QoS Bee Routing Protocol for Ad Hoc Networks

Jing Li
Department of Information Engineering, Taiyuan University of Technology, Taiyuan, China
Email: lj19831010@126.com

Yanping Li, Yuan Wang and Huakui Wang
Department of Information Engineering, Taiyuan University of Technology, Taiyuan, China
Email: liyanping@tyut.edu.cn

Abstract—**In this paper, we present QoS bee routing protocol (QBR), a new routing algorithm based on the characteristics of Ad hoc networks and the bee colony algorithm. The new routing protocol is inspired by the foraging behavior of bees in nature, which can find the solution of global optimal value very quickly with the characteristic that every bee is able to search local optimal solution. Biological bees are structured to several special bees' agents. The dynamic optimization methods are suitable for the basic characteristics of the self-organizing networks. In the end, in order to examine the performance of the new algorithm, we compare the QBR algorithm and the traditional classical routing protocol-MAODV in the NS2. The results show that the new algorithm is better than MAODV both in bandwidth requirement and end-to-end delay, which proves that the new routing protocol is more adapted to the sensitive, high density, and high quality of the service networks.**

Index Terms—**Ad hoc networks, bee algorithm, bandwidth requirement, QoS, end-to-end delay**

I. Introduction

With the rapidly increasing development of the mobile communication technology, the Ad hoc networks has become a hot discussion and research where people work and live gradually. The features of dynamic network topology, self-organizing, non-center, multi-hop, limited bandwidth transmission, and limited mobile terminal make the traditional routing algorithms are no longer appropriate for the Quality of Service (QoS) routing optimization solutions. To improve the original routing algorithm or to create a new technology to solve this problem is a popular research now.

In recent years, many routing protocols related to QoS for Ad hoc networks have been proposed, for example, QoS-MSR [1] which collects QoS information through route discovery of multipath source routing (MSR) and establishes QoS route with reserved bandwidth, QAR [2] which takes bandwidth as the metric of the admission scheme and considers the capability of the available node and path according to node congestion factor and path congestion factor, and so on [3]-[6]. In addition, inspired by natural phenomena and bionic technology, many new routing protocols such as ant colony algorithm and genetic algorithm have emerged [7]-[10].

In this paper, according to the QoS quality control of Wireless Ad hoc networks and artificial intelligence, we present QoS bee routing (QBR) protocol, which is an adaptive distributed routing protocol based on the behavior that bees can find food and return to the beehive to tell other bees the information about food. The QBR uses random broadcast form to find the routing neighbor nodes, and eventually find the destination node. We establish the system model of QBR protocol, describe the algorithm, give the routing table structure, and define the route discovery and route maintenance process. In the end of the paper, the proposed algorithm and the multi-Ad-hoc on demand distance vector routing (MAODV) algorithm [11], [12] are compared in NS2 simulation experiment, which show the superiority of the new algorithm in the indexes of average end-to-end delay and the average bandwidth.

II. QoS Bee Algorithm

A. Mechanism of Bee Colony Communication

Bees are highly organized social insects; they maintain their own lives by collecting nectar. Although the individual behavior of bees is very simple, the behavior of bee colony is very high complexity and well-organized. Food sharing information between bees transmits through the Scouters' swing dance. Scouters find the food and return to the beehive to tell other bees the information about distance, direction and the quality of the food through the swing dance. Other Followers get the information above according to the swing dance.

As we all known, bees deliver the information about food through the swing dance. First of all, they explore an area to find food as the identity of Scouter, and the beginning of the search is completely blind. After a period of discovery, if the Scouters find the certain location about the food source, they will remember the

Manuscript received July 22, 2013; revised August, 2013; accepted November, 2013.
Corresponding anther email: liyanping@tyut.edu.cn.

doi:10.4304/jsw.9.4.1019-1027

source area about food, and return to the beehive to transfer and share the location of food to other bees through the swing dance, and then some bees are recruited to be Followers. Followers gather honey based on the information provided by Scouters, and the number of Followers is proportional to the quality of food. This period is called the food exploration period. In this period, bees gather food and calculate the quality of the food to judge whether it needs to continue the action or give up.

QBR protocol is inspired by Bee colony communication [13]-[15]. Aiming at the characteristics of Wireless Ad hoc networks, it needs to design a special adaptive algorithm. Firstly, we will take the beehive as the base node or the source node, and take a bee as a data packet. The source node sends data packets to the node regarded as food source; other nodes can also send data to the source node. The intermediate nodes' role is to send relay packets on the link, which likes that bees relay the information back to the hive by swing dance. Each node has its independent routing table; the routing table contains the path information to the destination node and the only record about the details of the next hop address.

B. The Packet Format of QBR

Scouter and Follower are two important parts in this routing algorithm. They use the refresh mechanism to update available QoS information and neighbor list information of the source node or the intermediate nodes. During the route maintenance phase, the error Scouter package is used to alert related nodes to change their routing table because of the failures occurred in the link.

(1) Scouter: Scouters including forward Scouter and feedback Scouter are used in the route discovery process as control packets. At the beginning, the source node sends the forward Scouters to find the destination node. During this time, the prior knowledge of the search can be provided by the system, or can be completely random. When the destination node is found, the forward Scouters return to the source node and inform the source node the position and direction of the destination node. The Scouter on back way is called feedback Scouter.

Forward Scouter: In the route discovery process, use Forward Scouters to make sure the destination node can be found. They are generated from the source node and can find the destination node eventually through intermediate nodes. They temporarily record their information in every routing table they reached to ensure that they can return from the destination node to the source node accurately.

The format of forward Scouter packet is shown in Tab.I, each of forward Scouter is configured a unique incremental identification code by source node. Through the combination of identification codes of the source node (BeehiveId) and the Scouter (ScoutId), it can make the only route discovery request, which can effectively prevent route from infinitely sending in a small range up to form a loop.

Control packet of forward Scouter also contains the ID code of destination node (FoodId), the minimum bandwidth in routing request (Bandwidth), the maximum allowable delay (Delay) and the maximum lifespan parameter of forward Scouter package (Lifespan). The maximum lifespan parameter limits the maximum number of hops. If the forward Scouter does not reach the destination node until the maximum lifespan is exceeded, it will be discarded. If the source node does not receive any feedback information over the waiting time, it is necessary to increase the maximum number of hops. The HopCount field records the number of hops that Scouter has passed by from the source node to the current node. The Stamp field records the transmission time. All of these parameters can be used to calculate the available bandwidth and delay and then decide whether the network satisfies the QoS requests or not.

TABLE I.
FORMAT OF FORWARD SCOUTER PACKET

ScoutId	ID code of Scouter
BeehiveId	ID code of beehive
FoodId	ID code of destination node
Bandwidth	Manimum required bandwidth
Delay	allowable delay
Lifespan	Maximum number of hops
HopCount	Number of passed hops
Stamp	Stamp

Feedback Scouter: When the destination node is found, it will reverse the feedback Scouter package based on the path list of the forward Scouter. The same as the forward Scouter, the feedback Scouter packet also includes ScoutId, BeehiveId, FoodId, Bandwidth and Delay. Feedback Scouter package uses HopCount field to mark the number of hops from the source node to the destination node. After the destination node is found, and before the forward Scouter packet is discarded, Feedback Scouter will initialize the number of passed hops in the corresponding field. In addition, in order to inform the source node the real-time information about bandwidth and delay, Feedback Scouter initializes the Stamp field, which includes the consumed time of the forward Scouter. In order to ensure the feedback Scouter can accurately return to the source node, the package also keeps the maximum hop count parameters.

(2) Follower: Follower package is used for data transmission. When the destination node is found, Follower package with data would be sent to the destination node in turn.

C. Routing Table Structure

Each node has its independent routing table information; the routing table contains different paths to network terminals. Each path was arranged by weighting factors in associated with QoS constraint condition and then stored in priority order. Path with larger weighting factor is more suitable for transmitting data packets than path with smaller weighting factor. Each weighted routing table contains the following contents: FoodId, the identifier of the next hop node to the destination node, and the identifier of the last hop node back to the source node.

Data packet transmits through the next hop route. At the same time, the current nodes in the communication update their own last hop route timely until they return to the source node. The node or link failure will change the information about bandwidth and delay, so it is necessary to update these records information in real-time. These changes will be calculated by the real-time packets. Routing table also includes HopCount field to show the number of hops to reach the destination node, and the weighting factor associating with QoS. The weighting factor is calculated according to the path loss of the data packet. The lower the delay is, the higher the weighting factor is.

The ScoutId is also in the routing table, it is temporarily recorded in the routing table in the route discovery phase until the destination node is found. And then, it will be forever recorded in the routing table of feedback Scouter. If the source node does not receive the information about feedback Scouter when the maximum time limit exceeds, it will retransmit a new forward Scouter. The average end-to-end delay and the average bandwidth from the source node to the destination node will also be recorded. If QoS requirement is not satisfied or link fails, the route is considered invalid. Each intermediate node will be recorded in the neighbor lists.

D. Description of Routing Algorithm

(1) Neighbor routing connection discovery phase

At this stage, each node informs all neighbor nodes that connection is activated. It refreshes every link by estimating Qos requirement such as the appropriate bandwidth and delay, and broadcasts the updated information to the neighbor nodes. When the neighbor nodes receive updates, its routing table will identify that the node sending update news is effectively activated. In addition, the routing table will save the information of bandwidth and delay requirements.

If a node does not receive the feedback information from its neighbor nodes within a specified time, the information of this node will be discarded from the routing table of its neighbor lists. In this case, the system will send a wrong Scouter to notify other nodes that the link is broken.

(2) The route discovery phase

When the data is to be transmitted, the source node will firstly check whether the destination node is recorded in its routing table and whether the link can meet the QoS requirements, and then test whether the bandwidth of the data to be sent is less than the bandwidth requirement and whether the delay needed to transmit data is less than the delay requirement set in the routing table. If all of above requirements are met and there are enough Followers to send data, the source node starts to transmit data.

If there is no Follower, data transfer will be interrupted and wait to hire enough Followers. If no information of nodes returns from the destination node or the link cannot meet the needs of QoS, data transmission will also be interrupted and wait for a new route discovery. The source node generates a unique forward Scouter and copies some Scouters and then randomly broadcasts them to the neighbor nodes. Random broadcast means that the source node broadcasts to the neighbor nodes in accordance with a certain proportion. In this paper, assume the proportion is 80%.

In each route discovery process, all copied forward Scouters share the same ScoutId, BeehiveId, FoodId, and HopCount which is initialized as zero and increases one after each route until copied forward Scouters reach the destination node. Each time the Scouter visiting a node, the transmission time is accumulated in the Stamp field. The current node will compare the delay caused by Scouter transmission with the maximum allowable delay in Delay field, and also compare the bandwidth required for sending data with the available bandwidth of the Scouter in Bandwidth field. If the requirements of QoS can be met, then route discovery request still continues, otherwise route discovery request will be forbidden, and discard the forward Scouter package.

On the other hand, to prevent receiving the same Scouter package in the future, if the intermediate node receives a Scouter packet with the same ScoutId and BeehiveId as the Scouter packet received before, the later Scouter packet should be discarded. If there is no new packets with the same identification code arriving, the current node records ScoutId and BeehiveId in the routing table of the last hop. Then the intermediate node checks whether there is a path to the destination node in its own routing table, if it does not exist, the intermediate node will randomly transmit the Scouter packages like the source node. If there is already a path to the destination node in the intermediate node’s routing table, the intermediate node will send a feedback Scouter to the source node along the reverse route. On the way back home, feedback Scouter records the next hop routing address in its routing table.

When the feedback Scouter returns to the source node, the source node will calculate the weight of global link according to the records in the Stamp field, and hire a large number of Followers to send the data to the route discovery path conforming to the QoS requirements. Then the data is encapsulated with Followers and sent to the destination node. The rout selects path to transmit data according to the weighting factor, if the weighting factor of one route is higher than other routs, it will be preferred and sent more Followers than other routs, which can greatly reduce the network congestion, improve the bandwidth utilization and decrease the delay.

In the route discovery phase, Scouter visits the following three kinds of nodes: the source node or base station node (beehive), the intermediate node and the destination node. At first, the forward Scouter is sent from the source node and finally arrived at the destination node through each intermediate node, then feedback Scouter returns to the source node. At last, the source node transmits data with Followers.

(3) Route maintenance phase

In the Wireless Ad hoc networks, nodes discretionarily move, join in or leave the networks, wireless environment has various interferences, and network topology is dynamic, all of which indicates that wireless transmission condition is limited, so the QBR algorithm must maintain itself dynamically and adaptively to adapt the harsh transmission condition. In the route maintenance phase, nodes periodically send refresh requests to their neighbor lists to ensure the stability of the network connection, and the refreshed packets ensure the safety of the connection within the transmission range. If the node detects that the link is failure, other routes will consider the link is disconnection. Further, nodes can detect the situation that QoS requirements are not met, such as the bandwidth is too low or the delay is too large, then the nodes will send error Scouter to the source node to tell nodes in other reverse links that this link is failure and require them to remove the routing information about the failed link. If it still wants to transmit data, the source node should relaunch a new QoS routing request.

III. ALGORITHM DESIGN

A. System Model

In the simulation, we use a directed graph $G=(V,E)$ to model the wireless communication networks where V is a finite set of mobile nodes and E is a finite set of bi-direction links, the number of nodes is denoted by $|V|$;the number of links is denoted by $|E|$; $e(i,j)$ represents the links between node "i" and node "j"; $m(i,j)$ represents the QoS state parameter of the link (*i, j*). If $m(i,j)=m(j,i)$, the network is a symmetric network, if $m(i,j)\neq m(j,i)$, the network is an asymmetric network.

Path list to the specified destination saved in the router or other Internet's network device is called the routing table. In the self-organizing wireless networks, the path selection is closely related with constraint. The constraint is the probability to choose one of the neighbor nodes as the next hop address to the destination node through weighted coefficients. If node "a" has b neighbor nodes, then the routing table of node "a" can be expressed by mathematical modeling representation: $R_a=\left[r^a_{d,j}\right]_{|V|-1,b}$, where $|V|-1$ is the number of rows in the routing table, b is the number of columns in the routing table, $r^a_{d,j}$ is the probability of selecting "j" as the next hop route to the destination node "d". In the routing table, each row represents one destination node; each column represents one neighbor node. According to the theory of probability principle, weighting factor of each row satisfies the equation below:

$$\sum\nolimits_{j\in J(a)} r^a_{d,j}=1\,. \tag{1}$$

where $J(a)$ is a set of all neighbor nodes of node "a".

Fig.1 and Tab.II are examples of QBR routing table.

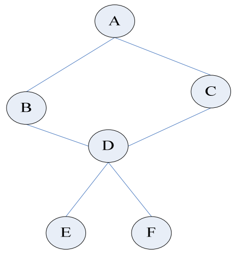

Figure 1. An example of QBR routing table

TABLE II.
AN EXAMPLE OF QBR ROUTING TABLE

Destination node \ Neighbor node	B	C
B	0.95	0.05
C	0.06	0.94
D	0.50	0.50
E	0.61	0.39
F	0.55	0.45

Assume "s" is the source node in the multicast network topology, the M $\left(M\in\{V-\{s\}\}\right)$ is the destination nodes set of multicast networks, $|M|$ is the number of destination nodes, $T(s,M)$ is the multicast tree from source node "s" to the destination nodes set *M*. $p(s,d)$ is the path from "s" to a destination node "d" in the *M*. Through modeling the route we can finally get the following relationships:

Overall bandwidth is the minimum bandwidth of each cascade link:

$$Bandwidth\left(p(s,d)\right)= \min\left\{bandwidth(e),e\in p(s,d)\right\}. \tag{2}$$

Delay is the sum of link delay and node delay:

$$Delay\left(p(s,d)\right)=\sum\nolimits_{e\in p(s,d)} delay(e) +\sum\nolimits_{v\in p(s,d)} delay(v)\,. \tag{3}$$

Loss is the sum of all link losses from the current node to the destination node:

$$Cost\left(p(s,d)\right)=\sum\nolimits_{e\in p(s,d)} cost(e)\,. \tag{4}$$

Packet loss rate is the product of the packet loss rate of each node:

$$Packet\_loss\left(p(s,d)\right)= 1-\prod\nolimits_{v\in p(s,d)}\left(1-packet\_loss(v)\right). \tag{5}$$

Delay jitter is the sum of all delay jitters of each node and each link:

$$Delay\_jitter\left(p(s,d)\right)=\sum_{e\in p(s,d)}delay\_jitter(e)+\sum_{v\in p(s,d)}delay\_jitter(v). \quad (6)$$

Multicast path loss is the sum of the losses of all links in the multicast:

$$Cost\left(T(s,M)\right)=\sum_{e\in T(s,M)}cost(e). \quad (7)$$

The total delay of multicast is the maximum delay value from the source node to each destination node:

$$Delay\left(T(s,M)\right)=max_{\forall d\in M}\left\{delay\left(p(s,d)\right)\right\}. \quad (8)$$

Multicast delay jitter is the absolute value of difference between any two paths to the destination node:

$$Delay\_jitter\left(T(s,M)\right)=\left|delay\left(p(s,d1)\right)-delay\left(p(s,d2)\right)\right|. \quad (9)$$

Multicast packet loss rate is the maximum packet loss rate from the source node to each destination node:

$$Packet\_loss\left(T(s,M)\right)=max_{\forall d\in M}\left\{packet\_loss\left(p(s,d)\right)\right\}. \quad (10)$$

As is known to all, a common feature of Ad hoc networks is that they are constantly changing, so its state information is always inaccurate and dynamic. The state information includes:

(1) State information about link connection, such as the amount of delay and reserved bandwidth in the link connection;

(2) Connection properties of the network topology;

(3) State information of the nodes, such as node loss and queue delay for data transmission and so on.

In all of the state information, the largest fluctuation is the amount of reserved bandwidth in the link connection, and the inaccurate performance brought by the reserved bandwidth is the most significant. Because the amount of reserved bandwidth in the link connection is very representative, in order to simplify the model, in this paper, we only consider the inaccuracy brought by the reserved bandwidth, which will not greatly affect the actual performance of route.

In this paper, we observe the inaccuracy of bandwidth in the link *(i, j)* according to the amount of change about the past and current reserved bandwidth, so that we can infer the amount of reserved bandwidth when Follower encapsulated data through this route. Using $\Delta b(i,j)$ to represent the maximum fluctuation range about $b(i,j)$, which means $\left(b(i,j)-\Delta b(i,j)\right)\le b(i,j)\le\left(b(i,j)+\Delta b(i,j)\right)$.

The maximum fluctuation about the amount of reserved bandwidth when a forward Scouter flies from node "i" to node "j" is $\Delta b_{old}(i,j)$, then the feedback Scouter will return to the source node along the corresponding reverse path, and this short path is *(j, i)*. The new maximum fluctuation about the amount of reserved bandwidth when the feedback Scouter arrive at node "i" is $\Delta b_{new}(i,j)$. The same as before, the amount of reserved bandwidth about the forward Scouter arriving at node "j" is $b_{old}(i,j)$, and the amount of reserved bandwidth about the feedback Scouter back to node "i" is $b_{new}(i,j)$, then we can get the following formula:

$$\Delta b_{new}(i,j)=\alpha\times\Delta b_{old}(i,j)+(1-\alpha)\times\left|b_{new}(i,j)-b_{old}(i,j)\right|. \quad (11)$$

where $\alpha(\alpha<1)$ is the control parameter, which is 0.3 in this paper, and it is used for controlling the disappearing speed about $\Delta b_{old}(i,j)$.

When the Follower encapsulated data package passes the link *(i, j)*, we assume its reserved bandwidth is uniform in the interval between $b_{new}(i,j)-\Delta b_{new}(i,j)$ and $b_{new}(i,j)+\Delta b_{new}(i,j)$. So the probability of the bandwidth requirement is:

$$P\left(b(i,j)\ge bandwidth\right)=\begin{cases}0 & bandwidth>b_{new}(i,j)+\Delta b_{new}(i,j)\\ \dfrac{b_{new}(i,j)+\Delta b_{new}(i,j)-bandwidth}{2\Delta b_{new}(i,j)} & otherwise\end{cases}. \quad (12)$$

B. Main Ideal

First of all, introduce the Scouter packet format, in order to meet the needs of multicast we add 4 fields: Nodes (the visited node set), Cost (the visited path loss set), Time (the Scouter' flying time record), Bw (the visited link bandwidth set).

The source node sends Scouters to the destination node periodically and randomly. Assume a forward Scouter is sent from the source node "s" to the destination node "d" along the route *(s, ..., i, j,..., d)*. As shown in Fig.2, when the forward Scouter is accessing node "i" now, it will record all visited nodes *(s, ..., i)* and store them in the Nodes field, record the visited path loss and store them in the Cost field. It will also calculate the flight time and then determine whether the Time is greater than the Delay. If the time of flight exceeds, we will discard the forward Scouter.

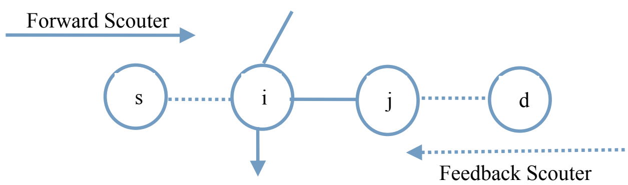

Figure 2. The workflow of Scouter

If the time of flight does not exceed, and node “j” has the maximum weighting factor in the routing table of node “i”, which means node “j” has the largest probability value as the next hop, forward Scouter will choose node “j” as the next hop to the destination node “d”, and estimate $b_{old}(i,j)$, which is the amount of reserved bandwidth about the link (*i, j*). If $b_{old}(i,j) \geq bandwidth$, which means node “j” satisfies the bandwidth requirement, then the forward Scouter will eventually chooses node “j” as the next hop. If the bandwidth demand is insufficient or allowable delay is overtime, it will choose another path that has the second greatest weighting factor and repeat the above assessments.

When the forward Scouter arrives at the destination node “d”, destination node will generate a feedback Scouter. The forward Scouter will copy all records in its routing table to the feedback Scouter and then be discarded. The feedback Scouter reverses from the destination node “d” back to the source node “s” along the path the forward Scouter has passed by. When the feedback Scouter arrives at node “j” from node “i”, it detects reserved bandwidth of the short path (*i, j*), which is expressed by $b_{new}(i,j)$. Then it can calculate the values of $\Delta b_{new}(i,j)$ and $P(b(i,j) \geq bandwidth)$ according to (11) and (12).

Feedback Scouter will cache the value of $b_{new}(i,j)$ in the routing table of the next node, and use the value to help the next forward Scouter through this node calculating $\Delta b_{old}(i,j)$. It will also update the related data according to (13) and (14):

$$r^{i}_{d,j} = \frac{\partial r + r^{i}_{d,k}}{1+\partial r}. \quad (13)$$

$$r^{i}_{d,k} = \frac{r^{i}_{d,j}}{1+\partial r}. \quad (14)$$

where “*k*” is the node different from node “i”.

∂r is the enhanced coefficient of probability, its calculation formula is:

$$\partial r = \frac{m}{c(s,j)} + n \times P(b(i,j) \geq bandwidth). \quad (15)$$

where *m* and *n* are system parameters, $c(s,j)$ is the total cost for path (*s, ..., i, j*), which is the path from node “s” to node “j”. The value of $c(s,j)$ is recorded in the Cost field.

C. The Algorithm Process

For the multicast networks, when a node in the networks launches a route discovery request to establish a multicast networks, it becomes the source node. The reserved bandwidth is *B*; the maximum network delay is *D*. Each node in the network topology initializes its own routing table according to the constraint conditions of QoS and the distribution of the amount of reserved bandwidth in current networks.

The process is shown as follows:

(1) Firstly, the source node select one destination node “d1” from *M* randomly, and then generates one forward Scouter, which initializes its packet format as Tab.III.

TABLE III.
INITIALIZATION OF THE FIRST SCOUTER PACKET

ScoutId	1
BeehiveId	s
FoodId	D1
Bandwidth	B
Delay	D
Lifespan	20
HopCount	0
Stamp	null
Nodes	S
Cost	0
Time	0
Bw	null

Then according to (13), (14) and (15), the forward Scouter will choose the node with the maximum weighting factor and satisfying the dual constraints of Bandwidth and Delay in the routing table as the next hop node. If there are several routes having the same weighting factor, it will select one randomly.

(2) The forward Scouter is flying at every node. At each intermediate node, it will follow the strategy in Step 1 to choice the route with the maximum weighting factor as the next hop and record the amount of reserved bandwidth until it finds the destination node“d1”. Once the forward Scouter arrivals at the destination node“d1”, it will copy the records to the feedback Scouter and die. In the process of returning to the source node, feedback Scouter will update the corresponding entries of each intermediate node in the routing table according to the (13), (14) and (15).

(3) The source node copies *L* Scouters with the same ID, and sends the second forward Scouter with the same ID after the first forward Scouter being sent in a very short time. Step 1 and step 2 repeat until the QBR algorithm convergence, which means the minimum loss path $p(s,d1)$ satisfying the QoS bandwidth and delay requirements has been found. Assume the set of nodes on the path of $(s,\cdots,d1)$ is expressed by *M1*. If there is no feedback information after the *L* forward Scouters have been sent out, or the algorithm is still no sign of convergence, the route discovery request is failed. One of the reasons is that the Lifespan field is too small, it is better to increase the maximum number of hops appropriately.

(4) When the destination node“d1” is found, the source node will select another destination node “d2” in destination node set *M* randomly, and generate the number of L new forward Scouters. The initialization of the second forward Scouter packet is shown in Tab.IV.

TABLE IV.
INITIALIZATION OF THE SECOND SCOUTER PACKET

ScoutId	2
BeehiveId	s
FoodId	D2
Bandwidth	B
Delay	D
Lifespan	20
HopCount	0
Stamp	null
Nodes	S, M1
Cost	0
Time	0
Bw	null

The Nodes field here is *S, M1*. If the next hop route *u* which satisfies the QoS bandwidth requirement and has the maximum weighting factor belongs to the Nodes field, so this path is same as the first destination node “d1”, then the forward Scouter moves to the node “u”. Forward Scouter flies to the next hop following step 1 and step 2, until the next hop node “v” does not belong to the Nodes field, then the forward Scouter will update its packet format on node “b” which is the last hop node of node “v”, the updated packet is shown as Tab.V.

TABLE V.
THE UPDATED SCOUTER PACKET

ScoutId	2
BeehiveId	b
FoodId	D2
Bandwidth	B
Delay	D
Lifespan	20
HopCount	0
Stamp	null
Nodes	B, M1
Cost	0
Time	0
Bw	null

It should be noted that the forward Scouter on node “b” already knows its next hop node “v” does not satisfy the Nodes field, and then, it repeats step 1 to step 3 to find optimal path which meets QoS bandwidth and delay requirements and has the maximum weighting factor from the source node “b” to the destination node “d2”.

(5)If $M-\{d1,d2,d3,\cdots,dn\}$ is not equal empty set, it indicates that the multicast path has been found. Otherwise, repeat step 4 to search a new route.

(6) If the multicast route is determined, the source node “s” will send a *PATH* message to each destination node along each path. As long as each destination node receives a *PATH* message, it will reverse back a *RECV* message to the source node “s” along the way *PATH* message has passed. When the *RECV* message returns to the destination node, it reserves resource for the multicast communication. If the *RECV* message of a node can return to the source node, indicating that resource reserves successfully, the source node “s” is going to encapsulate transmitting data to Followers and send Followers to the corresponding destination nodes. If the *RECV* message can not return to the source node successfully, indicating that resource fails to reserve, the intermediate node is going to send a *CANCEL* message to the source node “s” and the corresponding destination nodes to notify them that resources reservation has failed and to release the reserved resources.

IV. SIMULATION AND ANALYSIS

A. Simulation Environment

The simulation experiments in the NS2 environment. Assume *m* is 0.75, *n* is 0.02, the capacity of the networks varies from 10 to 100 nodes, the range of the network topology is 1000*1000 square meters, and the simulation time is 500 seconds. The nodes in the networks broadcast in the network topology stochastically according to the distribution of random point mode; half of the nodes are randomly selected as the source node. Node movement speed will be set respectively for vehicular speed ranging from 1 meter per second to 5 meters per second and walking speed ranging from 1 meter per second to 20 meters per second.

B. Simulation Experiments

In the simulation experiments, two aspects need to be measured: the average end-to-end delay and the average bandwidth, which is defined as follows:

$$D_{avg}=\frac{D_{pkt}}{N_{pkt}}. \tag{16}$$

$$B_{avg}=\frac{B_{nod}}{N_{nod}}. \tag{17}$$

In the formulates above, D_{avg} represents average end-to-end delay, B_{avg} represents average bandwidth, D_{pkt} represents the total delay about the data packets arrived at all nodes at a certain time, and N_{pkt} represents the total number of the data packets arrived at all nodes at a certain time. B_{nod} represents the total bandwidth every node in the networks occupied, and N_{nod} represents the total number of all nodes in the networks. In order to make the results more objective and more convincing, each value in experimental results is the average value after 100 times experiments.

In order to analysis the performance of QBR protocol, the average end-to-end delay and the average bandwidth should be considered. The average end-to-end delay includes the buffering delay in route discovery, the waiting time of interface queue, the data retransmission delay in the link layer, the broadcasting delay and transmission delay and so on. The average bandwidth refers to the average bandwidth of each node occupied. The purpose of the two important measures in QoS is to transmit more data in the shortest time.

The end-to-end delay simulation results are shown in Fig.3 and Fig.4:

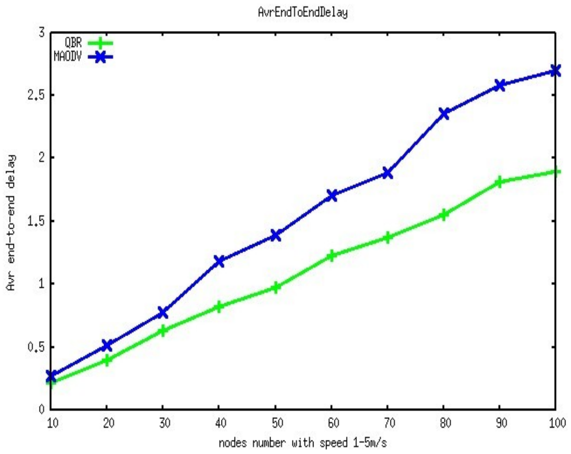

Figure 3. Average end-to-end delay with nodes speed 1-5m/s

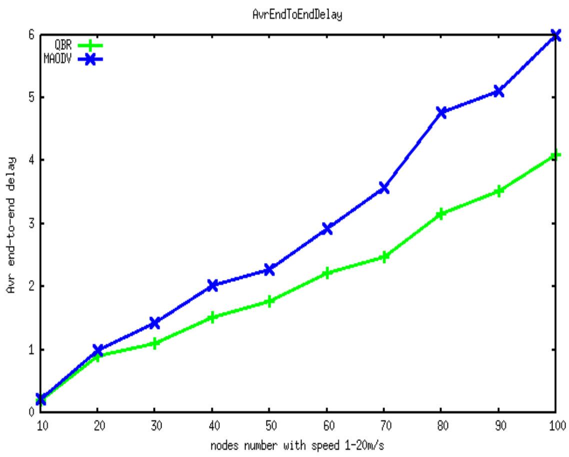

Figure 4. Average end-to-end delay with nodes speed 1-20m/s

As can be seen from the results, no matter vehicle speed or walking speed, the QBR protocol has better performance than MAODV protocol in average end-to-end delay. When the number of nodes varies between 10 and100, QBR always has a lower value in the average end-to-end delay. With the increasing of the number of nodes, the average end-to-end delay of QBR gradually becomes stable, but MAODV is still increasing. The reason is the forward Scouter packages refresh network topology information at regular time, and the feedback information timely transmits varied path information, so that the new node with higher weighting factor will be chosen as the new next hop.

Fig.5 and Fig.6 present the simulation results of the average bandwidth. As can be seen from the results, the QBR protocol still has better performance than MAODV in the average bandwidth. Although both of the protocols use the method of multipath route discovery, QBR uses the resource reservation method, which makes the average bandwidth been optimized. Noting the bandwidth decreases significantly between 50 and 75 nodes, due to the condition that the route discovery will occupy a certain bandwidth with the increasing of nodes. Therefore, properly reducing the number of nodes and Scouters in the Wireless Ad hoc networks is very necessary.

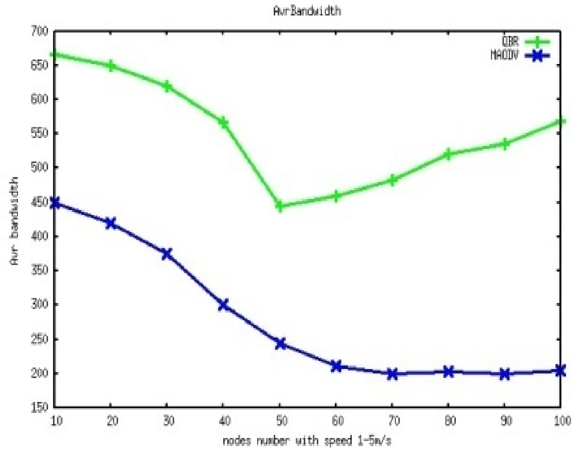

Figure 5. Average bandwidth with nodes speed 1-5m/s

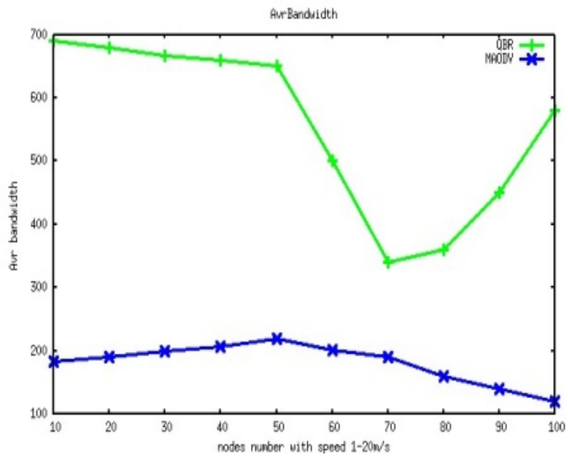

Figure 6. Average bandwidth with nodes speed 1-20m/s

V. Conclusions

In this paper, we propose QBR protocol based on the characteristics of Ad hoc networks and bee colony algorithm, and introduce the function of reserved bandwidth and delay to the route discovery algorithm. In this algorithm, the forward Scouter relies on calculating the weighting factor and path loss to find the optimal path to satisfy QoS requirements, which effectively reduces the control parameters and cuts down the network overhead. Additionally, because the algorithm introduces feedback Scouter mechanism, the protocol can still be used in asymmetric link. Finally, simulation experiments through NS2 show that QBR protocol has obvious advantages in the average end-to-end delay and the average bandwidth compared with traditional MAODV protocol, which indicates the new protocol is more suitable for the high density network environment.

REFERENCES

[1] Taishu Yan, Guanghong Wang, Lei Wang, Oliver W.W. Yang, Yongjie Fan, "Provisioning QoS guarantee by multipath routing and reservation in Ad hoc networks," *Journal of Computer Science and Technology*, vol.19, no.2, 2004, pp.128-137.

[2] Rong Geng, Zhe Li, "QoS-aware routing based on local information for mobile Ad hoc networks," *Distributed Computer and Internet Technology Lecture Notes in Computer Science*, vol.4317, 2006, pp.159-167.

[3] Qingwen Wang, Haoshan Shi, "QoS-aware multipath on-demand routing protocol based on link quality for mobile Ad hoc Networks," *Journal of Computer Engineering and Applications*, vol.46, no.29, 2010, pp.29-32.

[4] M. Natkaniec, K. K. Szott, S. Szott, and G. Bianchi, "A survey of medium access mechanisms for providing QoS in Ad hoc networks," *IEEE Communications Surveys & Tutorials*, 2012, pp.1–29.

[5] Zhiming Xu, Yu Wang, Jingguo Zhu, "A Reliable Multicast Routing Protocol for High-speed Mobile Ad Hoc Networks: R-ODMRP," *Journal of software*, vol. 5, no.1 (2010), 2010, pp.20-27.

[6] Li Zhang, Ban Qu, "Optimization for QoS-based multicast dynamic routing of self-organizing overlay network," *Journal of Computer Engineering and Applications*, vol. 49, no.3, 2013, pp.51-53.

[7] Shiau-Huey Wang, "An Exchange Framework for Intrusion Alarm Reduction in Mobile Ad-hoc Networks," *Journal of Computer*, vol. 8, no.7, 2013, pp.1648-1655.

[8] G.D.Caro, F.Ducatelle, L.Mgambardella, "an adaptive nature inspired algorithm for routing in mobile Ad hoc networks," *European: Transactions on Telecommunication (Special Issue on self-organization in mobile networking)*, vol. 6, 2005.

[9] Fangbin Liu, Fengyu Liu, Hong Zhang, "An Energy-efficient and Real-time Anonymous Routing Protocol for Ad hoc Networks," *Journal of Computers*, vol. 7, no.4 (2012), 2012, pp.900-912.

[10] Julio Barbancho, Carlos Leon, F.J. Molina, and Antonio Barbancho, "Using artificial intelligence in routing schemes for wireless networks," *Journal of Computer Communications*, vol. 30, no.14-15, 2007, pp. 2802-2811.

[11] C. E. Perkins, E. M. Royer, "Ad hoc on-demand distance vector routing," *IEEE Workshop on Mobile Computer Systems and Application*, New Orleans, LA, vol.1, 1999, pp. 90-100.

[12] C. E. Perkins, E. M. Royer, and S. R. Das, "Quality of service for Ad hoc on-demand distance vector routing," IETF Internet Draft, 2000.

[13] Dervis Karaboga, Bahriye Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC)," *Journal of Global Optimization*, vol.39, no.3, 2007, pp.459-471.

[14] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," CA: Erciyes University,Technical Repoter:TR06, 2005.

[15] Bilal Alatas, "Chaotic bee colony algorithms for global numerical optimization," *Expert Systems with Applications*, vol.37, no.8, 2010, pp.5682-5687.

Jing Li was born in Huanghua City, Hebei Province, China in 1983. She received the B.S. degree in Department of Electronic and Information Engineering in 2006, and received M.S. degree in Department of Communication Engineering in 2009, both in Taiyuan University of Technology, Taiyuan City, Shanxi Province, China. She is currently working toward the Ph.D. degree in the Institute of Information Engineering, Taiyuan University of Technology, Shanxi Province, China.

Her research interests include signal processing, Ad-hoc networks and Wireless Sensor Networks.

Yanping Li was born in Taiyuan City, Shanxi Province, China in 1963 and received the B.S., M.S., and Ph.D. degrees in the Institute of Information Engineering, Taiyuan University of Technology, Taiyuan City, Shanxi Province, China.

She is currently a Professor in the Institute of Information Engineering, Taiyuan University of Technology, Shanxi, China. She has directed more than 50 graduate students and widely published papers in signal processing of communications and Wireless networks. Her research interests include signal processing wireless communications, and bandwidth communications.

Huakui Wang was born in Changzhi City, Shanxi Province, China in 1946 and received the B.S. and M.S. degrees in the Institute of Information Engineering, Taiyuan Institute of Technology, Taiyuan City, Shanxi Province, China.

He is a Professor and doctoral tutor of the Institute of Information Engineering, Taiyuan University of Technology, Shanxi, China. He has directed more than 80 graduate students and widely published paper in signal processing and Wireless Sensor Networks. He published 6 monographs, and delivered more than 120 papers in famous journals or conferences both here and abroad.