An ABC Supported QoS Multicast Routing Scheme Based on Beehive Algorithm

Xingwei Wang, Qiang Chen, Rongzhu Zou, Min Huang
College of Information Science and Engineering, Northeastern University, Shenyang, 110004, P.R. China
wangxw@mail.neu.edu.cn

ABSTRACT

In this paper, a QoS multicast routing scheme with ABC (Always Best Connected) supported is proposed based on the beehive algorithm. To deal with the inaccurate network status and the imprecise user QoS requirement, the proposed scheme uses the range to describe them, introduces the edge bandwidth pricing, the edge evaluation and the tree evaluation, and tries to find a QoS multicast tree with the Pareto optimum under the Nash equilibrium on both the network provider utility and the user utility achieved or approached. Simulation results have shown that it is both feasible and effective.

Categories and Subject Descriptors

C.2.1 [**COMPUTER-COMMUNICATION NETWORKS**]: Network Architecture and Design –*Network communications*

General Terms

Algorithms, Performance, Design

Keywords

QoS (Quality of Service) Multicast Routing, ABC (Always Best Connected), Beehive Algorithm, Nash Equilibrium, Pareto Optimum

1. INTRODUCTION

NGI (Next Generation Internet) is now becoming an integrated network converged seamlessly by heterogeneous multi-segment multi-provider sub-networks, such as terrestrial-based, space-based, fixed and mobile ones. Its backbone and access links become diversified. Several different kinds of links may coexist on each hop for a user to choose when routing. It is possible for a user to be ABC (Always Best Connected) [1] to NGI in the course of communication and thus the so-called global QoS (Quality of Service) roaming should be supported seamlessly.

ABC means that a user can connect with NGI anytime, anywhere in the currently best way and switch to the better way adaptively and transparently whenever it comes forth. However, 'best' is a fuzzy concept, depending on many factors, such as user QoS requirement, cost a user willing to pay and user preference. In addition, with the gradual commercialization of the network operation, both the network provider and the user profit should be considered with their utility win-win supported. At the same time, NGI's heterogeneity and dynamics, terminal and even network mobility, unavoidable message transfer delay and its uncertainty make it hard to describe the network status exactly and completely [2]. The user QoS requirement is affected largely by a lot of subjective factors and often can not be expressed accurately, thus the flexible QoS description is needed. All these make it hard to provide QoS multicast routing with ABC supported in NGI.

QoS multicast routing has been proven to be NP-complete [3]. Many heuristic and intelligent optimization algorithms have been used to solve it. In [4], a QoS-aware algorithm is proposed to find the multicast routing tree by constructing the constraint vector with the parallel multiple paths searching. In [5], a heuristic QoS multicast routing algorithm is presented. It adopts distributed computing and routing label to reduce its time complexity. However, the inaccurate network status and the utility optimization of both the user and the network provider are not taken into account in [4, 5]. In [6], a multi-constrained QoS multicast routing algorithm under the inaccurate information is introduced. It uses the improved Bellman-Ford algorithm for routing. In [7], based on the probability theory, GA (Genetic algorithm) is used to find the optimal multicast routing tree under the inaccurate information. In [8], the fuzzy set is used to describe the inaccurate link state, and the multicast routing tree is built based on GA with the fuzzy measures introduced. Nevertheless, the proposed algorithms in [6, 7, 8] only consider meeting the user QoS requirement and do not consider achieving utility win-win between the user and the network provider. In [9], the SPEA (Strength Pareto Evolutionary Algorithm) is proposed to build a tree with the Pareto optimum. It considers the user utility, but does not consider the network provider utility. In [10], the utility function and the pricing scheme are introduced to make the multicast resource allocation fairly. However, its focus is on the network congestion control and not on the multicast routing. It also does not consider the inaccurate network status. In [11], a fair QoS multicast routing scheme is proposed with a Kelly/PSP model based pricing scheme introduced. Its focus is to solve the multicast fairness issue, not dealing with utility win-win between the user and the network provider and not considering the inaccurate network status.

By far, how to provide a QoS multicast routing scheme to support utility win-win between the user and the network provider under the imprecise network status and the inaccurate user QoS requirement from ABC viewpoint has not got much attention. In this paper, a QoS multicast routing scheme with ABC supported

Qshine08 July 28-31, 2008, Hong Kong, China
ISBN 978-963-9799-26-4
DOI 10.4108/ICST.QSHINE2008.3900

is proposed. In order to deal with the imprecise network status information and the flexible user QoS requirement, the range is used to describe them and the edge bandwidth pricing, the edge evaluation and the multicast tree evaluation are introduced. Based on the beehive algorithm [12], the proposed scheme tries to find a QoS multicast tree with Pareto optimum under Nash equilibrium on both the network provider utility and the user utility achieved or approached.

2. Model Description

2.1 Network Model and Routing Request

A network can be modeled as a graph $G(V,E)$, V is the node set and E is the edge set. $\forall v_i, v_j \in V$, there maybe exist several edges to represent different kinds of links between them, $i, j = 1,\ldots,|V|$. Just for simplicity, the node parameters are merged into the edge ones. $\forall e_l \in E$, consider the following parameters: network provider number z, total bandwidth tbw_l, available bandwidth bw_l, delay dl_l, delay jitter jt_l, error rate ls_l, bandwidth unit cost ct_l and bandwidth price p_l. $z \in NPS$ and NPS is the set of all network provider numbers. A QoS multicast routing request is denoted as $< v_s, D, R_{Bw}, R_{Dl}, R_{Jt}, R_{Ls}, PY, BD >$. $v_s \in V$ is its source node; $D = \{v_{d_1},\ldots,v_{d_{|D|}}\} \in V$ is its destination node set; $R_{Bw} = (\Delta_{Bw_1},\cdots,\Delta_{Bw_{|D|}})$, $R_{Dl} = (\Delta_{Dl_1},\cdots,\Delta_{Dl_{|D|}})$, $R_{Jt} = (\Delta_{Jt_1},\cdots,\Delta_{Jt_{|D|}})$ and $R_{Ls} = (\Delta_{Ls_1},\cdots,\Delta_{Ls_{|D|}})$ are its bandwidth, delay, delay jitter and error rate requirement vectors, with $\Delta_{Bw_i} = [BW_i^L, BW_i^H]$, $\Delta_{Dl_i} = [DL_i^L, DL_i^H]$, $\Delta_{Jt_i} = [JT_i^L, JT_i^H]$ and $\Delta_{Ls_i} = [LS_i^L, LS_i^H]$ representing bandwidth, delay, delay jitter and error rate requirement of the multicast user attached to v_{d_i} (below just use the term 'the i_{th} user' simply). The actually received QoS of one user from one edge is classified into four levels: $QL = \{\text{Excellent}, \text{Good}, \text{Fair}, \text{Poor}\}$. $PY = (py_1,\cdots,py_{|D|})$ and $BD = (bd_1,\cdots,bd_{|D|})$ are the expenditure and bid vectors each user willing to pay and offer, with $py_i \in \{py_{ie}, py_{ig}, py_{if}, py_{ip}\}$ and $bd_i \in \{bd_{ie}, bd_{ig}, bd_{if}, bd_{ip}\}$ corresponding to the specific expenditure and bid the i_{th} user willing to pay and offer when the i_{th} user actually received QoS level from e_l is a specific value Ql_{il} from QL. Here, it is assumed that a QoS multicast routing request is heterogeneous, that is, each user has different requirement on QoS, expenditure and bid. If it is homogeneous, that is, each user has the same requirement, $R_{Bw}, R_{Dl}, R_{Jt}, R_{Ls}$, PY and BD all only have one element.

In this paper, the objective of the proposed scheme is to find a QoS multicast tree T_{sD} with v_s as its root and $v_{d_i} \in D$ as its leaf, making the Pareto optimum under the Nash equilibrium between the user utility and the network provider utility along T_{sD} as well as the minimum cost of T_{sD} achieved or approached.

2.2 Bandwidth Pricing

To promote a user to consume bandwidth rationally, the edge bandwidth price can be divided into three regions, i.e., low, sound and high [13]. Assume η_l represents the loaded level of e_l and is calculated as follows:

$$\eta_l = 1 - \frac{bw_l}{tbw_l} \tag{1}$$

If $\eta_l < \eta_l^0$, e_l is considered to be low-loaded, its bandwidth price is at the low region and can be adjusted according to the formula (2); if $\eta_l > \eta_l^1$, e_l is considered to be high-loaded, its bandwidth price is at the high region and determined by biding; otherwise, e_l is considered to be moderate-loaded, its bandwidth price is at the sound region and can be adjusted according to the formula (3).

$$p_l = \begin{cases} p_l^{\min} & \eta_l < \eta_l^{\min} \\ \dfrac{A}{1+\alpha\times\eta_l^{-\beta}} & \eta_l^{\min} \le \eta_l \le \eta_l^0 \end{cases} \tag{2}$$

$$p_l = \begin{cases} p_l^{\max} & \eta_l^{\max} < \eta_l < \eta_l^1 \\ B\times(2-e^{-\delta\times(\eta_l-\eta_l^0)^2}) & \eta_l^0 \le \eta_l \le \eta_l^{\max} \end{cases} \tag{3}$$

η_l^0 and η_l^1 are preset values, $0 < \eta_l^0 < \eta_l^1 < 1$. $p_l^{\min}$ and $p_l^{\max}$ are the lower bound of the low region and the upper bound of the sound region of e_l, $\eta_l^{\min}$ and $\eta_l^{\max}$ are their edge loaded level. Use p_l^0 to represent the bandwidth baseline price of e_l, its edge loaded level is η_l^0, $p_l^{\min} \le p_l^0 \le p_l^{\max}$. For $\eta_l^{\min} \le \eta_l \le \eta_l^0$, p_l is semi-rising Cauchy distribution alike with $\beta = 2$ usually, $p_l = p_l^0$ when $\eta_1 = \eta_l^0$ and $p_l = p_l^{\min}$ when $\eta_1 = \eta_l^{\min}$ [14], thus A and α can be calculated by the formula (4) and (5).

$$A = p_l^0 \times \left(1 + \frac{p_l^0 - p_l^{\min}}{p_l^{\min}\times(\eta_l^{\min})^{-2} - p_l^0\times(\eta_l^0)^{-2}} \times (\eta_l^0)^{-2}\right) \tag{4}$$

$$\alpha = \frac{p_l^0 - p_l^{\min}}{p_l^{\min}\times(\eta_l^{\min})^{-2} - p_l^0\times(\eta_l^0)^{-2}} \tag{5}$$

For $\eta_l^0 \le \eta_l \le \eta_l^{\max}$, p_l is semi-rising normal distribution alike, $p_l = p_l^0$ when $\eta_l = \eta_l^0$ and $p_l = p_l^{\max}$ when $\eta_l = \eta_l^{\max}$ [14], thus B and δ can be calculated by the formula (6) and (7).

$$B = p_l^0 \tag{6}$$

$$\delta = -\frac{\ln\left(2 - \dfrac{p_l^{\max}}{p_l^0}\right)}{(\eta_l^{\max} - \eta_l^0)^2} \tag{7}$$

2.3 Adaptability Membership Degree

It is introduced to describe the adaptability of the edge status to the user QoS requirement. The bandwidth, delay, delay jitter and error rate adaptability membership degree function of e_l to the i_{th} user requirement are defined in the following formula (8) to (11).

$$g_{il}^1 = \begin{cases} 2\times(\frac{bw_l - BW_i^L}{BW_i^H - BW_i^L})^2 & BW_i^L < bw_l \le \frac{(BW_i^H + BW_i^L)}{2} \\ 1-2\times(\frac{BW_i^H - bw_l}{BW_i^H - BW_i^L})^2 & \frac{(BW_i^H + BW_i^L)}{2} < bw_l \le BW_i^H \\ 1 & bw_l > BW_i^H \end{cases} \tag{8}$$

$$g_{il}^2 = \begin{cases} 1 & dl_l \le DL_i^L \\ (\frac{DL_i^H - dl_l}{DL_i^H - DL_i^L})^k & DL_i^L < dl_l \le DL_i^H \\ 0 & dl_l > DL_i^H \end{cases} \tag{9}$$

$$g_{il}^3 = \begin{cases} 1 & jt_l \le JT_i^L \\ (\frac{JT_i^H - jt_l}{JT_i^H - JT_i^L})^k & JT_i^L < jt_l \le JT_i^H \\ 0 & jt_l > JT_i^H \end{cases} \tag{10}$$

$$g_{il}^4 = \begin{cases} 1 & ls_l \le LS_i^L \\ (\frac{LS_i^H - ls_l}{LS_i^H - LS_i^L})^k & LS_i^L < ls_l \le LS_i^H \\ 0 & ls_l > LS_i^H \end{cases} \tag{11}$$

The formula (8) is S distribution alike and the formula (9), (10) and (11) are k -parabolic distribution alike, all having smooth transition feature [14].

2.4 Utility

2.4.1 *User Utility*

According to the formula (8) to (11), get the evaluation matrix $G_{il} = [g_{il}^1 \quad g_{il}^2 \quad g_{il}^3 \quad g_{il}^4]^T$ of the i_{th} user on e_l . The weight matrix $\Lambda = [\lambda_1 \quad \lambda_2 \quad \lambda_3 \quad \lambda_4]$ is given by the application nature, λ_1 , λ_2 , λ_3 and λ_4 are the relative significance weights of bandwidth, delay, delay jitter and error rate on the application QoS respectively, $0 \le \lambda_1, \lambda_2, \lambda_3, \lambda_4 \le 1$, $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$. The satisfaction degree St_{il} of the i_{th} user to the status of e_l is as follows:

$$St_{il} = \Lambda \circ G_{il} \tag{12}$$

The mapping between St_{il} and Ql_{il} is defined as follows:

$$Ql_{il} = \begin{cases} Excellent & St_{il} \ge \alpha_1 \\ Good & \alpha_2 \le St_{il} < \alpha_1 \\ Fair & \alpha_3 \le St_{il} < \alpha_2 \\ Poor & St_{il} < \alpha_3 \end{cases} \tag{13}$$

α_1 , α_2 and α_3 are preset values. According to Ql_{il} , py_i can be determined from $\{py_{ie}, py_{ig}, py_{if}, py_{ip}\}$.

The satisfaction degree SC_{il} of the i_{th} user to the expenditure he paid for using e_l is defined as follows:

$$SC_{il} = \begin{cases} 1 & p_l \times abw_l \le \frac{1}{\chi} py_i \\ (\frac{py_i - p_l \times abw_l}{py_i - \frac{1}{\chi} py_i})^k & \frac{1}{\chi} py_i < p_l \times abw_l \le py_i \\ 0 & p_l \times abw_l > py_i \end{cases} \tag{14}$$

abw_l is the actually allocated bandwidth for the i_{th} user on e_l , $BW_i^L \le abw_l \le BW_i^H$; χ is a tuning factor, $\chi > 1$, k is preset value. The i_{th} user utility on e_l is as follows:

$$uu_{il} = \varpi_1 \times St_{il} + \varpi_2 \times SC_{il} \tag{15}$$

ϖ_1 and ϖ_2 are preference weights on St_{il} and SC_{il} , $0 \le \varpi_1, \varpi_2 \le 1$, $\varpi_1 + \varpi_2 = 1$.

2.4.2 *Network Provider Utility*

Assume that there are k edges owned by the different network providers between two nodes and consider the following attributes for each edge: available bandwidth, delay, delay jitter, error rate, loaded level, bandwidth price and being_selected_probability, constituting a $k \times 7$ evaluation matrix $F_{k\times 7}$. Its element f_{zy} represents the y_{th} attribute of the edge provided by the z_{th} network provider, $1 \le z \le k$, $1 \le y \le 7$. Do normalization to 'the larger the better' and 'the smaller the better' attributes according to the formula (16) and (17).

$$f_{zy}^{'} = \frac{f_{zy}}{f_{\max,y} + f_{\min,y}} \tag{16}$$

$$f_{zy}^{'} = \frac{f_{\max,y} + f_{\min,y} - f_{zy}}{f_{\max,y} + f_{\min,y}} \tag{17}$$

$$f_{\min,y} = \min_z \{f_{zy}\} \tag{18}$$

$$f_{\max,y} = \max_z \{f_{zy}\} \tag{19}$$

$f_{\min,y}$ and $f_{\max,y}$ are the minimum and maximum of the y_{th} attribute value.

The network provider utility is calculated as follows:

Step1: Construct the evaluation matrix $F_{k\times 7}$.

Step2: Modify $F_{k\times 7}$ as follows: normalize the delay, delay jitter, error rate and loaded level according to the formula (17) and the available bandwidth, bandwidth price and being_selected_probability according to the formula (16).

Step3: Calculate the standard deviation s_y for each attribute according to the formula (20), compute the weight for each attribute according to the formula (22).

$$s_y = \left(\frac{1}{k} \times \sum_{z=1}^{k} (f'_{zy} - \overline{f_y})^2\right)^{\frac{1}{2}} \tag{20}$$

$$\overline{f_y} = \frac{1}{k} \times \sum_{z=1}^{k} f'_{zy} \tag{21}$$

$$w_y = \frac{s_y}{\sum_{\kappa=1}^{7} s_\kappa} \tag{22}$$

Step4: Calculate the z_{th} network provider utility nu_{zl} on e_l according to the formula (23).

$$nu_{zl} = \sum_{y=1}^{7} w_y \times f'_{zy} \tag{23}$$

2.5 Gaming Analysis

The network provider and the user play game on the edge. Each network provider has two gaming strategies: whether he is willing to provide the edge to the user or not. The user also has two gaming strategies: whether he is willing to accept the provided edge or not. The gaming matrixes of the i_{th} user and the z_{th} network provider on e_l are as follows:

$$UU_{il} = \begin{bmatrix} \ln\frac{uu_{il}}{uu_{il_0}} & \gamma \times \ln\frac{uu_{il}}{uu_{il_0}} \\ -\mu \times \ln\frac{uu_{il}}{uu_{il_0}} & 0 \end{bmatrix} \tag{24}$$

$$NU_{zl} = \begin{bmatrix} \ln\frac{nu_{zl}}{nu_{zl_0}} & -\mu \times \ln\frac{nu_{zl}}{nu_{zl_0}} \\ \gamma \times \ln\frac{nu_{zl}}{nu_{zl_0}} & 0 \end{bmatrix} \tag{25}$$

The rows in UU_{il} and NU_{zl} correspond to the i_{th} user gaming strategies: accept or not, the columns correspond to the z_{th} network provider gaming strategies: provide or not. uu_{il_0} and nu_{zl_0} represent the lowest acceptable utilities of the i_{th} user and the z_{th} network provider on e_l. If the z_{th} network provider is willing to provide e_l but the i_{th} user rejects e_l or the i_{th} user is willing to accept e_l but the z_{th} network provider does not provide e_l, the i_{th} user or the z_{th} network provider should be punished. Here, μ is a penalty factor bigger than 1. If the z_{th} network provider does not provide e_l but the i_{th} user is willing to accept e_l or the i_{th} user rejects e_l but the z_{th} network provider is willing to provide e_l, the z_{th} network provider utility or the i_{th} user utility would be suffered, γ is a loss factor smaller than 1. If the z_{th} network provider does not provide e_l at the same time the i_{th} user rejects e_l, both utilities are 0. If the strategy pair $< p^*, q^* >$ satisfies the formula (26), it is a specific solution under Nash equilibrium [15]:

$$\begin{cases} uu_{p^*q^*} \geq uu_{pq^*} \\ nu_{p^*q^*} \geq nu_{p^*q} \end{cases} \tag{26}$$

If $< p^*, q^* >$ is $<$ accept, provide $>$, e_l will be selected, otherwise aborted, $p, q, p^*, q^* = 1,2$.

2.6 Being_selected_probability

Its initial value of e_l is calculated as follows:

$$pr_l^0 = \frac{1}{SE_l} \tag{27}$$

SE_l is the number of those edges sharing the same endpoints with e_l. All edges sharing the same endpoints have the same being_selected_probability initially. When routing, it is updated as follows:

$$pr_l^j = \frac{SS_j^2}{SS_j^2 + \theta_j^2} \tag{28}$$

$$SS_j = \begin{cases} \omega_1 SS_0 + \omega_2 \times \dfrac{\tau}{HP_j(sn(e_l), v_t)} & \text{the } j_{th} \text{ bee passing } e_l \\ SS_0 & \textit{otherwise} \end{cases} \tag{29}$$

$$\theta_j = \begin{cases} \omega_3 \theta_0 + \omega_4 \times \lambda \times HP_j(sn(e_l), v_t) & \text{the } j_{th} \text{ bee passing } e_l \\ \theta_0 + \sigma \times HP_j(sn(e_l), v_t) & \text{otherwise} \end{cases} \tag{30}$$

SS_j is stimulation signal value produced by the j_{th} bee, SS_0 is baseline value, $sn(e_l)$ is starting endpoint of e_l, v_t is another endpoint along the path having traversed by the j_{th} bee, $HP_j(sn(e_l), v_t)$ is hop count between $sn(e_l)$ and v_t along the path from v_s to v_t, θ_j is its own reaction valve value to v_{d_i} of the $(j+1)_{th}$ bee, θ_0 is its baseline value, ω_1, ω_2, ω_3 and ω_4 are preference weights, τ, λ and σ are constants, $\tau > 1$, $0 < \lambda, \sigma < 1$, $0 \leq \omega_1, \omega_2, \omega_3, \omega_4 \leq 1$, $\omega_1 + \omega_2 = 1$, $\omega_3 + \omega_4 = 1$.

2.7 Multicast Tree Evaluation

The i_{th} user utility UU_i^T, the z_{th} network provider utility NU_z^T, all user utility UU^T, all network provider utility NU^T along T_{sD}, the cost CT^T of T_{sD} and the comprehensive evaluation CE^T on T_{sD} are calculated as follows:

$$UU_i^T = \sum_{e_l \in T_{sD}} uu_{il} \times NE_{izl} \tag{31}$$

$$NU_z^T = \sum_{e_l \in T_{sD}} nu_{zl} \times NE_{izl} \tag{32}$$

$$UU^T = \sum_i UU_i^T \tag{33}$$

$$NU^T = \sum_z NU_z^T \tag{34}$$

$$CT^T = \sum_{e_l \in T_{sD}} ct_l \times abw_l \tag{35}$$

$$CE^T = \beta_1 \times \frac{\Omega_1}{UU^T} + \beta_2 \times \frac{\Omega_2}{NU^T} + \beta_3 \times \frac{CT^T}{\Omega_3} \tag{36}$$

$$NE_{izl} = \begin{cases} >1 & \text{Nash equilibrium} \\ 1 & \text{otherwise} \end{cases} \tag{37}$$

β_1, β_2 and β_3 are the preference weights on the user utility, the network provider utility and the tree cost respectively, $0 \le \beta_1, \beta_2, \beta_3 \le 1$, $\beta_1 + \beta_2 + \beta_3 = 1$; Ω_1, Ω_2 and Ω_3 are tuning factors, making Ω_1 / UU^T, Ω_2 / NU^T and CT^T / Ω_3 into the same magnitude order. According to the formula (36), the smaller the value of CE^T, the much possible for UU^T and NU^T to achieve or approach the Pareto optimum [16] under the Nash equilibrium, the much possible the minimum CT^T achieved or approached.

2.8 Mathematical Model

It is described as follows:

$$maximize\left\{UU_i^T\right\} \tag{38}$$

$$maximize\left\{UU^T\right\} \tag{39}$$

$$maximize\left\{NU_z^T\right\} \tag{40}$$

$$maximize\left\{NU^T\right\} \tag{41}$$

$$maximize\left\{UU^T + NU^T\right\} \tag{42}$$

$$minimize\left\{CT^T\right\} \tag{43}$$

s.t.

$$BW_{P_i} \ge BW_i^L \tag{44}$$

$$DL_{P_i} \le DL_i^H \tag{45}$$

$$JT_{P_i} \le JT_i^H \tag{46}$$

$$LS_{P_i} \le LS_i^H \tag{47}$$

$$BW_{P_i} = \min_{e_l \in P_i}\left\{abw_l\right\} \tag{48}$$

$$DL_{P_i} = \sum_{e_l \in P_i} dl_l \tag{49}$$

$$JT_{P_i} = \sum_{e_l \in P_i} jt_l \tag{50}$$

$$LS_{P_i} = 1 - \prod_{e_l \in P_i} \left(1 - ls_l\right) \tag{51}$$

P_i is the path from v_s to v_{d_i} in T_{sD}.

2.9 Algorithm Description

It is described as follows:

Step1: $IN = |D|$, $i = 1$.

Step2: If $i < IN$, go to Step3, otherwise go to Step20.

Step3: Set the maximum bee number BN, the sent bee number by far $j = 0$, the acceptable baseline bid bd_l^0 of each e_l, the bee sending period itv, the long distance bee life cycle ltv, the short distance bee life cycle stv, the hop counter $hp = 0$, the set of the constructed feasible paths from v_s to v_{d_i}: $PS_i = \varphi$; get pr_l^j for each e_l according to the formula (27). itv, ltv and stv are counted by hop.

Step4: If $j < BN$, go to Step5, otherwise go to Step18.

Step5: If hp can be divided by itv exactly, send a long distance Be_j from v_s, otherwise a short distance Be_j.

Step6: Set the initial information carried by Be_j as follows: bandwidth $bw = \infty$, delay $dl = 0$, delay jitter $jt = 0$, error rate $ls = 0$, the current node $cn = v_s$, the current path $P_c = \{v_s\}$, the set of those edges connecting with cn: $ne = \varphi$, the candidate edge set for the next hop: $ce = \varphi$.

Step7: Put all edges connecting with cn into ne; delete those edges from ne which cause loop, that is, $ne = ne - \{e_l | e_l \in ne \wedge an(e_l) \in P_c\}$, $an(e_l)$ represents another node of e_l except cn.

Step8: If Be_j is long distance and $hp > ltv$ or Be_j is short distance and $hp > stv$, Be_j died, go to Step17; otherwise, go to Step9.

Step9: If $ne = \varphi$, go to Step14, otherwise select one e_l from ne at random.

Step10: If $\min\{bw, abw_l\} < BW_i^L$ or $dl + dl_l > DL_i^H$ or $jt + jt_l > JT_i^H$ or $1 - (1 - ls)(1 - ls_l) > LS_i^H$, $ne = ne - \{e_l\}$, go to Step9; otherwise, get the loaded level of e_l by the formula (1).

Step11: If e_l is high-loaded, the i_{th} user determines its bid bd_i to e_l from $\{bd_{ie}, bd_{ig}, bd_{if}, bd_{ip}\}$ according to its Ql_{il} got by the formula (13), go to Step12; otherwise, get p_l according to the formula (2) or (3) if e_l is low-loaded or moderate-loaded, go to Step13.

Step12: If $bd_i < bd_l^0$, $ne = ne - \{e_l\}$, go to Step9; otherwise, $p_l = bd_i$, $bd_l^0 = bd_i$.

Step13: Get uu_{il} and nu_{zl} according to the formula (15) and (23), the user and the network provider play game on e_l according to section 2.5: if the Nash equilibrium is achieved and the gaming strategy is $< \text{accept,provide} >$, $ce = ce \cup \{e_l\}$ and $ne = ne - \{e_l\}$; otherwise, $ne = ne - \{e_l\}$ and go to Step9.

Step14: If $ce = \varphi$, Be_j died, go to Step17; otherwise, select one e_l from ce as the next hop by pr_l^j : $bw = \min\{bw, abw_l\}$, $dl = dl + dl_l$, $jt = jt + jt_l$, $ls = 1-(1-ls)(1-ls_l)$, $cn = an(e_l)$, $P_c = P_c \cup \{e_l\} \cup \{cn\}$, $ne = \varphi$, $ce = \varphi$, $hp = hp + 1$.

Step15: If $cn \neq v_{d_i}$, go to Step7.

Step16: $PS_i = PS_i \cup \{P_c\}$.

Step17: $j = j + 1$, update pr_l^j for each $e_l \in P_c$ according to the formula (28), go to Step4.

Step18: If $PS_i = \varphi$, routing failed and the algorithm ends.

Step19: $i = i + 1$, go to Step2.

Step20: Set the maximum number of the constructed feasible multicast tree TN , the current number of the constructed feasible multicast tree $tn = 0$, the current optimal multicast tree $T_{sD_{bt}} = \varphi$ and $CE^{T_{sD_{bt}}} = \infty$.

Step21: If $tn = TN$, go to Step25.

Step22: Take one path from each PS_i randomly and construct a feasible multicast tree $T_{sD_{tn}}$ with them. If there exists any cycle in $T_{sD_{tn}}$, do cycle elimination.

Step23: Calculate $CE^{T_{sD_{tn}}}$ according to the formula (36): if $CE^{T_{sD_{tn}}} < CE^{T_{sD_{bt}}}$, $T_{sD_{bt}} = T_{sD_{tn}}$ and $CE^{T_{sD_{bt}}} = CE^{T_{sD_{tn}}}$.

Step24: $tn = tn + 1$, go to Step21.

Step25: If $CE^{T_{sD_{bt}}} < \infty$, output $T_{sD_{bt}}$ as the problem solution, routing succeeded; otherwise, routing failed. The algorithm ends.

3. Performance Evaluation and Conclusion

Simulated implementation of the proposed QoS multicast routing scheme in this paper has been done on NS2 (Network Simulator 2). Some performance indexes have been evaluated, such as QoS multicast routing request succeeded rate (RSR), user utility (UU), network provider utility (NU), comprehensive utility (CU=UU+NU) and ratio of the Pareto optimum under the Nash equilibrium (RPN). Table 1 is the comparison results between the proposed scheme (denoted as B) and the SPF (Shortest Path First) based one (denoted as S) when doing simulation over the CERNET topology. It can be concluded that the proposed scheme has better performance than the SPF based one. In future, its prototype system will be developed to improve its practicality.

Table 1. Performance Comparison

Index	RSR	UU	NU	CU	RPN
B:S	1.36	1.65	1.54	1.62	2.12

4. ACKNOWLEDGMENTS

This work is supported by the National High-Tech Research and Development Plan of China under Grant No. 2006AA01Z214; the National Natural Science Foundation of China under Grant No. 60673159 and No. 70671020; Program for New Century Excellent Talents in University; the Key Project of Chinese Ministry of Education under Grant No. 108040; Specialized Research Fund for the Doctoral Program of Higher Education under Grant No. 20060145012 and No. 20070145017; the Natural Science Foundation of Liaoning Province under Grant No. 20062022.

5. REFERENCES

[1] Eva Gustafsson, Annika Jonsson, “Always Best Connected”, IEEE Wireless Communications. vol. 10, no. 1, pp. 49-55, February 2003.

[2] Dean H. L., Ariel O., “QoS Routing in Networks with Uncertain Parameters”, IEEE/ACM Transactions on Networking, vol. 6, no. 6, pp. 768-778, December 1998.

[3] Wang Z. , Crowcroft J., “Quality of Service Routing for Supporting Multimedia Applications”, IEEE Journal on Selected Areas in Communications, vol. 14, no. 7, pp. 1288-1294, September 1996.

[4] Li LY, Li CL, “A QoS Multicast Routing Protocol for Dynamic Group Topology”, Information Sciences, vol. 169, no. 1-2, pp. 113-130, January 2005.

[5] Lowu F, Baryamureeba V, “On Efficient Distribution of Data in Multicast Networks: QoS in Scalable Networks”, In: Lirkov I, eds. Large-Scale Scientific Computing. LNCS 3743, Berlin, Heidelberg: Springer-Verlag, pp. 518-525, February 2006.

[6] Wang L, Li ZZ, Song CQ, Yan Y, “A Dynamic Multicast Routing Algorithm with Inaccurate Information Satisfying Multiple QoS Constraints”, Acta Electronica Sinica, vol. 32, no. 8, pp. 1244-1247, August 2004.

[7] Li LY, Li CL, “Genetic Algorithm-Based QoS Multicast Routing for Uncertainty in Network Parameters”, In: Zhou X, eds. Web Technologies and Applications: 5th Asia-Pacific Web Conference. LNCS 2642, Berlin, Heidelberg: Springer-Verlag, pp. 430-441, January 2003.

[8] Chen P. Dong TL, “A Fuzzy Genetic Algorithm for QoS Multicast Routing”, Computer Communications, vol. 26, no. 6, pp. 506-512, April 2003.

[9] Crichigno J, Barán B, “A Multicast Routing Algorithm Using Multiobjective Optimization”, In: Souza D, eds. 11th International Conference on Telecommunications. LNCS

3124, Berlin, Heidelberg: Springer-Verlag, pp. 1107-1113, July 2004.

[10] Deb S, Srikant R, “Congestion Control for Fair Resource Allocation in Networks with Multicast Flows”, IEEE/ACM Transactions on Networking, vol.12, no. 2, pp. 274-285, April 2004

[11] Wang XW, Liu C, Cui JY, Huang M, Das SK, “A Fair QoS Multicast Routing Scheme for IP/DWDM Optical Internet”, In: McKinley P, eds. The 25th IEEE International Conference on Distributed Computing Systems Workshops. Los Alamitos: IEEE Computer Society, pp. 624-629, June 2005.

[12] Horst F. Wedde, Muddassar Farooq, Yue Zhang, “BeeHive: An Efficient Fault-Tolerant Routing Algorithm Inspired by Honey Bee Behavior”, Proceedings of the Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence (ANTS 2004), Lecture Notes in Computer Science, no. 3172, pp. 83-94, November 2004.

[13] Andrew Odlyzko, “Paris Metro Pricing: the Minimalist Differentiated Services Solution”, IEEE/IFIP IWQoS’99, pp. 559-561, June 1999.

[14] Yang GB, Gao YY, *Theory and Application of Fuzzy Mathematics.* 3rd ed. Guangzhou: South China University of Technology Press, pp. 76-79, 2001.

[15] Drew Fudenberg, Jean Tirole, *Game Theory* [M], Beijing: China Renmin University Press, pp. 10-23, 2003.

[16] F P Kelly, A Maulloo, D Tan, “Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability”, Operational Research Society, vol. 49, no. 3, pp. 237-252, March 1998.