found which originates from the source s, and covers every multicast node m ($m\in M$), to minimize the total cost and satisfy the QoS requirements of each multicast node. Let p represent a unicast routing to the multicast node m, $m\in M\subseteq V$, then the multicast routing problem can be expressed as:

$$\min\left\{cost(T)=\sum\nolimits_{i=1}^{L}\omega_i x_i\right\}, \tag{1}$$

$$\text{for } \omega_i\in\omega,\quad x_i=\begin{cases}1, & e_i\in T,\\ 0, & \text{otherwise,}\end{cases}$$

$$delay(p)=\sum_{v\in p}nd(v)+\sum_{e\in p}ld(e)\le dl_0, \tag{2}$$

$$bandwidth(p)=\min_{e\in p}\{bandwidth(e)\}\ge bw_0, \tag{3}$$

$$jitter(p)=\sum_{v\in p}jitter(v)\le jt_0, \tag{4}$$

$$loss(p)=\sum_{v\in p}loss(v)\le ls_0, \tag{5}$$

where nd and ld denote delay on nodes and on links, respectively; dl_0, bw_0, jt_0 and ls_0 represent the constraint thresholds corresponding to delay, bandwidth, jitter and loss rate, respectively; $v\in V$, $e\in E$.

TWO-STAGE EVOLUTIONARY ALGORITHM

As the total cost of a multicast path is associated with the unicast routings of multicast nodes, the TEA decomposes the process of creating the multicast path into two stages. First, the TEA creates the unicast routing population, and then composes the multicast path with unicast routings. Corresponding to the two stages, the TEA adopts two kinds of encoding methods, i.e., string and link set encodings, and two kinds of evaluations, i.e., affinity and fitness evaluations. The affinity evaluation is used for unicast routings to get predominant routings in immune space and the fitness evaluation is used for the optimization of the multicast path.

Multicast encoding

The TEA encodes the unicast routing and the multicast path separately.

Unicast routing encoded in string takes the node identifiers as genes when the routing passes those nodes and lines them in order. $p_i^k=\{n_1, n_2, ..., n_l\}$ represents the routing to the ith multicast node in the kth multicast path, where n_j (j=1, 2, ..., l) represent the node IDs, and l is the number of nodes in the unicast routing. An array of unicast routings in a multicast group constructs a routing matrix.

Before describing the multicast path encoding, we introduce the following definitions:

Definition 1 Linkage matrix expresses linkage relationship between pairs of nodes in a network. In the linkage matrix, each row corresponds to a node and associates with other nodes on columns. A linkage matrix is shown below:

$$\begin{array}{cc} & \begin{array}{cccccc} 1 & 2 & \cdots & j & \cdots & n \end{array} \\ \begin{array}{c} 1\\ 2\\ \vdots\\ i\\ \vdots\\ n \end{array} & \left[\begin{array}{cccccc} & & & & & \\ & \ddots & & \vdots & & \\ & & & & & \\ & \cdots & & \boldsymbol{m}_{ij} & & \\ & & & & \ddots & \\ & & & & & \end{array}\right] \end{array},$$

where $\boldsymbol{m}_{ij}$=[*cost bw delay*] is a link vector between nodes i and j. It can be ascertained by $\boldsymbol{m}_{ij}$ whether a linkage exists between nodes i and j. If $\boldsymbol{m}_{ij}.bw>0$, then the answer is yes; otherwise, no.

Definition 2 Link-duplicate-degree recodes the number of unicast routings passing through a particular link in $G(V, E, \omega)$. Let i denote the link ID, λ_i the link-duplicate-degree of link i, $\forall i$, $\exists\lambda_i\in\mathbb{N}_+$, if the ith link is passed by a routing j (j=1, 2, …, m), then $\lambda_i=m$; otherwise, $\lambda_i=0$.

A multicast path is encoded in a link set with the length which equals the number of links in a network. Each gene takes the link-duplicate-degree which corresponds to the link ID in sequence, the link set $S_{\text{link}}=\{\lambda_1, \lambda_2, ..., \lambda_L\}$. This encoding is easy to be decoded as a link vector $[l_1, l_2, \ldots, l_n]$, $n<L$.

Multicast path creation

The process of multicast routing creation is divided into two stages, i.e., creating unicast routing populations and composing the multicast path.

1. Unicast routing population creation

The TEA creates a unicast routing population for each multicast node and adopts GAs to promote the diversity of populations (Zhu *et al.*, 2007).

To further intensify the diversity, immune theory is used in evolutionary processes to restrain the individual similarity of populations in the Hamming method. Let $\{n_1, n_2, \ldots, n_l\}$ denote a routing, and then the distance between the ith and jth routings is defined as

$$D_{ij}=\sum_{k=1}^{N}\delta_{ij}^k,\quad \delta_{ij}^k=\begin{cases}1, & n_i^k\neq n_j^k,\\ 0, & \text{otherwise},\end{cases} \tag{6}$$

where n_i^k is the kth node gene of the ith routing.

The TEA drops those routings which do not meet the QoS requirement, and creates a routing population for each multicast node.

2. Multicast path composition

First, the TEA takes an initial node from the multicast set M at random. From unicast routing population of the initial node, the TEA chooses a unicast routing with the minimal cost as an initial path of the multicast group. Then, the TEA takes a second node from M at random and composes the multicast path with the initial multicast path and a unicast routing in the population of the second node tentatively. The TEA chooses the partial multicast path with the best fitness, and then updates the S_{link}. This process is repeated until every multicast node has been added to the multicast path. Finally, a population of multicast paths is created and the path with the best fitness on evaluation is chosen as the result of the TEA. The composition algorithm can be described as follows:

```
Take a multicast node $n_0$, $n_0\in M$;
Choose a unicast routing of $n_0$ with min(cost) as $r_0$;
Partial-path $P:=r_0$;
For each node $n_i$, $n_i\in M$, $n_i\neq n_0$
    for each routing $r_j$ in a population of $n_i$
        partial-path $p_j$:=compose-path($P$, $n_i$, $r_j$);
```

```
        fitness_j:=pre_evaluation of p_j;
    end
    partial-path P:=p_j with min(fitness);
End
Path:=partial-path P;
Update the link-duplicate-degree in S_link.
```

An example of multicast path composition is shown in Fig.1. The initial path for a multicast is 1→5→6, and the *link-duplicate-degree* for links 1→5 and 5→6 is 1. The second routing 1→5→6→7 is selected to be composed with the initial path, and the composed result is still 1→5→6→7, but the *link-duplicate-degree* for links 1→5 and 5→6 becomes 2.

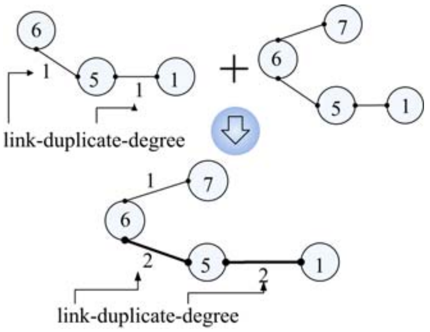


Fig.1 An example of multicast path composition

Multicast evaluation

There are two criteria to evaluate the quality of multicast paths. The first is the total cost of the multicast path, and the second is the cost of unicast routings to the last multicast node.

Let *mFit* represent the fitness of a multicast path, and $cost_i$ the cost of the *i*th link, then the multicast path evaluation is defined as

$$mFit(G,M) = M\Big/\sum_{k=1}^{M}\sum_{i=1}^{L} a_i^k cost_i + 1\Big/\sum_{i=1}^{L} b_i cost_i\,, \quad (7)$$

where

$$a_i^k = \begin{cases} 1, & \text{if the } i\text{th link} \in \text{the } k\text{th routing}, \\ 0, & \text{otherwise}, \end{cases}$$

$$b_i = \begin{cases} 1, & \text{if the } i\text{th link} \in \text{a new routing}, \\ 0, & \text{otherwise}, \end{cases}$$

a_i^k indicates whether the *k*th unicast routing passes through the *i*th link or not, $\sum_{i=1}^{L} a_i^k cost_i$ is the cost of the *k*th unicast routing. $\sum_{i=1}^{L} b_i cost_i$ is the cost of a unicast routing to the last multicast node, and $1\big/\sum_{i=1}^{L} b_i cost_i$ makes the cost of unicast routings as small as possible when different multicast paths take the equal total cost.

For the source-based multicast, we adopt Eq.(8) to stabilize the multicast backbone with a single source. For the core-based multicast, we design Eq.(9) to distribute the multicast paths along the network to balance the network loads.

$$mFit(G,M) = \left(M\Big/\sum_{k=1}^{M}\sum_{i=1}^{L} a_i^k cost_i + 1\Big/\sum_{i=1}^{L} b_i cost_i\right)\cdot \sum_{i=1}^{L} c_i \Big/ \sum_{i=1}^{L} d_i\,, \quad (8)$$

$$mFit(G,M) = \left(M\Big/\sum_{k=1}^{M}\sum_{i=1}^{L} a_i^k cost_i + 1\Big/\sum_{i=1}^{L} b_i cost_i\right)\cdot \left(1-\sum_{i=1}^{L} c_i \Big/ \sum_{i=1}^{L} d_i\right), \quad (9)$$

where $c_i = \begin{cases} 1, & \text{if } \lambda_i \ge 3, \\ 0, & \text{otherwise}, \end{cases}$ $d_i = \begin{cases} 1, & \text{if } \lambda_i \ge 1, \\ 0, & \text{otherwise}, \end{cases}$ $\sum_{i=1}^{L} c_i$ represents the number of backbone links, $\sum_{i=1}^{L} d_i$ is the number of links, and $\sum_{i=1}^{L} c_i \big/ \sum_{i=1}^{L} d_i$ means the backbone ratio of the multicast path.

DYNAMIC MULTICAST ALGORITHM

Multicast path changes with dynamic memberships. We propose dynamic algorithms to add nodes to or remove nodes from the multicast group.

Algorithm for adding nodes to the multicast group

The algorithm for adding nodes to the multicast group performs in an incremental way. When a join request is received from a node, the algorithm gets the unicast routing population of this node first and composes new multicast paths with the routings, evaluates the fitness of the composed paths, and then takes the path with the best fitness. At last, the algorithm updates S_{link} of the multicast group.

Suppose that node *k* requests to join a multicast group. Then the algorithm is as follows:

```
$P_k$:=create a routing population of node $n_k$;
For each routing $r_j$ in $P_k$
    Path $p_j$:=compose-path(path, $n_k$, $r_j$);
    fitness$_j$:=pre_evaluation of $p_j$;
End
Multicast_Path:=Path with min(fitness);
Increase 1 to the related link-duplicate-degree in $S_{\text{link}}$;
Add unicast-routing(k) to the routing matrix of the path.
```

Algorithm for removing nodes from the multicast group

Algorithm for removing nodes from the multicast group only needs to decrease 1 from the related *link-duplicate-degree* in S_{link} and delete the unicast routing from the routing matrix.

Let node k leave a multicast group. Then the algorithm is described as:

```
Get unicast-routing(k) from the path routing matrix;
Transform unicast-routing(k) to the link vector;
Decrease 1 from the related link-duplicate-degree in $S_{\text{link}}$;
Delete unicast-routing(k) from the routing matrix.
```

Repairing algorithm for the dynamic network

Node and link failures may happen in a network incidentally and the multicast path may break. On messages transferred by routing protocols, such as the link state protocol, the repairing algorithm can correct the multicast path and performs in the following way:

```
Get failure links;
Update the linkage matrix related with the failure links;
For each failure link
  get $\lambda_i$ from $S_{\text{link}}$, $i<L$;
  if $\lambda_i \le 0$
    no repair is needed;
  else
    scan the unicast routing matrix;
    find node m affected by failure links, $m \in M$;
    remove node m from the multicast group;
    add node m to the multicast group again;
End
```

SIMULATION EXPERIMENTS

The simulation experiments run in Matlab 7.1 on a personal computer with 1.6 GHz Intel Pentium CPU and 512 M memory. We take four kinds of network topologies created by the BRITE network topology generator (Medina *et al*., 2001) in the Waxman model, with the parameters listed in Table 1. The multicast nodes are less than 18% of the total nodes in the network. As for the GA parameters, the mutation probability p_{m}=0.1, and the crossover probability p_{c}=0.9. If there is no additional note, we take the population size *popsize* as N/2, N/3, N/4 corresponding to the network with 33, 66, 100 nodes respectively, and we take the memory size c_{m} value in the same way. For convenient comparison, we set the cost value of each link as 1 except for the first network, and take the first node as the multicast source. Each experiment was repeated more than 30 times.

Table 1 Parameters for the multicast network topologies

Nodes	Directed links	Multicast nodes	α	β
8	24	2	–	–
33	132	6	0.9	0.7
66	264	12	0.9	0.7
100	400	18	0.9	0.7

α, β are parameters of edge density in the range (0, 1). Larger values of α cause higher edge densities, while small values of β increase the density of short edges relative to longer ones

First, we test the multicast cost and CPU time and make comparison among the TEA, the SPT and Garrozi algorithms. The experimental results are shown in Figs.2 and 3.

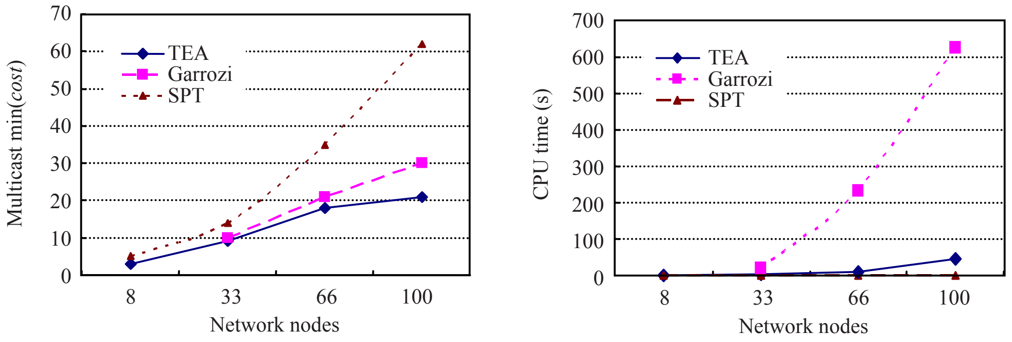
In Fig.2a, the total cost of the TEA is lower than those of the SPT and Garrozi algorithms. Enlarging the network scale, the total cost of the TEA grows slowly, while the total cost of the SPT goes up quickly. This result shows that the TEA makes the multicast path more efficient.

Fig.2b illustrates an experimental comparison of CPU time between the TEA and Garrozi's. There is no rapid increase in CPU time taken by the TEA with the increase of the network nodes.

Fig.3a shows the relationship between the multicast path cost and the number of multicast nodes. When the number of multicast nodes increases, the total cost in the TEA grows slower than that of the SPT, which shows a better adaptability of the TEA to the number of multicast nodes.

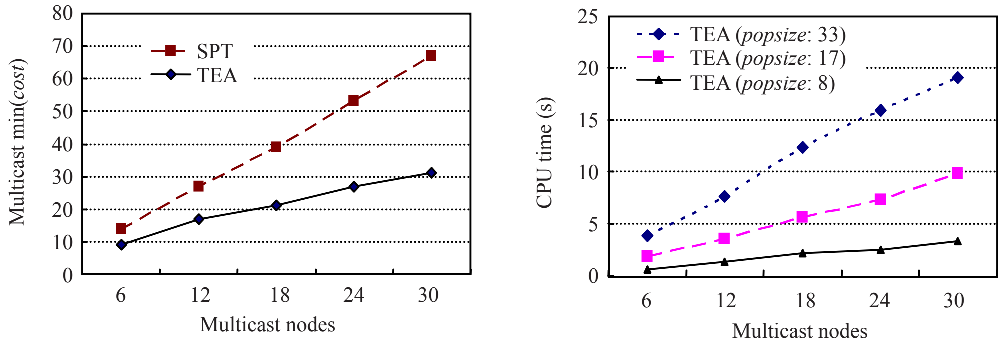
Fig.3b shows the relationship between CPU time and the number of multicast nodes for different population sizes in the TEA. If the total cost is kept in a minimal level, enlarging the population size does not speed up the algorithm, but on the contrary, slows down the performance. This result shows that a proper small population size is recommended.

To test the quality of dynamic multicast in experiments, we used probability P_r to determine when an addition or removal event occurs. We create a random number every time, if the random number is less than P_r, an addition event occurs; otherwise a removal event. To stabilize the number of multicast members, we took P_r=0.5. We conducted experiments on three network scales. For each scale, we create 100 random events and repeat the experiments 10 times over. The average results are shown in Fig.4a.



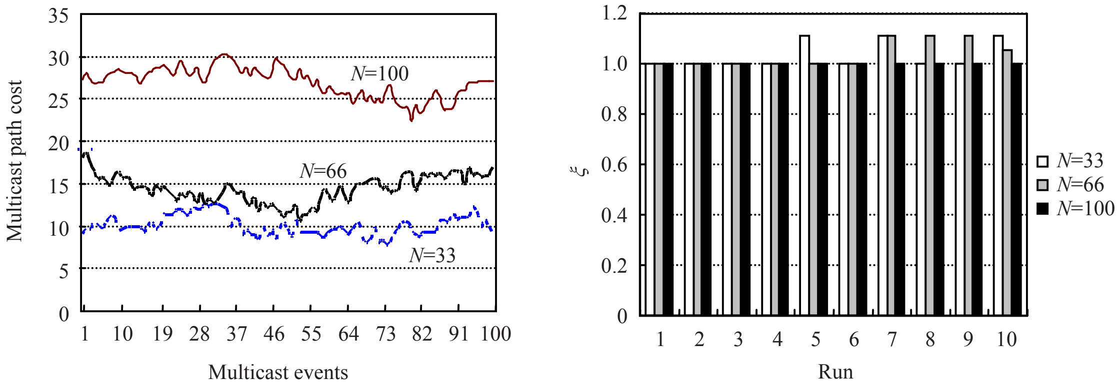
(a) (b)

Fig.2 Comparison of multicast cost (a) and CPU time (b) with the increase of network nodes among the TEA, SPT and Garrozi algorithms



(a) (b)

Fig.3 (a) Multicast cost vs. the number of multicast nodes for SPT and TEA; (b) CPU time vs. the number of multicast nodes in TEA for different *popsize* (a total of 33 nodes)



(a) (b)

Fig.4 (a) Cost of multicast path changes with dynamic events (add nodes or remove nodes), P_r=0.5; (b) Ratio (ξ) of the multicast cost to the optimal cost (Γ), ξ=*cost*/Γ

In each set of experiments there are random fluctuations of the multicast cost along with dynamic multicast events, but the trend remains horizontal in general, which shows that the quality of multicast paths does not worsen with the dynamic multicast events.

Fig.4b shows a situation where each TEA multicast cost departs from the optimal one. Let ξ represent the ratio of the multicast cost to the optimal cost, and then $\xi=cost/\Gamma$, Γ is the optimal cost. We see that the rate is 1 or close to 1 for all the three network scales.

At last, we test the convergence of the algorithm. Fig.5 shows the relationship between the number of unicast routings created and evolutionary generations for different multicast groups. The numbers of routings drop down quickly along with the evolutionary generations. Most routings are created in the first 10 generations and the trend is descending along with evolutionary generations.

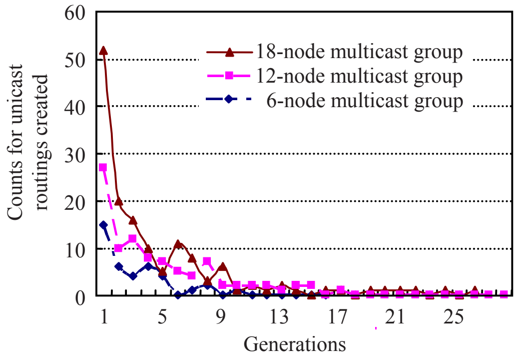


Fig.5 Relationship between the number of unicast routings created and evolutionary generation for different multicast groups (a total of 33 nodes)

CONCLUSION

This paper presents an evolutionary algorithm for dynamic multicast routing. This algorithm encodes or decodes the multicast path in link-duplicate-degree, and is simple and efficient both in multicast path creation and in dynamic group maintenance. Simulation experiments show that the algorithm has less multicast cost than SPT and Garrozi algorithms, and achieves better efficiency than Garrozi's. Based on global evaluation, the algorithm keeps the quality of dynamic multicast stabilized when multicast members change over time, which was verified by simulation experiments. Unlike the other algorithms of dynamic multicast, the rearrangement for the multicast paths is not necessary in the TEA, and thus the disturbance in the ongoing sessions is turned away. Experimental results show that the algorithm is convergent.

The algorithm is suitable for source- or center-based multicast and provides QoS guarantee. The next improvements for this algorithm will be to adapt the algorithm for hierarchical networks and to parallelize the algorithm.

References

Chen, L., Xu, Z.Q., 2006. Effective multicasting algorithm for dynamic membership with delay constraint. *J. Zhejiang Univ. Sci. A*, **7**(2):156-163. [doi:10.1631/jzus.2006.A0156]

Dalal, Y.K., Metcalfe, R.M., 1978. Reverse path forwarding of broadcast packets. *Commun. ACM*, **21**(12):1040-1048. [doi:10.1145/359657.359665]

Erlangung, Z., 2003. Algorithms for the Steiner Problem in Networks. Ph.D Thesis, University of Saarlandes, Saarbrücken, Germany.

Fu, H.Z., Li, C.P., 2005. An Improved Genetic Algorithm for Cost-Delay-Jitter QoS Multicast Routing. Proc. IEEE Int. Conf. on Computational Intelligence for Modelling, Control and Automation, **2**:1098-1103. [doi:10.1109/CIMCA.2005.1631616]

Garrozi, C., Araujo, A.F.R., 2006. Multiobjective Genetic Algorithm for Multicast Routing. IEEE Congress on Evolutionary Computation. Vancouver, BC, Canada, p.2513-2520. [doi:10.1109/CEC.2006.1688621]

Kompella, V., 1993. Multicast Routing Algorithms for Multimedia Traffic. Ph.D Thesis, University of California, San Diego, USA.

Liu, C.H., Chiang, T.C., Huang, Y.M., 2006. A Near-Optimal Multicast Scheme for Mobile Ad Hoc Networks Using a Hybrid Genetic Algorithm. Proc. 20th Int. Conf. on Advanced Information Networking and Applications, **1**:465-470. [doi:10.1109/AINA.2006.39]

Medina, A., Lakhina, A., Matta, I., Byers, J., 2001. BRITE: An Approach to Universal Topology Generation. Proc. 9th IEEE Int. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, p.346-353. [doi:10.1109/MASCOT.2001.948886]

Raghavan, S., Manimaran, G., Siva, R.M.C., 1999. A rearrangeable algorithm for the construction of delay-constrained dynamic multicast trees. *IEEE/ACM Trans. on Networking*, **7**(4):514-529. [doi:10.1109/90.793020]

Sun, L.J., Wang, R.C., Liu, C.L., 2007. A QoS multicast routing approach based on parallel quantum genetic algorithm. *J. Nanjing Univ. Posts Telecommun. (Nat. Sci.)*, **27**(2):72-75 (in Chinese).

Vijay, A., Thierry, T., Shivkumar, K., 2005. Encoding of multicast path. *LNCS*, **3462**:992-1004.

Viswanath, K., Obraczka, K., Tsudik, G., 2006. Exploring mesh and path-based multicast routing protocols for MANETs. *IEEE Trans. on Mobile Computing*, **5**(1):28-42. [doi:10.1109/TMC.2006.11]

Waxman, B.M., 1988. Routing of multipoint connections. *IEEE J. Sel. Areas Commun.*, **6**(9):1617-1622. [doi:10.1109/49.12889]

Wu, J.G., Yang, Y.Y., Chen, Y.X., Ye, X.G., 2006. A QoS aware overlay multicast routing protocol. *Chin. J. Computers*, **29**(11):1937-1947 (in Chinese).

Yue, C.J., Zheng, X.P., Jing, Y.W., 2007. QoS multicast routing based on chaotic genetic algorithm. *J. Northeastern Univ. (Nat. Sci.)*, **128**(10):1446-1449 (in Chinese).

Zhou, J.Y., Lin, Y., Hu, H.P., 2007. Dynamic Zone Based Multicast Routing Protocol for Mobile Ad Hoc Network. Proc. IEEE Int. Conf. on Wireless Communications, Networking and Mobile Computing, **3**:1528-1531. [doi:10.1109/WICOM.2007.385]

Zhu, L., Li, Z.S., Xing, J.C., Cheng, Y.H., 2007. An Immune Genetic Routing Algorithm for Mesh Network with QoS Constraints. Proc. IEEE Int. Conf. on Wireless Communications, Networking and Mobile Computing, **3**:1701-1704. [doi:10.1109/WICOM.2007.428]