論文2003-40TC-12-8

Real-Time Application의 효과적인 QoS 라우팅을 위한 적응적 Route 선택 강화 방법

(Reinforcement Method to Enhance Adaptive Route Search for Efficient Real-Time Application Specific QoS Routing)

吳 在 錫 *, 裵 晟 日 *, 安 鎭 浩 *, 姜 成 昊 *

(Jae Seuk Oh, Sung-Il Bae, Jin-Ho Ahn, and Sungh Kang)

요 약

본 논문은 real-time 어플리케이션들을 위한 보다 효과적이고 효율적으로 ant-like mobile agent들이 QoS metrics를 고려하여 네트워크상에서 목적지까지 가장 최적화된 route을 찾는 Ant 알고리듬을 바탕으로 한 QoS 라우팅 알고리듬에서의 route 선택 강화 계산방법을 제시한다. 시뮬레이션 결과 본 논문에서 제시하는 방법이 기존의 방법보다 delay jitter와 bandwidth를 우선으로 하는 real-time application에 대한 가장 최적화된 route을 보다 효과적이고 보다 네트워크 환경에 적응적으로 찾아내는 것을 확인하였다.

Abstract

In this paper, we present a new method to calculate reinforcement value in QoS routing algorithm targeted for real-time applications based on Ant algorithm to efficiently and effectively reinforce ant-like mobile agents to find the best route toward destination in a network regarding necessary QoS metrics. Simulation results show that the proposed method realizes QoS routing more efficiently and more adaptively than those of the existing method thereby providing better solutions for the best route selection for real-time application that has high priority on delay jitter and bandwidth.

Keywords : Reinforcement value, Ant algorithm, mobile agent, QoS routing, Real-time applications

Ⅰ. Introduction

As the technology in computers and communications improves in fast-pace mode, the worldwide demands for faster, secure and better communications networks services grow as well. Just a decade ago, much fewer people were getting various information electronically or online via Internet or any other types of electronic means, and more were depended on offline resources, such as books and newspapers to gather needed information.Contents provided by various Internet Service Providers (ISOs) were limited and slow in service compare to those of now. But soon as more people started to realize the conveniences of Internet services and email, the

---

* 正會員, 延世大學校 電氣電子工學科

(Yonsei University, Electrical and Electronic Engineering)

※ This work was supported by the Brain Korea 21 Project in 2003

接受日字:2003年11月8日, 수정완료일:2003年12月3日

importance of online communications grew exponentially.And it eventually led to demand and supply of faster and better services.

Nowadays, it is very difficult to find a computer using a modem to get an Internet access via a phone line, which was common just a few years back.However, the lack of expandability of phone lines, in terms of speed, made ISOs to search for faster means, and people switched off from slower modem to faster Internet access using cable modem, Home LAN, VDSL and etc. with no more dial-ups.Faster speed and 24/7 access to Internet led to an emergence of new service contents such as Video on demand (VOD), real-time Video conference, peer-to-peer connection, and instant messaging services.The world is no longer far apart, but it's just a click away.

As Internet expands, the demand for real time and quality of services (QoS) in a network increases. The quality of services are sensitive to the network's characteristics such as bandwidth, delay, delay jitter, packet loss and cost depending on the type of applications. Furthermore, the use of multiple metrics is needed to better characterize a network and to support a wide range of QoS requirements[1].

Among many different kinds of routing algorithms, there is an algorithm called, Ant algorithm, which is inspired from the adaptive and distributive behaviors of real ants of locating the fastest route to the food source by depositing a chemical substance called, pheromone, on the trail during the trip to the destination. Furthermore, this particular form of indirect communication is typical among social insects to coordinate their activities, and it is called stigmergy[2]. This algorithm is proved to be very effective in terms of solving Traveling Salesman Problem[3], Quadratic Assignment Problem[4], and etc However, thus far, there haven't been many researches that are done to realize QoS routing based on Ant algorithm, leaving large room for improvements and explorations.

The main purpose of this paper is to propose a new method to calculate reinforcement value reflecting all the necessary QoS metrics to better realize an adaptive behavior of Ant algorithm for real-time applications.

This paper is organized as follows. In section Ⅱ, QoS routing algorithm based on Ant algorithm is introduced. In section Ⅲ, detailed description of the proposed method of reinforcement calculation under the QoS routing algorithm described in section Ⅱ is introduced. In section Ⅳ, experimental results are presented; and at last, conclusion from this research is drawn out in section Ⅴ.

Ⅱ. Conventional QoS Routing Algorithm Based on Ant Algorithm

Ant algorithm is an evolution algorithm that mimics the distributive, adaptive, and most of all social behaviors of real ants. Real ants are capable of finding the shortest path from a food source to their nest without using visual cues by exploiting pheromone information[3]. Here, pheromone is a chemical substance that each ant deposits on the trail to mark its path as well as to attract other ants to follow the same path toward the identical destination. Initially, ants take various paths to their destination, the food source, because none or an insignificant amount of pheromone to attract ants is deposited on any of the possible paths as shown in Figure 1.As

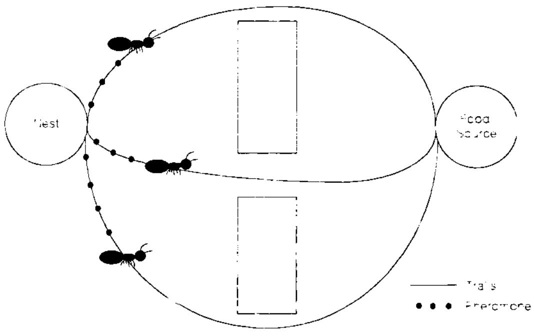


그림 1. 식량 (목적지)까지의 거리를 찾을 때의 개미의 초기 검색

Fig. 1. Initial search of ants for the shortest path to food source(destination).

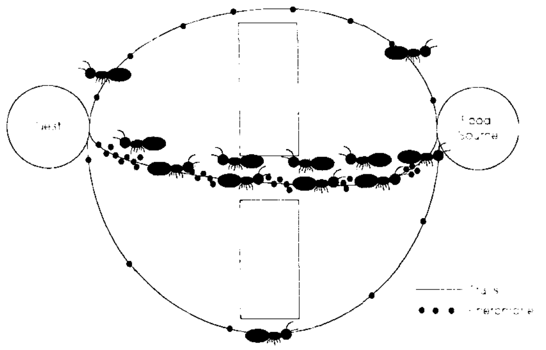


그림 2. 여러번의 반복적인 검색을 통한 식량 (목적지)까지의 최적화된 행로 발견

Fig. 2. After certain number of trips ants found the best path to food source(destination)

more and more ants travel to the same location, a path with the shortest distance between the nest and the food source gets more pheromone deposits. This is due to what is called pheromone decay. Pheromone decays with time; therefore, longer the distance means more pheromone gets decayed in that path. Therefore, a path with the shortest trip time or the shortest distance eventually left with the most pheromone, which, in turn, attracts more and more ants as shown in Figure 2.Now ants have established or found the shortest path to their destination.

In Ant algorithm, there are two types of mobile agents, forward and backward ants that mimic the aforementioned behaviors of ants. The forward ant is first sent out from a source node to its randomly selected destination, and it gathers the network information of the path it took to reach the destination, which will be used to update routing table later on. At each node, the forward ant locally selects a next neighbor node to hop in accordance with probability values of all links connecting the current node to its neighbor nodes that are stored in a routing table. Therefore, a link with the highest probability to be taken to go to the destination node has the best chance to be selected as the next node to visit toward destination. However, it does not guarantee the best edge to be chosen at every single time, which, in turn, allows ants to adapt their exploration activity to the varying data traffic distribution[1, 5, 6].

The backward ant is created at the destination node once the forward ant safely arrives. The backward ant literally goes back to the source node by taking the same path that the forward ant took. On its way back, it pops the stored network information to know which node to travel next and also to update routing table of the node it is currently standing. The backward ant repeats this until it arrives back at the source node, and dies. This is very brief and basic explanation of Ant algorithm and with this in mind, QoS routing algorithm based on Ant algorithm in[1] is as follows. There are T sets of ants where every set is consisting of M types of ants belonging to M different call requirements, where each ant type must find the best path to its destination that satisfies all the requirements. Furthermore, the properties of pheromone deposits of each ant type are different from each other, so that an ant selects a route relying only upon the pheromone deposited by of same ant type. Since the major tasks of ant algorithm are to choose a path in accordance with pheromone deposits, and to adjust pheromone amount laid on the trail, the state transition rule and the pheromone updating rule are proposed in[1].

Under the state transition rule proposed in[1], a d type of an ant at node r selects a next node s to travel according to following rule

If $q \le q_0$, then

$$\rho_d(r,s) = \begin{cases} 1 & \max(phero(d,r,s)),\ s \in J_d(r) \\ 0 & otherwise \end{cases} \tag{1}$$

otherwise,

$$\rho_d(r,s) = \begin{cases} \dfrac{phero(d,r,s)}{\sum_{u \in J_d(r)} phero(d,r,u)} & s \in J_d(r) \\ 0 & otherwise \end{cases} \tag{2}$$

where, q_0 represents a constant value that lies between 1 and 0, which is used to compare with q, a randomly chosennumber between 1 and 0, to determine how often either (1) or (2) is used to determine the probability of choosing the next node from the current node out of all neighbor nodes that lead to the destination. In other words, q_0 represents percentage of choosing equation (1) or (2). For example, suppose q_0 is 0.7, then about 70 % of times a dtype ant will choose the next node according to the probabilities set up by equation (1), and the other 30% of times the next node will be chosen by equation (2). $\rho_d(r,s)$ represents probability of choosing node s as the next node to go to; *phero(d,r,s)* represents amount of *d* type pheromone deposited on the edge between node *r* and node *s* and *Jd(r)* represents neighbor nodes of the current node *r* that *d* typeant has not passed during its trip toward the destination.

The pheromone-updating rule is further divided into two sub-rules: local updating rule and global updating rule, the concept which was first proposed in[3]. Local measures are related to the variation that a given node introduces in the QoS characteristic; whereas, global measures are related to the absolute values expected for the QoS characteristic at a given point[7]. Measurements of non-cumulative QoS characteristics are always global; however, on the contrary, with cumulative QoS characteristics, measurements could be global or local[7].

Under the local updating rule, suppose a *d* type ant at node *r* chooses a neighbor node s as the next node to travel, the amount of pheromone *phero(d,r,s)* is adjusted in accordance with equation (3), otherwise no pheromone amount gets adjusted.

$$phero(d,r,s) = (1-a_0) \bullet phero(d,r,s) + a_0 \bullet cons \qquad (3)$$

where, a_0 is a value between 0 and 1, and cons is a constant. In this way, ants will make a better use of pheromone information; without the local updating, all ants would search in a narrow neighborhood of the best previous path[1].

The global updating rule is used when the globally best path to the destination is found. Once the globally best path is determined, pheromone amounts ofedges between all nodes in the globally best path are adjusted in accordance with equation (4), and pheromone amounts of all the other edges get adjusted by equation (5)

$$phero(d,r,s) = (1-a_1) \bullet phero(d,r,s) + a_1 \bullet F \qquad (4)$$

$$phero(d,r,s) = (1-a_1) \bullet phero(d,r,s) \qquad (5)$$

where, a1 is a value between 0 and 1, and Fis the cost function, and it plays the same role as the reinforcement value in *AntNet*[8] for adjusting pheromone amount deposited in the path in accordance with the network conditions experienced by an ant that found the globally best path toward the destination. According to[8], the reinforcement is a critical quantity that has to be assigned by considering three aspects listed below:

i) Paths should receive an increment in their selection probability proportional to their goodness;

ii) The goodness is a relative measure, which depends on the traffic conditions, that can be estimated by means of the local traffic model, and;

iii) It is important not to follow all the traffic fluctuations. This is important since uncontrolled oscillations in routing tables are one of the main problems in the shortest paths routing[8].

The purpose of the global update rule is to search the globally best resolution[1]. Furthermore, the value of F is calculated by following equations:

$$F = -F_1 + F_2 \qquad (6)$$

where,

$$F_1 = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} LC_{ij} \bullet P_{ij}^{d} \qquad (7)$$

$$F_2 = A \bullet \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} H(Z_{ij}) + B \bullet H(Z_2) + C \bullet H(Z_3) \qquad (8)$$

where, N is the node number, A, B and C are positive real coefficients to indicate importance of each term in reinforcement calculation according to the QoS constraints. Furthermore, F_1 represents the total cost of the route selected by an ant; and F_2 represents the QoS constraints[1].

H(Z) = 0, if Z<0, otherwise H(Z) = Z.

$$Z_{ij} = P_{ij}^{d} \bullet LB_{ij} - B_w \tag{9}$$

$$Z_2 = D_w - (\Sigma\Sigma LD_{ij} \bullet P_{ij}^{d} + \Sigma N_i^{d} \bullet ND_i) \tag{10}$$

$$Z_3 = \prod_{i=1}^{N}(1 - N_i^{d} \bullet NL_i) - (1 - L_w) \tag{11}$$

Here, P_{ij}^{d} = 1 if an edge between node i and node j is an edge in the d type ant selected route, otherwise P_{ij}^{d} = 0. N_i^{d} = 1 if node i is the node in the d type ant selected route, otherwise N_i^{d} = 0. Symbols LB_{ij}, LC_{ij} and LD_{ij} are the available bandwidth, cost and delay of an edge between node i and node j respectively, and Lw, Bw and Dw represent link, bandwidth and delay constraints respectively

Following the aforementioned rules, the algorithm takes the following steps. At first, if actual end-to-end delay jitter is greater than the delay jitter constraint, then routing fails; if not, then next step is to eliminate the link that does not satisfy the bandwidth constraint. These first two steps are the modifications to speed up the algorithm for real time applications. Next, the amount of pheromone deposited on every edge in the network topology is initialized for every type of ants. Then a set of ants of every type is sent out one at a time at constant interval toward corresponding destinations to collect network information, while choosing its path by repeatedly applying the state transition rule. After the set of ants has chosen the paths between their own source nodes and destination nodes successfully, the amount of every type of pheromone on every path is adjusted by using the local updating rule[1]. The previous step gets repeated for another set of ants, until all sets of ants finish the step. Next, choose the globally best ant of each type, and then use the global updating rule to adjust amount of pheromone on each path in the route that is selected by the best ant of each type[1]. Finally, the steps after the initialization step are repeatedly performed, until the accuracy requirement is satisfied[1].

Ⅲ. Proposed Method for Efficient Route Reinforcement

As described in the previous sections,[1] calculates the differences between actual QoS measurements and QoS constraints in the process of calculating reinforcement value. However, the equations described in the previous section do not realize the adaptive behavior well enough to demonstrate adaptiveness of Ant algorithm. Unlike[1], the proposed reinforcement calculation uses ratios between QoS measurements and QoS constraints. The proposed method to calculate the reinforcement value is as follows.

If $(F_2 >= 1)$,

$$F = F_2 - k(\frac{F_1}{F_2}) \ . \tag{12}$$

else

$$F = F_2 - k \bullet F_1 \ . \tag{13}$$

$$F_1 = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} LC_{ij} \bullet P_{ij}^{d} \ . \tag{14}$$

$$F_2 = A \bullet \min(Band_{ij}) + B \bullet Dly + C \bullet PLR + D \bullet DJ \ . \quad \textit{where } i \neq j, \textit{ and } i \in P \textit{ and } j \in P, \quad P = \textit{All nodes in the chosen path} \tag{15}$$

where, F is the cost function or the reinforcement value F_1 is the total cost of the route; F_2 is the QoS constraints; and k is weight constant for cost to indicate its importance compare to other QoS metrics.

$$Band = \begin{cases} \dfrac{B_{mea}}{B_w}, where\ B_{mea} >= B_w \\ \left(\dfrac{B_{mea}}{B_w}\right)^2, where\ toleration\ rate =< \dfrac{B_{mea}}{B_w} < 1 \\ 0, \quad where\ \dfrac{B_{mea}}{B_w} < toleration rate \end{cases} \quad (16)$$

$$Dly = \begin{cases} \dfrac{D_w}{\sum D_{mea}}, where \sum D_{mea} =< D_w \\ \left(\dfrac{D_w}{\sum D_{mea}}\right)^2, where\ toleration\ rate =< \dfrac{D_w}{\sum D_{mea}} < 1 \\ 0, \quad where \dfrac{D_w}{\sum D_{mea}} < toleration\ rate \end{cases} \quad (17)$$

$$PLR = \begin{cases} \dfrac{1-(1-L_w)}{1-\prod(1-L_{mea})}, where \prod(1-L_{mea}) >= (1-L_w) \\ \left(\dfrac{1-(1-L_w)}{1-\prod(1-L_{mea})}\right)^2, where\ toleration rate < \dfrac{1-(1-L_w)}{1-\prod(1-L_{mea})} < 1 \\ 0, \quad where \dfrac{1-(1-L_w)}{1-\prod(1-L_{mea})} < toleration rate \end{cases} \quad (18)$$

$$DJ = \begin{cases} \dfrac{J_w}{\sum DJ_{mea}}, where \sum J_{mea} =< J_w \\ \left(\dfrac{J_w}{\sum J_{mea}}\right)^2, where\ toleration\ rate =< \dfrac{J_w}{\sum J_{mea}} < 1 \\ 0, \quad where \dfrac{J_w}{\sum J_{mea}} < toleration rate \end{cases} \quad (19)$$

Among equations (12) through (19), equation (15) calculates the amount of positive influence the QoS measurements put on the reinforcement calculation by considering the goodness of each QoS measurement compare to its constraint, and their rate of importance in the calculation[1]. The goodness of each QoS measurement is calculated using equations (16) through (19).

Equation (16) shows the goodness calculation of bandwidth measurement. Since actual bandwidth measurement needs to be greater than or equal to its constraint, itsgoodness is measured by dividing the value of actual bandwidth measurement by the constraint. Thus, the goodness of bandwidth changes in accordance with its proportional value to the constraint. According to[7], the bandwidth is a non-cumulative QoS characteristic where the definition of non-cumulative QoS characteristics given in[7] is the ones for which the final influence of a set of consecutive nodes equals the worst influence of each node taken separately. So, when dealing with non-cumulative QoS characteristics, all modules have exactly the same performance responsibilities[7]. Such aspect had not been applied in[1]. However, according to equations (15) and (16) of the proposed method, the goodness of bandwidth that influences the calculation of F2, which is the total amount of goodness of the QoS measurements, is the worst influence taken in the trip. For instance, suppose the bandwidth constraint equals 90, andthe bandwidths of all edges of the chosen path are 100, 110, 90, and 70, then the value of the first term of equation (15) will be 0, since the bandwidth of 70 does not satisfy the constraint.

Equation (17) is for calculating the goodness of the end-to-end delay measurement in comparison with the constraint. Similar to equation (16), the goodness is measured by calculating the fractional value of the constraint compare to the actual measurement. In other words, the goodness is measured by dividing the delay constraint by the end-to-end delay, since lower the end-to-end delay the better.

Equation (18) is for calculating the goodness of the packet loss rate in comparison with its constraint. The packet loss rate is transformed to an equivalent metric that follows the multiplicative composition rule according to the definition given in[9]. The goodness is also measured as the fractional representation of the ratio between the multiplicative value of the packet loss rate measurement and the constraint.

Last but not least, equation (19) is for calculating the goodness of the delay jitter measurement in comparison with its constraint. This is also measured as the fractional representation of the ratio between the constraint and the measurement. In[1], this term was not presented to calculate reinforcement value, since it was taken outside the ant algorithm to speed up the algorithm. Delay, delay jitter, cost and packet loss rate are cumulative QoS characteristics. Cumulative characteristics include the ones for which the influence of a set of consecutive modules equals the sum of the influence of each module taken separately[7].

Equations (16) through (19) have term "toleration rate." Toleration rates are set individually for each QoS metric with values between 0 and 1. Each toleration rate represents percentage of negative discrepancy that the QoS metric can tolerate. Constraints are the boundaries where the QoS measurements have to meet to guarantee the quality of service. However, in the case of having no paths satisfying the QoS constraints, toleration rate can be used to find a path that provides a decent level of quality of service but with some service degradation; or QoS constraints are missed by a very small margin.

Toleration rate of 1 is as saying there is no room for an error. Toleration rate less than 1 means there are room for less qualified paths; however, squaring the fractional value of the goodness of the measurement that is less than 1 but satisfies the toleration rate discriminates its positive influence on the reinforcement calculation to be less than that of values greater than or equal to 1. When the fractional value of the ratio between the measurement and the constraint does not satisfy the toleration rate, the goodness is set to 0 to indicate no influence this measurement adds on the reinforcement calculation since the measurement does not satisfy its constraint nor the toleration rate.

Furthermore, since the goodness of each QoS metric is calculated as the ratio between the actual measurement and the constraint that the goodness amount of each QoS term changes dynamically with the constraints and the measurements, which, in turn, reflects the network condition better in terms of deciding the best route. Suppose there aretwo cases where in one, delay constraint is 5 and actual end to end delay is measured as 4; and in other case, delay constraint is 100 and actual end to end delay of 99. In both cases, actual delay measurements are 1 unit of delay better than their constraints; however, 1 unit of delay must have higher impact on the former case than the latter one, since the delay size of the former is much less than that of latter. However, under the calculation introduced in[11], 1 unit of delay always has the same impact on the reinforcement calculation regardless of the delay size.

Equations (12) and (13) are the top level calculation of reinforcement. Equation (12) subtracts the total amount of goodness of all QoS measurements by the fractional value of the ratio between the total cost of the chosen path and the total amount of goodness of all QoS measurements. The reason for dividing the total cost by the total amount of goodness is to be able to control the importance of cost in relationship with other QoS metrics in reinforcement calculation. Otherwise, the cost would always be the major factor for determining the best route. Furthermore, under the proposed equation (12) other QoS metrics can be the major factors to determine which path is the best amongst all other possible paths exist between the source and the destination.

Equation (13) is for when F_2 less than 1. For example, if there exists a toleration rate less than 1; and all QoS measurements, except for the one that satisfies its toleration rate of lessthan 1, fail to satisfy the requirements, then F_2 might be getting a value less than 1. If this were the case, using equation (12) would resulted in higher negative impact the second term has on the reinforcement calculation than the cases where F_2 is zero. However, even though the value of F_2 is a value greater than zero, it is not good enough to overcome the total cost. Therefore, negative reinforcement is applied to diminish the pheromone deposited in the path to indicate that this path is not a preferable path to take; henceforth increases the likelihood of agents choosing different routes.

In addition to the reinforcement calculation, the global update rule is also proposed to get the best out of the proposed reinforcement calculation.

$$phero(d,r,s) = (1-a_1) \bullet phero(d,r,s) + c \bullet F \quad (20)$$

where, c is coefficient of value between 0 and 1.

Equation (20) is used to adjust pheromone amount on the globally best route, and equation (21) shown below is a new method to adjust pheromone amount of less qualified paths that were chosen in the process.

$$phero(d,r,s) = (1-a_1)\bullet phero(d,r,s) - d\bullet(F - F_{others}) \quad (21)$$

Here d is weight coefficient and F_{others} is the reinforcement value of less qualified path. Since the reinforcement value under the proposed method is relative to the delay size, if there exist some routes that have reinforcement values almost as good as the globally best route, then the pheromone amounts of the nearly good routes reduce at a rate close to $(1-a_1)\bullet phero(d,r,s)$.

Along with changes in the global update rule, some changes were made to the state transition rule as well. As described earlier, in the algorithm of[1], the probability of an ant choosing a neighbor node out of all the neighbor nodes of the current node as the next node to visit is set in two ways. One is to give probability of 1 to the edge that has the maximum pheromone deposit, otherwise set to 0. Thus, in this case, an edge with the maximum pheromone gets chosen all the time; therefore, no other edges can be ever chosen when this is the case. The other way is to set probability of an edge by dividing the pheromone amount of the edge by the sum of the pheromones of all edges which have not been passed in the trip. In the proposed method, random-proportional rule introduced in[3, 10] is adopted, which is as shown below.

$$\rho_d(r,s) = \begin{cases} \dfrac{[phero(d,r,s)]^{\alpha}\bullet[\eta_{rs}]^{\beta}}{\sum[phero(d,r,s)]^{\alpha}\bullet[\eta_{rs}]^{\beta}}, & if\ s \in allowed\ next\ node\ to\ travel \\ 0, & otherwise \end{cases} \quad (22)$$

where, $\eta_{rs} = \dfrac{1}{d+pl}$ (d = link delay, pl = packet loss rate), r and s represent current node and next node, and are parameters which determine the relative importance of pheromone versus delay and packet loss rate[3]. By multiplying heuristic value of η_{rs}, we favor the choice of edges, which have less delay and packet loss rate, and have greater amount of pheromone, since the delay and the packet loss rate affect the probability.

Ⅳ. Simulation

Fig. 3 depicts the topology of a network system, which consists of eight nodes and twelve edges connecting the nodes. This topology is adopted from the simulation environment used in[1] to ensure that the simulation results of the existing method in this experiment agree with the results presented in[1].

The values in the parenthesis near each node in Fig. 3 represent node delay, packet loss rate and node delay jitter respectively; whereas, the values in the parenthesis near each edge represent link cost, bandwidth and link delay respectively.

The QoS requirements are set to B_w = 70, D_w = 8, L_w = 10^{-5} and J_w = 3. For the simulation of[1], the simulation parameters are set to T (sets of ants) = 8, M(call requests) = 3, a_0 = 0.069, a1 = 0.079, cons = 0.32, q_0 = 0.20, A = 0, B = 10 and C = 15. For the simulation of the proposed method, most of the simulation parameters are set exactly same as in the

표 1. 최적화된 행로을 찾기위한 평균 반복 수행수

Table 1. Average iterations to find the globally best route.

Routing Requests (s, d)	Existing method of [1]		Proposed method w/ TR=0.99		Proposed method w/ TR=1	
	Selected Route	Avg. Iter.	Selected Route	Avg. Iter.	Selected Route	Avg. Iter.
(0, 5)	0 → 1 → 3 → 5 0 → 1 → 3 → 5	62.2	0 → 1 → 3 → 5	31.4	0 → 1 → 3 → 5 0 → 1 → 3 → 5	36.9
(1, 5)	0→1→3→5	27.2	0→1→3→5	25.2	0→1→3→5	25.5
(1, 6)	0→1→3→5	36.8	0→1→3→5	31.4	0→1→3→5	31.5
(2, 6)	0 → 1 → 3 → 5	48.7	0 → 1 → 3 → 5	29.7	0 → 1 → 3 → 5	29.2
(2, 7)	0 → 1 → 3 → 5	30.6	0 → 1 → 3 → 5	28.5	0 → 1 → 3 → 5	28.3
(5, 7)	0 → 1 → 3 → 5	32.6	0 → 1 → 3 → 5	27.7	0 → 1 → 3 → 5	27.1
(4, 7)	N/A	N/A	. . . .	27.5	0 → 1 → 3 →	27.8

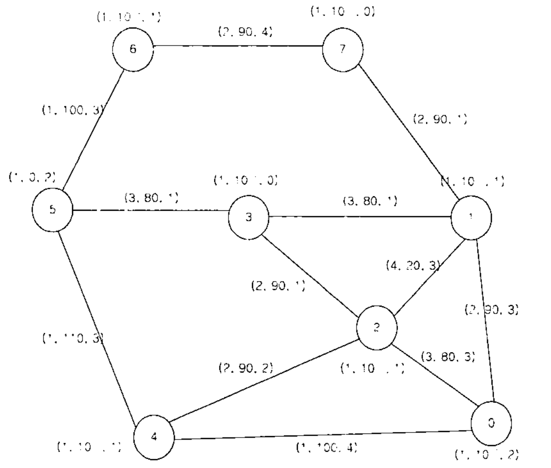


그림 3. 시뮬레이션에 이용된 네트워크 분포 모델과 QoS 변수[1]
Fig. 3. Network Topology model and its parameters used for simulation[1].

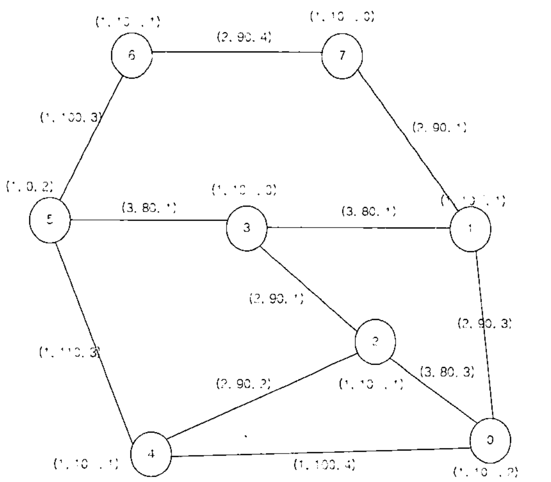


그림 4. QoS constraints에 따라 필터된 네트워크 분포도와 QoS 변수
Fig. 4. The filtered topology and its parameters[1].

existing method; however, since the calculation method is different from the existing method, some new parameters are introduced. And those parameters are set to C = 0.01, D = 0, k = 1, c = 0.079, c_l = 0.02, d = 0.01, α= 1and β= 1; and these are set relative to[1].

In accordance with the algorithm in[1], edges with bandwidth smaller than the constraint are filtered, andthe filtered topology is shown in Fig.4. In the

표 2. 실행된 행로들의 기존 방법과 제안한 방법 사용시의 강화 값 비교
Table 2. Comparisons of reinforcement values between the existing method and the proposed method for all routing requests.

Routing Requests (s, d)	Reinforcement value of the best route under [1]	Reinforcement value of the best route under the proposed method
(0, 5)	2	10.7286
(1, 5)	44	26.4518
(1, 6)	16	13.1356
(2, 6)	4	10.914
(2, 7)	23	15.5625
(5, 7)	22	15.5
(4, 7)	-9	9.1

given topology, the edge between node 1 and node 2 is eliminated due to the filtering process.

At first, several unicast routing requests: node 0 to node 5, node 1 to node 5, node 1 to node 6, node 2 to node 6, node 2 to node 7, node 5 to node 7, and node 4 to node 7 are assumed. By simulation, the globally best routes are found for each method as shown in Table 1. Here columns 'selected route' represent the globally best route chosen for each routing request, and 'average iteration' columns represent number of iterations to have all ants to converge to the globally best route that are chosen through repetitive process.

Compare to the simulation results of the routing requests (0,5), (2,6) and (4,7), the simulation results of all other requests do not seem to show the advantage of the proposed method has over the existing method since the improvements in average iteration to converge are small. However, it is very significant as well as important to understand what is underneath. Under the existing method, the reinforcement value changes at much greater rate than the proposed method; therefore, under the same network information, the reinforcement value of a globally best route is normally greater than that of the proposed method. In other words, pheromone in the globally best route gets adjusted with higher value under the existing method than the proposed method, resulting in faster convergence. For instance, the reinforcement values of the globally best routes of the routing requests (1,5) and (2,7) under the existing method are 44.00 and 23.00 respectively; and 26.45 and 15.56 respectively for the proposed method as shown in Table 2. Although the reinforcement values are greater under the existing method, according to Table 1 the proposed method converged faster. This is due to the adaptive behavior that is realized in the proposed method of the global update using equations (20) and (21).

표 3. 두개의 다른 constaint를 적용했을 경우의 (1, 5) 행로에서의 시뮬레이션 결과
Table 3. Simulation result of routing request (1,5) under two different constraints.

	Existing method of [1]		Proposed method	
	Selected Route	Average Iteration	Selected Route	Average Iteration
Under Small Delay Constraint ($D_w = 8$)	0→1→3→	27.63	0→1→3→	25.22
Under Large Delay Constraint ($D_w = 98$)	0→1→3→	27.47	0→1→3→	34.53

표 4. 마지막 실험 시뮬레이션 결과
Table 4. Simulation result of the last experiment.

Routing Requests (s, d)	Under large delay constraint			
	Existing method in [1]		Proposed method	
	Selected Route	Average Iteration	Selected Route	Average Iteration
(1, 5)	0→1→3→	27.47 (27.47)	0→1→3→	28.53 (34.53)

Next, an experiment is performed to see how the proposed method dynamically adjusts the reinforcement value, by comparing the simulation results of one with small delay constraint and the other with large delay constraint, thereby identifying the advantages of the proposed method.

To simulate one with small delay constraint, the QoS information on edges between node 5 and 6, and node 6 and 7 are changed to (1, 100, 0) and (2, 90, 1) respectively; and to simulate one with large delay constraint, the QoS information on edges between node 0 and 1, 1 and 3, 2 and 3, 1 and 7, 5 and 6, and 6 and 7 are changed to (2, 90, 93), (3, 80, 91), (2, 90, 91), (2, 90, 90), (1, 100, 1) and (2, 90, 1) respectively. Simulation results of both figures under the existing method and the proposed method are shown in Table 3.

Assuming the routing request of (1,5), the globally best route should be $1 \to 3 \to 5$ since it satisfies the constraints and has the smallest delay measurement. However, the route $1 \to 7 \to 6 \to 5$ also satisfies the constraints and has the end-to-end delay of 4 and 94, just one unit of delay more than $1 \to 3 \to 5$ in both cases. If we look at Table 3, the average iterations of two different cases under the existing method are nearly identical to each other. However, the average iterations of two different cases under the proposed method show some discrepancy.

The reason for such discrepancy is due to how the reinforcement values are calculated in the methods. Under the existing method, amount of impact a unit of delay exerts on the reinforcement value is constant regardless of delay size. However, in the proposed method, such fact is realized, and that is why it takes more iterations to converge when the delay size and the constraint are large.

One last experiment is done using network topology with edges between node 0 and 1, 1 and 3, 2 and 3, 1 and 7, 5 and 6, and 6 and 7 having QoS information of (2,90,93), (3,80,91), (2,90,99), (2,90,90), (1,100,1) and (2,90,10) respectively. Assuming the routing request (1,5), the simulation result is compared with that of the previous experiment under large delay constraint.

If we compare the results in Table 3 with Table 4, the average iteration of[1] is the same, while the average iteration of the proposed method is reduced. The reason for such reduction in number of iteration under the proposed method is due to equation (21), which reflects more of an adaptive behavior. Since the reinforcement value under the proposed method is relative to the delay size, if there exists some routes that have reinforcement values almost as good as the globally best route, then the pheromone amounts of the nearly good routes reduce at a rate close to $(1-a_1) \bullet phero(d,r,s)$, which was the case for the previous experiment.

V. Conclusions

Ant algorithm is a routing algorithm, realizing adaptive and social behavior of ants of finding the best route to the food source from the nest by indirect communications between ants using a chemical substance called pheromone.

This paper has presented with an adaptive method to reinforce route in real-time application specific QoS routing algorithm based on Ant algorithm. Unlike in the existing method, under the proposed method, the reinforcement value is calculated using the fractional ratios between the measurements and the constraints, which, in turn, provides with the reinforcement value relative to the measurement's size and the constraint's range. This way a unit of change in a QoS measurement adds different amountof influences on the reinforcement value depending upon the measurement's size and the constraint's range. And with further changes in the global update equations, the proposed method shows more adaptive behaviors than the existing method, providing faster convergence time.

참 고 문 헌

[1] Zhang, S., Liu, Z.: "A QoS Routing Algorithm Based on Ant Algorithm," 25th Annual IEEE Conference on Local Computer Networks, pp. 574 578, (2000).

[2] M. Dorigo, E. Bonabeau, & G. Theraulaz, "Ant Algorithms and Stigmergy," Future Generation Computer Systems, Number 16, pp. 851-871, 2000.

[3] Dorigo, M., Gambardella, L. M.: "Ant colony system: A cooperative learning approach to the traveling salesman problem," IEEE Trans. on Evolutionary Computation, pp. 53-66, (1997).

[4] Stutzle, T., Dorigo, M.: "ACO algorithms for the quadratic assignment problem," New Ideas in Optimization, pp. 33-50, McGraw Hill, (1999).

[5] G. Di Caro, & M. Dorigo, "An adaptive multi-agent routing algorithm inspired by ants behavior," Proc. Intelligent Agents for Telecommunications Applications 1998, 1998.

[6] G. Di Caro, & M. Dorigo, "Mobile Agents for Adaptive Routing," Proc. of the 31stInternational Conference on Systems Sciences, The Big Island of Hawaii, Jan. 1998.

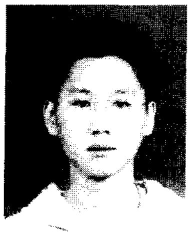
[7] G. Quadros, E. Monteiro, & F. Boavida, "A QoS Metric for Packet Networks," Proceedings of SPIE International Symposium on Voice, Video, and Data Communications Conference, Nov. 1998.

[8] M. Dorigo, & G.. Di Caro, "AntNet: Distributed Stigmergetic Control forNetworks," Journal of Artificial Intelligence Research, Number 9, pp. 317 365, 1999.

[9] Z. Wang, & J. Crowcroft, "Quality of Service Routing for Supporting Multimedia Applications," JSAC, pp. 228 1234, 1996.

[10] C. Chu, J. Gu, X. Hou, & Q. Gu, "A Heuristic Ant Algorithm for Solving QoS Multicast Routing Problem," Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on, Vol. 2, pp. 12-17, 2002.

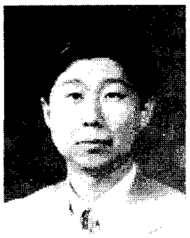
저 자 소 개



吳 在 錫(正會員)
2001년 5월 : University of Bridge-port, Computer Engineering 졸업 (B.S.). 2002년 3월~현재 : 연세대학교 전기전자공학과 석사과정

裵 晟 日(正會員)
1998년 2월 : 경북대학교 전자공학부 졸업(학사). 1998년 3월~2000년 2월 : 연세대학교 전기전자공학과 졸업(공학석사). 2000년 3월~현재 : 연세대학교 전기전자공학과 박사과정

安 鎭 浩(正會員)
1995년 2월 : 연세대학교 전기공학부 졸업(학사). 1995년 3월~1997년 2월 : 연세대학교 전기공학과 졸업 (공학석사). 1997년~2002년 8월 : LG 전자 DTV 연구소. 2002년 9월~현재 : 연세대학교 전기전자공학과 박사과정



姜 成 昊(正會員)
1986년 2월 : 서울대 공대 제어계측공학과 졸업. 1988년 5월 : The University of Texas at Austin 전기 및 컴퓨터공학과 졸업(공학석사). 1992년 5월 : The University of Texas at Austin 전기 및 컴퓨터공학과 졸업(공학박사), 미국 Schlumberger 연구원, Motorola 선임 연구원. 현재 : 연세대학교공과대학 전기전자공학과 부교수