A Quality of Service Routing Scheme for Packet Switched Networks based on Ant Colony Behavior

Liliana Carrillo, J.L. Marzo, Pere Vilà, Lluís Fàbrega, Carles Guadall
Institut d'Informàtica i Aplicacions (IIIA), Universitat de Girona (UdG)
Campus Montilivi, Av. Lluís Santalò s/n, Edifici P-IV, 17071 Girona, Spain
{lilianac, marzo, perev, fabrega, guadall}@eia.udg.es
http://bcds.udg.es

Abstract. Quality of Service (QoS) guarantees must be supported in a network that intends to carry real-time multimedia traffic effectively. A key problem in providing QoS guarantees is routing which consists of finding a path in a network that satisfies several constraints such as bandwidth, delay, delay jitter, etc. This paper proposes a novel QoS routing scheme called AntNet-QoS which provides QoS by adaptively learning routing tables in packet switched networks. AntNet-QoS is based on the so-called AntNet routing, a swarm intelligence router developed by G. Di Caro and M. Dorigo, that only considers delay when calculating the probabilities used in its routing tables. The innovative AntNet-QoS proposal is based on the utilization of different types of ants (routing packets) for the different classes of services (e.g., premium, best effort and so on) with different QoS requirements. AntNet-QoS considers parameters such as guaranteed bandwidth and maximum delay, enabling the distribution of different packet-based services using a model derived from standard AntNet.

Keywords. Quality of Service, Routing, Ant Colony Optimization, Ant-based algorithms, DiffServ.

1. INTRODUCTION
To carry real-time multimedia traffic effectively, a network must provide quality of service (QoS) guarantees. QoS guarantees are based on such parameters as loss rate, throughput, security, and delay. Furthermore, the support of QoS services is underpinned by QoS routing (QoSR).

QoS-based routing is defined as “a routing mechanism under which paths for flows are determined based on some knowledge of resource availability in the network as well as the QoS requirement of flows” [1]. Thus, the task of QoSR is to find a path in the network subject to constraints on metrics such as bandwidth, delay, delay jitter and cost.

Some recent QoS routing algorithms consider only a single metric, such as hop-count, delay, cost, etc. to characterize routes in a network. One such scheme is AntNet Routing which determines routes in the network inspired by the trail-following behavior of real ant colonies. AntNet has been adopted as a basis for this proposal because it offers good performance compared with other routing algorithms [2, 3] and has positive scalability [4] and essential part of any specific QoS proposal [1].

In this paper, the proposed work focuses upon the provision of an innovative scheme to provide QoS routing, where the use of multiple metrics are used to better characterize a network and to support a wide range of QoS requirements.

The basic theory associated with Ant-based algorithms was summarized in the original thesis proposal [5] and is considered in detail in [2, 3, 6, 7]. We present some details in the subsequent section and comment on some recent work related to the application of Ant Colony Optimization (ACO) to QoS. Next, the AntNet-QoS algorithm is presented. Finally, we present conclusions, future work.

2. ROUTING BASED ON ANT COLONY BEHAVIOR
Well known Swarm Intelligence (SI) has been the basis of different approaches to solving distributed system optimization problems. In the area of network routing, algorithms have taken inspiration from the notion of stigmergy that describes the indirect communication taking place among individuals through modifications induced in their environment. Most of the algorithms, in this area, have taken their inspiration from natural ant colonies because ants are able to find shortest paths using as only the pheromone trail deposited by other ants.

In a packet network context, such pheromones are simply probabilities represented at each node by a routing table and are derived from the number of visitations associated with route-finding packets or ants.

Our proposal is based on the so-called AntNet routing algorithm, an ant-based algorithm developed by G. Di Caro and M. Dorigo, where mobile agents [2, 3] search for short routes in packet switching networks.

In AntNet, the information processed by ants is represented in every node by a routing table and an
In the routing table, every value of node along the return path. and updates the LTS structure and the routing table at each source node using the reverse path taken by the forward ant. Another ant is activated, a backward ant, which returns to its destination node (see Figure 1). The forward ant that finds the path from source node to destination node , works as follows:

The forward ant, , moves along the path 1→2→3→4→5 saving onto a memory stack the identifier of each visited node and the time elapsed between its departure from the start node and its arrival at this node, that following on it is inserted in a dictionary structure , carried by the forward ant. Once the forward ant arrives at node 5, it launches the backward ant that will travel in the opposite direction. In each node , the backward ant will use the stack contents to update the values for 

The routing table is updated by incrementing the goodness of , of the last node , the ant , came from, for the case where node is a destination, while the other nodes on the sub-paths are taken into consideration as destination nodes only if the trip time associated with the corresponding sub-path of the forward ant is statistically good.

![Forward Ant (1 → 5)](image)

**Figure 1.** Representation of the information in every node by means of the traditional AntNet routing.

Figure 1 shows the node structures used by mobile agents in AntNet for the case of a node with L neighbors and a network with N nodes. The routing table is organized as in vector-distance algorithms, but the entries are probabilistic values. The Local Traffic Statistics (LTS) function as an adaptive model for the traffic toward each possible destination.

In AntNet, two types of ants (routing packets) are used, forward ants and backward ants. Forward ants are explorative ants that have the goal of discovering new routes and constantly evaluate the state of existing routes, analyzing the delay. When a forward ant reach its goal, another ant is activated, a backward ant, which returns to its source node using the reverse path taken by the forward ant and updates the LTS structure and the routing table at each node along the return path.

In the routing table, every value of represents the probability or appropriateness of choosing as next node when the destination node is . These values are calculated according to the information (delay) gathered by forward ants and modified by taking into account historical aspects or statistics of local traffic.

The concept is more clearly illustrated with an example (see Figure 2). The forward ant that finds the path from source node to destination node , works as follows:

The forward ant, , moves along the path 1→2→3→4→5 saving onto a memory stack the identifier of each visited node and the time elapsed between its departure from the start node and its arrival at this node, that following on it is inserted in a dictionary structure , carried by the forward ant. Once the forward ant arrives at node 5, it launches the backward ant that will travel in the opposite direction. In each node , the backward ant will use the stack contents to update the values for 

The above methodology has been the basis for different kinds of routing strategies. There have been some successful applications of ant behavior to network control, the most prominent being AntNet [2, 3], and Ant-based Control (ABC) [8]. However, there also exist some attempts to apply ant behavior to QoS. In [9], Zhang Subing and Liu Zemin present a formulation of the application of ant algorithm to solve the QoS routing problem with multiple metrics, such as bandwidth, loss rate, delay and delay jitter. In [10], Chao-Hsien Chu and others present a heuristic ant algorithm for solving QoS multicast routing problem. Their algorithm considers multiple QoS metrics, such as bandwidth, delay, delay jitter, and packet loss rate, to find the multicast tree that minimizes the total cost.

In [11], Kazumasa Oida and Masatoshi Sekido describe an agent-based routing system for a datagram network, called ARS, which supports QoS routing, resource reservation and admission control functions by using ant-like agents. Their algorithm supports two QoS constraints: bandwidth and hop-count.

These algorithms have innumerable advantages, however lack a number of important features such as scalability, routing oscillations, routing loops and full utilization of the network capacity. Then, here we attempt to profit from their advantages and avoid their weaknesses and so propose in order to suggest a potentially move powerful QoS algorithm called AntNet-QoS.

### 3. A NOVEL ANNET-QOS APPROACH: ANT COLONY BEHAVIOR TO PROVIDE QUALITY OF SERVICE

AntNet-QoS is a quality of services routing algorithm based on ant colony behavior that enables the distribution of services.
different packet-based services using a model derived from traditional AntNet routing. Such services are likely to follow the differentiated service (DiffServ) model [12], and the scheme can, in summary, operate as follows:

1. Each flow arrives at a node with certain parameters of requested quality of service, defined by its differentiated services codepoint (DSCP) field [12]. These parameters take into account both delay values bandwidth, which must be guaranteed.

2. To find an achievable and low-cost route and to determine the network load, L-class ants which represent L-traffic classes or forwarding classes according to type of services (ToS) [12] (ToS such as premium, best effort, etc.), and it is assumed that the properties of pheromones deposited by the L-class ants are independent from each other.

3. Two types of ants are used in AntNet-QoS, in a manner similar to the traditional AntNet (forward and backward ants).
   - Forward ants are exploration ants that have the goal of discovering new routes and constantly evaluate the state of existing routes, analyzing the delay and the available or residual bandwidth.
   - At each node along the return path, backward ants update two structures in every node of the QoS data model and the routing table.

4. The probabilities in the routing tables represent the probability of that forwarding class of data achieving the required QoS.

In this way, AntNet-QoS modifies the AntNet node model (shown in figure 1), as illustrated in Figure 3.

![Figure 3](image_url)

**Figure 3.** Node model associated with AntNet-QoS

The innovative AntNet-QoS scheme has the following key aspects:

- To find an achievable and low-cost route and to investigate the network load, we use L-class ants to represent L-traffic classes (premium, best effort, etc.), and assume that the properties of pheromones deposited by the L-class ants are independent from each other. An ant selects a route solely based on the pheromone deposited by the same type of ant on the path.

- For each L-class ant, an independent forward ant, $F_{s,d}$ that experiments the network load as L-traffic class, is activated at regular time intervals $\Delta t$ from each node $s$ towards a destination node $d$. Each forward ant shares the same queues with data packets appropriate to QoS requirements in order to experience queue delay appropriate to its L-class ant. The destination nodes are locally selected according to the probability $P_d$ which is related to the traffic patterns generated by local load related to each L-class ant. $P_d$ represents the probability of activating an ant from the node $s$ to the node $d$, and therefore it adapts the exploratory activity of forward ants according to the current traffic distribution for each traffic class.

- The bandwidth available and the time passed since the ant was activated until its arrival to the $k$-th node is introduced into a memory stack $S_{s,d}(k)$. The forward ant $F_{s,d}$ transports a structure $D_{s,d}$ that contains information about all visited nodes in its memory stack.

- Every forward ant $F_{s,d}$ that is in a transit node $k$, selects the next hop node $n$ among all neighboring nodes, omitting those visited previously by the same ant. If all neighboring nodes have been visited, one is selected at random. The neighbor node $n$ is selected according to probability $P_n$, which is related to the probabilities in the routing table of each traffic class (one table for best effort traffic, another for premium traffic and so on) and the current link state.

- As in traditional AntNet, if a cycle is detected, the nodes in this cycle are removed from the memory stack. If the cycle is longer than the lifetime of the ant, the ant is destroyed to disable ants that are carrying old and invalid values about the network state.

- When a forward ant reaches the destination node $d$, another ant lives because of its death (the forward ant is destroyed). The new ant, named backward ant $B_{d,s}$, receives from the forward ant all its memory (information about times and bandwidth), and carries now the information contained in the structure $D_{s,d}$ previously converged by the forward ant $F_{s,d}$.

- The backward ant $B_{d,s}$, one independent for each L-traffic class, returns to node $s$ using the reserve path taken by the forward ant $F_{s,d}$. At every node $k$, along the path, the backward ant uses the information about the last position of its memory stack, to determine the next hop node. Backward ants do not need to share the same queues as the data packets because their task is to propagate the information collected in the memory stack to the routing tables as quickly as possible. Such ants are given the highest possible priority on their return journey.

- When each backward ant arrives at a node $k$ from a neighboring node $f$, the backward ant $B_{f,s}$ updates two main data structures of the node (see figure 3):
1. The Local QoS Statistics that contains two structures with information about the cost of the route from node $k$ to each node $i$ of the network.

   a. The first structure contains the delay averages and their respective degree of trust (e.g. trust intervals). Using these values, the probability $P_T$ of obtaining the requested delay conditions for the different service classes is calculated (each backward ant can reflect its network status experience only about its own class).

   b. The second structure contains the minimum values of remaining detected bandwidth for each traffic-class (hop by hop), its respective degree of trust and a threshold value for the degree of trust which is used to decide if a better remaining minimum exists. With these values, the probability $P_{bw}$ of obtaining the requested bandwidth conditions for the different service classes is calculated.

2. A routing table similar to the used in the vector–distance algorithm is constructed. In this table, a probability value $P_{QoS}$ is associated with the selection of node $j$ as the next hop, for the requested QoS, when the destination node is $i$. The probability weights are subject to the following constraint:

   $$\sum_{j \in N_i} P_{QoS_{ij}} = 1$$

   where $i \in [1,N]$ and $N_i = \{ \text{neighbor node } k \}$

   Probability $P_{QoS_{ij}}$ is a function that depends on the probabilities $P_{T_{ij}}$ and $P_{bw_{ij}}$ ($P_{QoS_{ij}} = F(P_{T_{ij}}, P_{bw_{ij}})$). The probability $P_{QoS_{ij}}$ is incremented by a value proportional to received reinforcement $r$ (positive or negative according to accomplishment of certain conditions of reinforcement and inspired by the original reinforcement used in AntNet) and to the previous value of the probability in its corresponding routing table.

   The routing table is changed by either increasing the probability $P_{QoS_{ij}}$ associated with the node $f$ when destination is node $d$ or decreasing probability $P_{QoS_{ij}}$ associated with the other nodes $j$ in the neighborhood for the same destination $d$.

4. CONCLUSIONS AND FUTURE WORK

QoS-based routing has been recognized as a key element in the evolution of QoS-based service offerings in the Internet. This contains a proposal for research examining QoS-based routing in the Internet based on Swarm Intelligence. This framework is based on extending the traditional AntNet routing model for supporting QoS.

Most recent QoS routing algorithms consider only a single metric, such as hop-count, delay, cost, etc. to characterize routes in a network, but the use of multiple metrics is needed to better characterize a network and to support a wide range of QoS requirements. This is the prime motivation in this work.

The idea of Ant Algorithm to solve the QoS routing problem with multiple metrics, such as bandwidth, delay, etc. is presented. The aim of AntNet.QoS is to support a distribution of types of services described by DiffServ using the QoS parameters carried by the AntNet.QoS ants.

In defining the new AntNet-QoS routing mechanism, the proposals from B. Barán and S. Sosa to improve the behavior of traditional AntNet [13] will be considered along with alternatives proposals for AntNet working with QoS [9-11].

As a simulation model, we plan to develop a realistic simulation in order to evaluate our QoS scheme performance. We are interested in measures of performance such as scalability, delay, overhead, resource utilization and so on.

The knowledge about the characteristics of the incoming streams is fundamental in networks able to provide QoS. AntNet-QoS routing could act in different ways depending on the traffic class.

We plan to consider an AntNet-QoS scenario with four classes of ants related to four traffic classes. An Expedited Forwarding (EF) class is defined to transport real-time traffic, two Assured Forwarding (AF1 and AF2) classes are used by traffic with two different flavors for losses and, as usual, a Best Effort (BE) class for traffic with no QoS requirements (see table 1 for more details).

<table>
<thead>
<tr>
<th>Traffic Class (DS field)</th>
<th>Services</th>
<th>Ants</th>
</tr>
</thead>
<tbody>
<tr>
<td>EF class – requires low loss, low latency, low jitter and assured bandwidth</td>
<td>Premium</td>
<td>EF class</td>
</tr>
<tr>
<td>AF1 class – high level of forwarding assurance, with 3 sub-classes of drop precedence: AF11, AF12, AF13</td>
<td>Assured Gold</td>
<td>AF1 class</td>
</tr>
<tr>
<td>AF2 class – lower level of forwarding assurance, with 3 sub-classes of drop precedence: AF21, AF22, AF23</td>
<td>Assured Silver</td>
<td>AF2 class</td>
</tr>
<tr>
<td>BE class – the simplest traffic is best effort, with no QoS requirements</td>
<td>Best Effort</td>
<td>BE class</td>
</tr>
</tbody>
</table>

Furthermore, we plan to make simulations considering two network topologies, the NSFnet (14 nodes, 21 bi-directional links) and NTTnet (57 nodes, 162 bi-directional links) to test the performance of AntNet-QoS. In order to compare the performance of AntNet-QoS with the Ant algorithm described in [9], we plan to make simulations using the same environment and network topology.

We will also consider overhead and scalability problems related to AntNet-QoS scheme.

Currently, we are improving the system and implementing the AntNet-QoS algorithm.
5. ACKNOWLEDGEMENTS

The authors would like to thank the Spanish Education Ministry for their partial financial support by means of the grant MCyT [TIC2003-05567].

Liliana Carrillo would like to thank the ‘Universitat de Girona’ for the financial support (UdG grant) awarded to assist in the completion of her Ph.D. dissertation.

References


