On the Model of OoS Multicast Routing Problems in Active Networks

PAN Yun[1] YU Zhenwei[1] WANG Licheng[2] WANG Kehong[2]
(1 China University of Mining and Technology-Beijing, Beijing 100083)
(2 Knowledge Engineering Group of Computer Dept. of Tsinghua University, Beijing 100084)

Abstract In this paper, we propose the concept of order of services (OoS) and construct the programming model of OoS multicast routing in active networks. We also give methods to assure services and to compute the violation degree of OoS. Then, based on an improved Prüfer encoding method, we design a novel genetic algorithm for the new model. The complexities of encoding and decoding of our algorithm do not exceed $O(n^2)$. A great number of numerical experiments suggest that our algorithm is feasible and efficient.

Keywords active networks, multicast routing, QoS, OoS, genetic algorithms, Prüfer sequence

1 Introduction

With the booming of Internet, many new multimedia services based on the World Wide Web have emerged quickly and the standard architecture of Internet has already become incapable of meeting the new requirements under current situation. Meanwhile, because the function of processor has become more and more powerful while its price has become lower and lower, people want to use networks to provide not only best-effort transmission tasks, but also various of managements and controls, such as quality of services (QoS), multicasting, web caching, congestion control and so forth. In the traditional Internet, the implementations are complicated and transmissions delayed while performances are poor because all of these managements and control are placed in users' terminals.

Then, active networks and programming networks[1~2] are brought forward to conquer these drawbacks of the traditional architecture of Internet. The new idea is that the application systems can be implemented on both the terminals and routers. The programming ability has been introduced into networks to cater for various new services which have been demanded now and which would have been demanded in future. By this approach, the performance of networks have been improved evidently and the flow from source to destinations can obtain some new common services which are added on routers. For example, in a distributed video application, compression and decompression tasks can be implemented on routers instead of source and destinations. What is more, we can implement some simple services on different routers and combination them for some advanced services[12].

By multicasting, related users can share services and utilize the networks resource efficiently. In active networks, multicast routing should find a multicast tree which includes not only all destinations but all requested service routers.

In active networks, to assure services in a multicast routing includes, at least, two facets. The one is the quality of services (QoS) and the other is that each path from the source to each destination should contains all demanded services in proper order. For instance, compression service should preceded with decompression service. We named these constraints as order of services (OoS). At present, there are lots of papers and research works are concerned with QoS[10~11], but few of them have discussed about OoS. S.Y. Choi et al have proposed a layering idea to solve the problems of services combination in active networks[3]. However, they have not defined the concept of OoS and they have not given a discussion about the general constraints of OoS. Multicast routing problem is equivalent to the minimal Steiner tree problem, which is NP-hard. Based on the evolutionary mechanism, genetic algorithms have excellent performance in solving some NP-hard problems. This encourage us to employ genetic algorithms to solve the OoS multicast routing problem prompted in this paper.

In the second section of this paper, we introduce some concepts and construct the optimization model of OoS multicast routing problems; In the third section, to solve this model, a new genetic algorithm is designed by using the improved Prüfer sequence as chromosome code for multicast tree and a service tag vector as the companied chromosome. And then, in section four, numerical experimental results for supporting these methods will be presented out. Finally, we discuss the further topic in section five.

2 The OoS multicast routing model in active networks

For a given network $G = (V, E, c)$,V denotes the nodes(represent routers and subnets) set and E the edges(represent links between nodes) set. And the weight function $c: E \to R^{+}, c(e)$ represents the communication cost between two nodes. In active networks, active nodes can provide certain services. When an active node performances an assigned service, it needs CPU slices, buffers and other resources. We defined the resources, which is needed for an active node to complete certain service, as the service cost.

At first, we assume that: 1)Each active node can provide only one kind of service. If an active node v can provide m(m>=2) kinds of service, we replace it with a completed graph K_m which includes m active nodes labeled v_1, v_2, ……, v_m respectively. And then assume each edge weight $c(v_i,v_j)$ is zero and each node v_j provides(and only provides) a distinctive service among these m kinds of services. What's more, if there is a node u which is adjacent to v, then we let u adjacent to each v_j and $c(< u, v_j >) = c(< u, v >)$; 2)In a given multicasting route, each active node may choose to provide the service or not to do so. Then, we need to distinguish two concept: "capable of providing service" and "provide service". When we say "an active node is capable of providing a service", we just mean that this node has the ability to provide the service but it may not afford the service--maybe, this service is not needed in the given multicast; Or, this service is provided by other active nodes in this route. However, when we say "an active node provides a service", we mean that this service is assigned to this node and this node afford this service in this route.

Secondly, we introduce the service cost function, the service category set and the OoS constraint set as follows.

Definition 1   The service cost function is defined as

$$c':V \to [0,\infty), c'(v) = \begin{cases} \text{cost when node } v \text{ provides service, } v \text{ is an active node} \\ 0, \text{otherwise} \end{cases} \quad (1)$$

and the service category function is defined as

$$s: V \to N, s(v) = \begin{cases} j, \text{node } v \text{ can provide } j^{\text{th}} \text{ kind of service} \\ 0, \text{otherwise} \end{cases} \quad (2)$$

Definition 2 The service category set is defined as

$$C \overset{def}{=} V / R = \{[v] : v \in V\} \quad (3)$$

Where R is a binary relation defined on V as follows

$$R \overset{def}{=} \{< u, v >: s(u) = s(v)\} \quad (4)$$

It is easy to proof that R is an equivalent relation, so the definition 2 is well-defined.

When $s(v)=j$ holds, we can say that active node v provide $[v]$ kind service or *j-th* kind service.

Definition 3 Set $D \subseteq V$, we define the OoS constraints set $L_D \subseteq C \times C$ on D as a partial order set

$$L_D \overset{def}{=} \{<[u],[v]>: u, v \in D, [u] \text{ kind of service should not lagged behind } [\text{v}] \text{ kind of service}\} \quad (5)$$

Definition 4 In order to describe the violation degree of a given multicasting tree T to the OoS constraints set L_D, we define

$$\overrightarrow{a_T} = (a_1, a_2, ..., a_{|V|}), a_j = \begin{cases} 1, v_j \in V_T \text{ and provide service} \\ 0, \ \ others \end{cases} \quad (6)$$

$$A_{L_D} = (l_{ij})_{n\times n}, n = |V|, l_{ij} = \begin{cases} a_i * a_j, <[v_i],[v_j]> \in L_D \\ 0, others \end{cases} \quad (7)$$

$$A_T = (t_{ij})_{n\times n}, n = |V|, t_{ij} = \begin{cases} 1, v_i \text{ is the ancestor of } v_j \text{ in } T \\ 0, others \end{cases} \quad (8)$$

$$A_v = (1_{n\times n} - A_{L_D}).*A_T = (\overline{l_{ij}} * t_{ij})_{n\times n} = \begin{cases} a_i * a_j, < v_j, v_i > \in L_D, v_i \text{ is the ancestor of } v_j \text{ in } T \\ 0, others \end{cases} \quad (9)$$

Where $\overrightarrow{a_T}$ is the service tag vector of multicast tree T and $\overline{l_{ij}} = 1 - l_{ij}, (1 \le i, j \le n)$ holds. The operation .* between two matrix $A_{n\times n}$ and $B_{n\times n}$ is defined as a new matrix $C_{n\times n}$ which is defined as $C = (a_{ij} \cdot b_{ij})_{n\times n}$.

Then, set $\Lambda = \{T : T \text{ is the multicasting tree of network } G\}$, for $\forall T \in \Lambda$, the violation function of T to L_D can be defined as

$$v : \Lambda \to [0,1],\ v(T) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\overline{l_{ij}} * t_{ij} \tag{10}$$

i.e. the proportion of non-zero elements in the matrix A_v.

In addition, in order to assure a given kind of service, the follow concepts are needed.

Definition 5 We define two vectors

$$\overrightarrow{p_{L_D}} = (x_1, x_2, ..., x_k), k = |C|, x_j = \begin{cases} 1, [v_j] \in C_{L_D} \\ 0, otherwise \end{cases} \tag{11}$$

and

$$\overrightarrow{p_T} = (y_1, y_2, ..., y_k), k = |C|, y_j = \begin{cases} 1, [v_j] \in C_T \\ 0, otherwise \end{cases} \tag{12}$$

as project vectors of L_D and T to C. Where

$$C_{L_D} \overset{def}{=} \{[u] : [u] \in C, \exists [v] \in C(<[u],[v]> \in L_D \vee <[v],[u]> \in L_D\} \tag{13}$$

and

$$C_T \overset{def}{=} \{[u] : [u] \in C, u \in V_T\} \tag{14}$$

are called projects of L_D and T to C , respectively.

Apparently, when $\overrightarrow{p_{L_D}} = \overrightarrow{p_T}$ holds, the multicasting tree T assures each kind of services which are emerged in the OoS constraints set L_D.

In summary, the OoS active networks model can be represented with a weighed graph $G = (V, E, c; s, c')$, and a multicast routing instance can be described as the triple $M = (source, D, L_D)$. Where $source \in V$ is the source node of this multicasting, D the destinations set and L_D the OoS constraints set. The objective of this problem is to find a “minimal” multicasting tree which takes *source* as the root, contains all nodes in D and does not violate with L_D. Here, the term “minimal” means that the total cost of multicasting(including the communication costs among nodes and the active service costs) reach minimal. This is equivalent to solve the programming problem as follows

$$\begin{cases} \min\limits_{T \subseteq G} \quad f = \sum_{e \in E_T} c(e) + \sum_{j=1}^{n} c'(v_j) * a_j \\ s.t. \quad v(T) = 0, \\ \qquad \overrightarrow{p_{L_D}} = \overrightarrow{p_T}, \\ \qquad T \text{ is the multicasting tree of } G \text{ and } source \text{ is } T\text{'s root while } D \subseteq V_T. \end{cases} \tag{15}$$

Where a_j, $v(T)$, $\overrightarrow{p_{L_D}}$, $\overrightarrow{p_T}$ are defined as formula (6)~(12).

Employing penalty function method, we convert (15) into (16)

$$\begin{cases} \min\limits_{T \subseteq G} \quad f = \sum_{e \in E_T} c(e) + \sum_{j=1}^{n} c'(v_j) * a_j + P_1(v(T)) + P_2\left(\left\|\overrightarrow{p_{L_D}} - \overrightarrow{p_T}\right\|\right) \\ s.t. \quad T \text{ is the multicasting tree of } G \text{ and } source \text{ is } T\text{'s root while } D \subseteq V_T. \end{cases} \tag{16}$$

Where $\|\cdot\|$ is the form which can be chosen according to different applications. We use 2-form here.

And penalty functions P_1 and P_2 can be designed according the method used to solve (16). The principle is that $P_1(0) = P_2(0) = 0$ and $\exists M > 0$ such that $P_1(x) \to \infty, P_2(x) \to \infty$ hold when $x > M$.

What is the complexity of this problem? At first, we know that the minimal Steiner tree problem is NP-hard. And this problem is equivalent to solve the constrained minimal Steiner tree problem, so it is also NP-hard. Genetic algorithms have excellent performance in solving some NP-hard problems and this encourage us to employ genetic algorithms to solve (16).

3 The genetic algorithm for OoS multicast routing model in active networks

Genetic algorithms (GAs) are random, adaptive search methods inspired by Darwin's theory about evolution. Solution to a problem solved by genetic algorithms is evolved and the approach is gaining a growing following in the physical, life, computer and social sciences and in engineering. A genetic algorithm consists of several crucial aspects, such as chromosome encoding and decoding, population initializing, individual’s fitness value computing and scaling, and genetic operating (selecting, crossing and mutating), and so forth.

Based on the discussion above, we designed a novel genetic algorithm to solve the OoS multicasting model (16). The penalty functions are designed as

$$P_1(x) = P_2(x) = \begin{cases} (1+\alpha)^{k+x} \log_{\max G} k, x > 0 \\ 0, otherwise \end{cases} \tag{17}$$

Where k and max$G$ are the current generation number and maximum generation number of evolution respectively. And $\alpha \in (0,1]$ is needed to choose carefully. After lots of random experimental simulations, we find that $\alpha = 0.05$ is fit for general circumstances.

Obviously, the penalty functions are dynamic. They will gradually strengthen penalty to infeasible solutions with the increasing of k.

3.1 Dual chromosomes

We design dual chromosomes to represent a multicasting tree and a service tag vector:

1) Multicasting tree chromosome T There are n^{n-2} distinct sequences (called Prüfer sequences, or Prüfer codes)[6] of length n-2 with entries being from natural number 1 to n. The Prüfer sequences provide one of the most concise encoding methods for designing genetic algorithms to solve the optimization problems that are correlated with spanning trees. A Prüfer sequence denotes a free

spanning tree of the complete graph $K_{n.}$ It is different from a multicast tree. The latter needs a labeled node as root. So we prompt an improved Prüfer sequence which includes *n*-1 entries and the last one denotes the root of *T* and each directed edge always starts from the first node in P and ends at the first node in Q (Fig.1 and Fig.2 illustrate the process of coding and decoding).

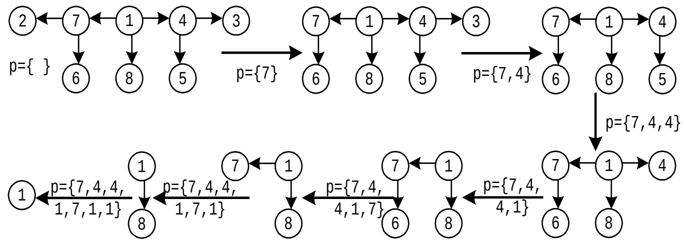

Fig.1 The process of coding

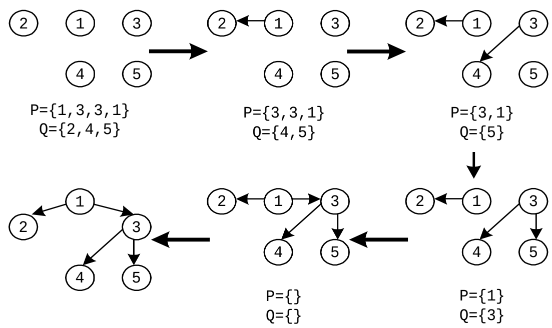

Fig.2 The process of decoding

What's more, we need to prune leaves that do not belong to the multicast destination group *D*. The pruning algorithm, labeled *prune*, is as follows:

Algorithm *prune*

Input: *T*, the spanning tree of K_n and *D*, the destination group of *G*;

Output: *T*, the multicasting tree;

[Start]

1) Visit each vertex while traversing *T* in post order and do a) in each step;

a) If the current vertex is a leaf and not in *D*, then delete it from *T*;

2) Output *T*.

[End]

2) The service tag vector $\overrightarrow{a_T}$ This is a 0-1 vector which is random generated and a_j=0 if and only if v_j belongs to *T* and provide service in this multicast routing.

3.2 The novel genetic algorithm for (16)

A genetic algorithm consists of several crucial aspects, such as chromosome encoding and decoding, population initializing, individual's fitness value computing and scaling, and genetic operating (selecting, crossing and mutating), and so forth. Based on the discussion above, we designed a novel genetic algorithm for (16). First, we decode a multicasting tree chromosome code, i.e. an extended prufer sequence to a spanning tree, denoted *T*, of completed graph K_n and prune *T* to a multicasting tree of the given network *G* using *prune* algorithm. Second, we compute $v(T)$ and

$\left\|\overrightarrow{p_{L_D}} - \overrightarrow{p_T}\right\|_2$ according the service tag chromosome, i.e. the 0-1 vector $\overrightarrow{a_T}$, and the formula (6)~(14). Then we compute penalty values by formula (17). Finally, We employ the formula (16) to estimate the fitness value, denoted by *g*, of the multicasting tree and then scaling *g* to *f* by the formula *f*=1/(1+*g*). We adopt an improved roulette wheel selection operator[7], a random multi-points crossover operator[9], and three mutation operators: a single gene mutating operator, a gene fragment reversing operator and a gene fragment shifting operator in the new algorithm. Elitism strategy is also adopted[7~9]. In each population, the best individual, called elitist, is kept from crossing and mutating.

3.3 The Computational Complexity Analysis

The complexities of decoding a Prüfer sequence to a spanning tree and pruning a spanning tree to a multicast tree are $O(nlogn)$ and $O(n)$, respectively. The most complicated compute among formula (6)~(14) is the operation .* between two matrix. So the complexity of computing and scaling the fitness is $O(n^2)$. All the complexities of crossing operator and mutating operators do not exceed $O(n)$. In conclusion, the total complexity of the new genetic algorithm, given the population size *p* and the maximum generation max*G*, is $O(p \times \max G \times n^2)$.

4 The simulation and analysis

The topology structure of networks used in our experiments is Fig.3. In the figure, these random numerical values are departed two parts, numerical values on links represent communication costs and the numerical values in active nodes represent service costs. Supposed node 1 is the source of multicasting and the destination group *D*={ 4 ,9,10}, the best multicasting tree are consisted of the thick lines in Fig.3. We run our algorithm more than 100 times with population size *p*=40 and the max evolution generation max*G*=100, 300, 500, 700 and 1000 respectively. The results, which reflect the change between the maximum evolutional generation and the success rate of the new algorithm that find feasibility solutions and global optimal solutions, are obtained and showed in Fig.4 and Fig.5. In the average of each run, the decreasing trend of multicasting cost along with evolutional generation in undirected graph with 1 kind of services is showed in Fig.6.

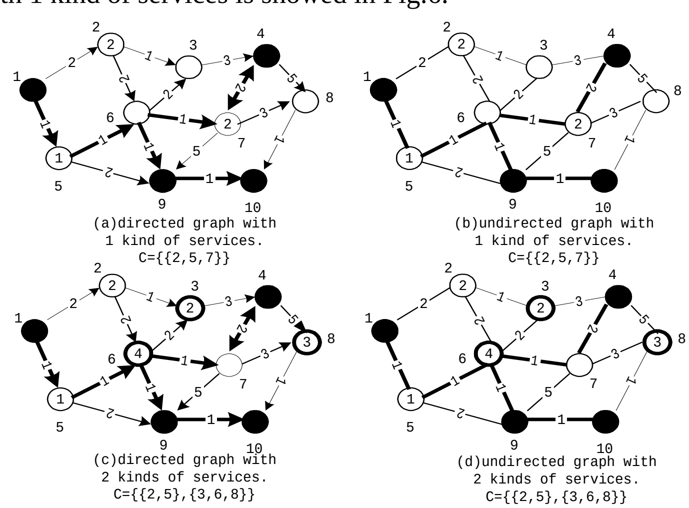

Fig.3 Topology structure of networks

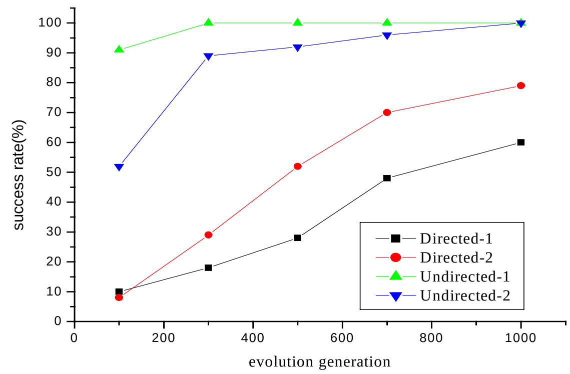

Fig.4 the relation between feasibility solutions(cost<18) and maximum evolutional generation

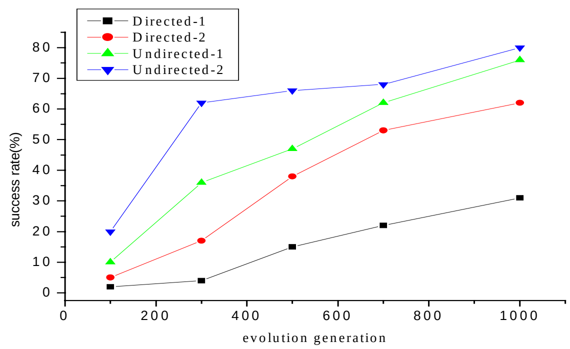

Fig.5 the relation between global optimal solutions and maximum evolutional generation

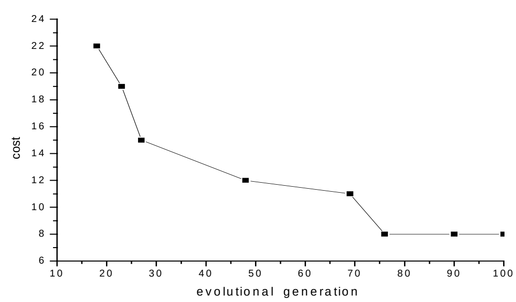

Fig.6 the decreasing trend of multicasting cost along with evolutional generation

From the simulation results, we can conclude that:

1) It is easier to solve OoS multicast routing problems of undirected graph than those of directed graph, and our algorithm is more efficient to undirected graph.

2) It is easier to solve OoS multicast routing problem with less service categories.

3) From Fig.6, it indicates that it is feasible to solve OoS multicast routing problems using genetic algorithms.

5 Conclusion

In active networks, to assure services means, at least, to assure the quality of services (QoS) and to assure the order of services (OoS). At present, there are few of works and papers are concerned with the latter. In this paper, we propose the concept of order of services, define a serial of related concepts such as service cost function, service category function, service category set, partial set of order of services, and so on. We also bring forward methods to assure the services and to describe the violation degree of a given multicasting tree to a given OoS set. After these works, we construct the programming model of OoS multicast routing in active networks. Then, based on an improved prufer coding method, we design a novel genetic algorithm for the new model. The complexities of coding and decoding of our algorithm do not exceed $O(n^2)$. A great number of numerical experiments suggest that our algorithm is feasible and efficient. By the way, we should point out that OoS multicast routing problem is also an unfertilized topic in overlay networks and this is just the next work we want to do.

Reference

[1] Andrew T. Campbell, Herman G. De Meer, Michael E. Kounavis, Kazuho Miki, John B. Vicente, and Daniel Villela. A survey of programmable networks. Computer Communication Review, 29(2):7–23, April 1999.

[2] David L. Tennenhouse, Jonathan M. Smith, W. David Sincoskie, David J. Wetherall, and Gary J. Minden, “A survey of active network research,” IEEE Communications Magazine, vol. 35, no. 1, pp. 80–86, Jan. 1997.

[3] Sumi Y. Choi, Jonathan Turner, and Tilman Wolf, “Configuring session in programmable networks,” Proceedings of IEEE Infocom 2001,Apr. 2001.

[4] Salama H F,Reeves D S,Viniotis Y. Evaluation of multicast routing algorithms for real- time communication on high- speed networks. IEEE Journal on Selected Areas in Communications, 1997,15(3): 332 - 345

[5] Kou L,Markowshy G,Berman L. A fast algorithm for Steiner trees [J]. Acta. Informatics,1981 ,15:141-145

[6] Luc Devroye，Enumeration and Representation of Free Trees，in Data Structures and Algorithms，April 19, 2000 at http://jeff.cs.mcgill.ca/~luc/251.html.

[7][Japan]GEN Mitsuo，CHEN Runwei，Genetic Algorithm and engineering design，Beijing: Publishing
House of Science. 2000

[8] J.H.Holland, Genetic Algorithms, Scientific American July 1992: 44~50.

[9] Dr. Thomas Bäck, Evolutionary Algorithms in Theory and Practice, Oxford University Press, New York Oxford: 1996.

[10] Maimour, M.; Pham, C.D. A throughput analysis of reliable multicast protocols in an active networking environment Computers and Communications, 2001. Proceedings. Sixth IEEE Symposium on , 2001 Page(s): 151 -158

[11] Sonoda, M.; Okamura, K.; Araki, K. Design of general reliable multicast architecture with active

network framework Information Networking, 2001. Proceedings. 15th International Conference on , 2001 Page(s): 825 –830
[12] D. Xu, K. Nahrstedt,, “Finding Service Paths in a Media Service Proxy Network,” San Jose, CA, Jan. 2002, Proc. Of SPIE/ACM Multimedia Computing and Networking Conference(MMCN’02).