

Quality of Service Aware Ant Colony Optimization Routing Algorithm

Chris Saliba, Reuben A. Farrugia

Department of Communications and Computer Engineering, University of Malta
Malta

csal0002@um.edu.mt

reuben.farrugia@um.edu.mt

Abstract— The demand for Internet connectivity has grown exponentially in the past few years. Moreover, the advent of new services such as Voice over IP (VoIP), Video on Demand (VOD) and Videoconferencing applications have caused a sever increase in traffic, which makes it extremely hard to ensure an acceptable level of quality. This is mainly attributed to current routing strategies, such as Link State (LS) and Distance Vector (DV), which are not optimal in terms of Quality of Service (QoS).

This paper presents a QoS-Aware routing strategy based on the Ant Colony Optimization (ACO) concept, where a set of artificial ants are used to determine the optimal path. The proposed method was compared to other state of the art ACO procedures and traditional routing schemes like LS and DV. Simulation results clearly demonstrate that the proposed scheme outperforms all the methods considered in this paper where throughput gains of 8% and a reduction in delay of 38% for time critical applications were achieved.

I. INTRODUCTION

Traditional routing strategies such as Link State (LS) and Distance Vector (DV) are widely used in current Internet. These methods try to find the optimal path from source to destination based on physical characteristics such as distance and delay [2], [3]. However, these mechanisms do not manage to optimize the paths in terms of throughput. Furthermore, the shortest path may become congested over time, while other alternative paths may provide better Quality of Service (QoS) [4].

Due to the limitations portrayed above, traditional routing methods are not flexible enough to sustain bandwidth hungry applications such as Voice over IP (VoIP), Video on Demand (VOD) and Videoconferencing [5]. Furthermore, these applications demand a different level of QoS when compared to traditional Internet applications. This has driven the development of new routing strategies which provide greater flexibility in order to satisfy multiple QoS requirements.

Several intelligent routing strategies can be found in literature. The centralized Genetic approach proposed in [10] adopts the knowledge of the whole network topology to derive the optimal routing solution. However, this approach will collapse if the centralized node fails. On the other hand, the Genetic Ant Routing Algorithm (GARA) presented in [11] is a decentralized routing mechanism which derives the required information from the neighbouring nodes. Nonetheless, the operation of this approach is highly complicated and requires a considerable amount of time to derive the optimal route.

Ant Colony Optimization (ACO) [8] is considered to provide a better compromise between optimality and complexity. In ACO, artificial ants are launched within a virtual environment, and as these agents move they modify their environment until a good solution to the problem is determined. The Ant Based Control (ABC) [6] is the first implementation of ACO which was adopted for connection-oriented networks. This algorithm provides promising results where it achieves load balancing without affecting delay. The ABC-Smart Ant algorithm [7] proposed some modifications to the ABC scheme where it was reported to outperform it.

This paper presents several enhancements to the ACO strategy. The proposed QoS-Aware routing strategy, the traditional routing strategies (LS and DV), the ABC [6] and ABC-Smart Ants [7] algorithms were implemented using the NS-2 network simulator environment. Simulation results show that significant improvements were achieved where throughput gains of 8% and a reduction in delay for time critical applications of 38.4% were registered. This resulted in an improved overall level of QoS, a property which is essential for today's multimedia traffic.

This paper is organized as follows. The following two sections describe the ABC [6] and the ABC-Smart Ants [7] algorithms, respectively. Section IV presents the proposed QoS-Aware strategy while the simulation results are presented in the following section. The final comments and conclusion are provided in Section VI.

II. ANT BASED CONTROL

Consider there are N nodes in the network, and a node N_i with $n(i)$ neighbours is characterized by a pheromone table $R_i = [r_{l,m}^i]_{N,k(i)}$ that has N rows and k columns, where each row represents a possible destination node and each column the possible next node. Further, $r_{l,m}^i$ represents the probability that a given message, whose destination is N_l , be routed from the current node N_i to the node N_m as shown in Fig. 1.

Packets representing artificial ants are launched at regular intervals from all the nodes in the network towards a random destination. As these ants travel along the nodes they update the routing tables, therefore sharing the "knowledge" they gather about a portion of the network with other ants. Data packets and ants are routed in the network according to the strongest pheromone trail given by the largest probability in the routing table for the required destination.

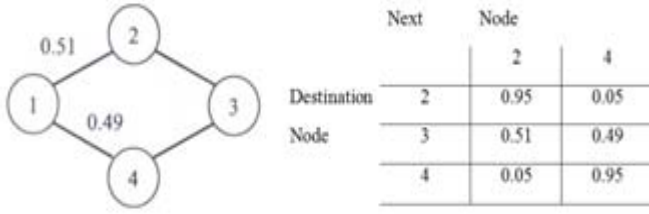


Fig. 1. The pheromone table for node 1

Each node visited by the ant has its routing table updated. When an ant reaches its destination it updates the local routing table and is deleted from the network. The update of the routing tables is performed using a pheromone value Δp . The route taken by the ant is reinforced according to (1) while all the other entries for the same destination decay according to (2):

$$r_{l,m}^i(t+1) = \frac{r_{l,m}^i(t) + \Delta p}{1 + \Delta p} \quad (1)$$

$$r_{l,s}^i(t+1) = \frac{r_{l,s}^i(t)}{1 + \Delta p} \quad (2)$$

The relative and absolute reinforcement of the probability $r_{l,m}^i$ depends on its initial value, such that smaller probabilities are more reinforced. This feature would allow new routes to be discovered more rapidly when the shortest route becomes congested or is momentarily unavailable. Thus, an alternative uncongested route can be selected increasing the throughput of the network.

The pheromone value Δp used in this algorithm is chosen in order to optimize the quality of the path taken by the ants. This factor should therefore change depending on the length of the path taken and on the time spent by ants waiting at nodes due to congestion. The ABC algorithm employs a method called ageing where the pheromone value Δp decreases progressively with the age of an ant (3)

$$\Delta p = \frac{a}{age} + b \quad (3)$$

where parameters a and b are numerical values obtained heuristically and dependent on the network characteristics. The ant's *age* is given by

$$age = r + D \quad (4)$$

where r represents the number of hops performed by an ant and D is the time that an ant spends at a node waiting in a queue, and is modelled using

$$D = c(e^{-ds}) \quad (5)$$

where parameters c and d are again constant variables obtained heuristically, while s is the spare capacity of the queue.

As *age* is a function of the time spent at a node's queue, the paths determined by the ants are optimized both in terms of delay and throughput, effectively performing load balancing. Furthermore, in order to avoid the reinforcement of congested links, the flow of ants passing through congested links is delayed.

The authors in [6] propose an exploration factor $f [0, 1]$ which avoids that ants fall in a "freezing" state where routes become static. In this way, an ant is allowed to choose a random path with probability f , and a path according to the pheromone table with probability $1-f$. This exploration factor ensures that no route achieves extremely high pheromone values, and also that alternative routes are always present and freshly updated. In this way ants are able to react rapidly to sudden changes in topologies, facilitating the discovery of new routes. Data packets on the other hand will always follow paths having the highest pheromone concentration, ensuring that these packets are always routed according to the optimal path in terms of throughput and delay.

III. ABC SMART ANTS

The ants suggested in the ABC scheme only update the entry in the routing table corresponding to the source of the ant which limits the speed by which the optimal path is determined. Guérin [8] suggested a modification such that an ant would upgrade all the entries in the routing table corresponding to all the intermediate nodes visited by the ants. This modification requires that the ants remember the address of all the visited nodes and the time at which such visit occurred. The pheromone value Δp is derived according to

$$\Delta p = \frac{a}{T_i - T_m} + b \quad (6)$$

where T_i is the ant's absolute age and T_m is the time at which the intermediate node was visited. This was done in order to eliminate synchronization issues.

This modification allows the routing algorithm to perform better than the original ABC scheme, as the routing tables converge and adapt more rapidly to network variations. The drawback of the ABC-Smart Ant algorithm is the increased complexity of the suggested ants since such ants now require not only to remember their source and the absolute time, but also all the intermediate nodes visited, and the absolute time at which each intermediate node is visited by the artificial ant. Furthermore the size of the ant packets increase proportionally to the number of visited nodes. These limitations force this algorithm to be limited to small Ad Hoc networks and make it unsuitable for Internet backbone applications.

IV. PROPOSED QOS-AWARE ACO

The proposed scheme is similar to the ABC-smart Ant scheme at its core. However, several modifications were implemented in order to improve the performance of the ABC-smart Ant algorithm. These modifications are explained in the following sub-sections.

A. Non-Linear Ant Launching

Ants were launched at a rate proportional to the current traffic load and not at a constant interval as suggested by the other ACO schemes. In the schemes proposed in [6], [7], artificial ants were launched from any node in the network at regular time intervals. Such time interval should be tuned to maximize the performance of the system, where a small number of ants may not be enough to obtain a solution. On the

other hand, the performance of the system will degrade if the number of ants being launched is too large. A more appropriate way is to increase the number of ants being launched as the network becomes more and more congested. This will facilitate the discovery of alternative uncongested routes when experiencing heavy traffic conditions. Similarly, under low traffic conditions, fewer ants should be launched allowing more stable routing tables.

In the proposed algorithm, a non-linear ant launching procedure was adopted. Artificial ants are launched based on the queue size of the nodes, which reflects the degree of traffic experienced. To prevent that a node launches too few artificial ants, a maximum time-interval representing the maximum time between two consecutive ants was adopted. Similarly, to prevent that an excessive number of artificial ants are generated, a minimum time interval threshold was set. The rate of ant launching was then varied proportional to the queue length within these bounds. These bounds were determined through extensive simulations.

B. Verification Mechanism

The ACO schemes considered in this paper [6], [7] forward packets according to the entries in the pheromone tables, and do not consider the condition of the link. This implies that, if the link is currently unavailable, the packet would be lost. The proposed algorithm adopts a verification mechanism which checks the state of the link before forwarding any packet onto it. If the link is available, data packets are forwarded according to the highest entry in the pheromone table, while if the link is not available the next best available link is chosen.

The verification function continuously checks the status of the links using both artificial ants and data packets. In this way ants are able to rapidly interact with sudden network change caused by link failures as ants are not allowed to choose unavailable links and therefore alternative available links are enforced and made available for future communication. Therefore, the proposed algorithm adapts immediately to rapid topology changes.

C. Priority Scheduling

The traditional ACO schemes do not distinguish between the different data types flowing over a network. This means that all data packets are routed in the same way, as these techniques are mainly designed to reduce congestion and improve network performance. However, modern communication networks have to support multimedia data flows, requiring a different level of QoS.

The proposed method is able to discriminate between the different data flows, therefore providing a better QoS. This was achieved by assigning higher priorities to delay intolerant information such as VoIP and Videoconferencing applications, while other content such as FTP is assigned a lower priority. The priority levels were used to determine the order in which packets were scheduled once they arrive at the destination node. A pre-emptive priority scheduling scheme was adopted.

The packets representing the artificial ants were set to a medium priority. In this way, ants would still be affected by the congestion at the nodes, penalizing ants choosing

congested paths, while at the same time ants are still processed at an acceptable rate.

D. Smart Initialization

The pheromone tables adopted by traditional ABC methods are generally initiated at random. This means that initially a setup time is required. The proposed algorithm eliminates this converging period through the use of a smart initialization process. To reduce the convergence period, a scheme similar to the one adopted in [9] was applied. This scheme assigns an initial greater probability value to the neighbouring nodes, when such nodes are listed as destination.

The biasing of probabilities in the case when prior knowledge of the network topology was not redeemable, is performed using

$$r_{l,m}^i = \frac{1}{n_i} + \frac{3}{2} * \frac{(n_i-1)}{n_i^2} \quad (7)$$

where $r_{l,m}^i$ represents the probability that a given message with destination l is routed from the current node i to node m , and n_i represent the number of neighbours of the i th node. The other entries in the same row of the table are normalized using

$$r_{l,s}^i = \frac{1-r_{l,m}^i}{n_i-1} \quad (8)$$

E. Fixed Size Ants

The ABC-Smart Ant scheme [7] enforces an artificial ant to upgrade all the entries in the routing table corresponding to all the intermediate nodes visited by the ants. This modification showed encouraging results. However, ants were required to remember the address of all the visited nodes and the time at which such visit occurred, causing the ants to grow proportionally to the number of nodes visited.

This dynamicity in the ant's size is undesirable, since varying size packets require a longer processing time than fixed sized packets [10]. The proposed algorithm defines fixed sized ants. This implied that the number of updates performed at a routing table by a single ant would be limited to a predefined number v . In this way, ants still retain the possibility of multiple updates, therefore being responsive to topology changes, whilst retaining fixed size ants.

V. EXPERIMENTAL RESULTS

All the algorithms mentioned in the previous sections were simulated, using the NSFNET network topology of the National Science Foundation (United States), with 14 nodes and links of 1.5Mbps as in [11]. To test the performance of the different algorithms, each algorithm was tested under three different traffic conditions; light, medium and heavy traffic with an inter-arrival time of 333ms, 114ms, and 85ms respectively. Various experiments were performed for each traffic condition. The heuristic coefficients used in the equations for the reinforcement and decay of the routing table probabilities were set as follows: $a = 0.08$, $b = 0.005$, $c = 80$, $d = 0.075$, $f = 10$ and $v = 3$.

Fig. 2 illustrates the average instantaneous delay under transient traffic. It can be observed that under light traffic condition, the schemes exhibit similar delays. However, as the traffic load increases, the ACO based schemes initially perform worse, with their performance improving with time until they outperform the traditional routing schemes. This is mainly due the fact that the routes in the ACO schemes are not fixed but adapt to the traffic conditions, causing a transient effect. Note that the proposed scheme is the fastest to respond. This is attributed to the ability to launch ants at a rate proportional to the load, together with the ability that a single ant is able update multiple entries in the routing table.

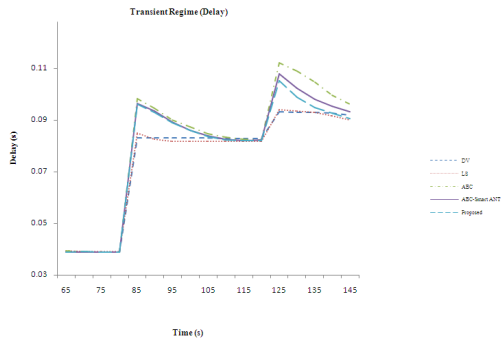


Fig. 2. Average instantaneous delay under transient traffic

The ACO based algorithms perform considerably better than traditional routing schemes, since as shown in Fig. 3, they manage to deliver successfully up to 8% more packets under heavy traffic conditions. This occurs since the LS and DV schemes optimize their paths based on the shortest paths and their performance will drop if the shortest paths are congested. Conversely, the proposed scheme optimize their routing tables both by the physical length and the congestion level experienced by each node, thus providing load balancing. Furthermore, the proposed scheme provides the best throughput relative to the other ACO mechanisms considered in this paper.

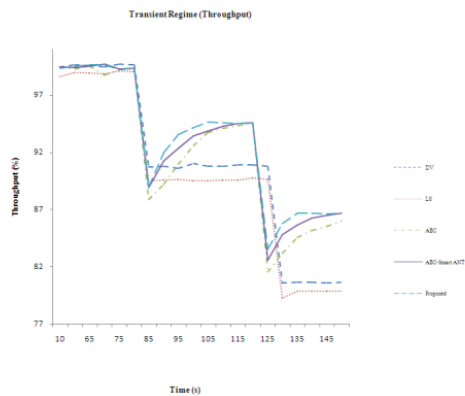


Fig. 3. Average throughput under transient traffic

When a link failure occurs, all the simulated schemes suffer an increase in delay. However the DV and LS algorithms are less prone to such increase. This occurs as such algorithms update their routing tables all at once to reflect the topology

changes, causing new routes to be immediately identified and stabilized. On the other hand, due to the adaptation process exhibited by the ACO routing algorithms, the increase in delay is not constant but transient. Furthermore the alternative paths discovered by the ants immediately after a link failure are not optimized unlike the routes found by the DV and the LS schemes, providing a more acute increase in.

However, as shown in Fig. 4, the delay provided by the proposed system is quite small relative to both DV and LS routing procedures, especially under heavy traffic conditions where it is almost negligible. Furthermore, as shown in Fig. 5, the proposed algorithm outperforms all other schemes in terms of throughput where gains of around 7% in throughput relative to traditional approaches were registered at low and medium traffic conditions.

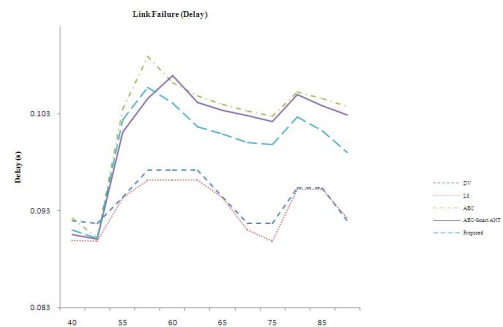


Fig. 4. Average instantaneous delay under link failure

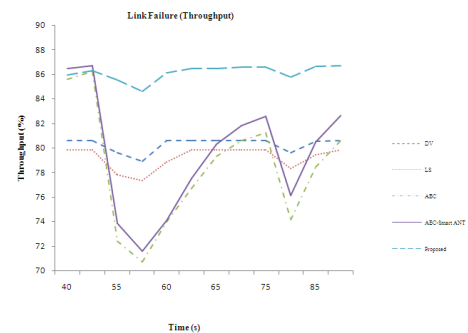


Fig. 5. Average throughput under link failure

This occurs as whilst the other ACO schemes continue to forward packets over the broken link, the proposed scheme verifies the availability of the link before forwarding any packet. In this way no packets are lost immediately after a link is broken. This implies that the proposed scheme is the most robust in terms of throughput of all the simulated schemes, achieving and maintaining a high throughput even under link failure conditions.

Another consideration is the initialization process. Due to the random initialization of the routing tables, and the adaptation process of the ACO schemes, a convergence period is present before useful and coherent routes are present in the routing tables. As shown in Fig. 6, if no prior information of the network topology is available, the proposed algorithm has a short convergence time, comparable to the one achieved by the ABC-smart algorithm, even though the number of updates

performed by the proposed scheme is limited due to its fixed size ant packets. This performance is attributed to the smart initialization procedure, which effectively allows up to 20% of all the routes to be determined immediately.

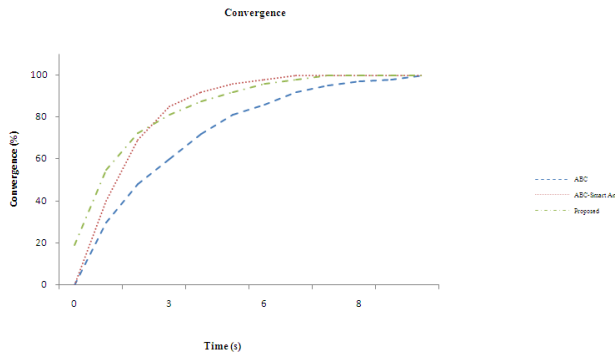


Fig. 6. Convergence Speed of the ACO schemes

One should note that having fixed ant packets is a desirable property as this could allow piggy-backing, where ants packets are embedded within data packets, effectively deferring the overhead cost of the ant packets from the available bandwidth completely. This is on the other hand not possible with the ABC-smart ant scheme, as in this case the ant packets have a varying length depending on the number of nodes an ant visits.

Table I illustrates the performance of the different routing strategies considered in this work in terms of delay. Here, all the schemes except the proposed algorithm apply a First In First Out (FIFO) scheduling procedure, where all the packets are delivered at the same average delay. This implies that delay intolerant applications are treated like delay independent applications, irrelevant of the different QoS requirements. On the other hand, the proposed algorithm adopts a pre-emptive priority based scheduling procedure which is able to discriminate between the different data flows types. The proposed procedure reduces the average delay of time critical applications by 38.4%.

TABLE I
QoS PERFORMANCE BASED ON DELAY CONSTRAINTS

Schemes	Average Delay (Percentage Variation from Mean)		
	Low Priority Data	Medium Priority Data	High Priority Data
DV	0.0%	0.0%	0.0%
LS	0.0%	0.0%	0.0%
ABC	0.0%	0.0%	0.0%
ABC-Smart Ant	0.0%	0.0%	0.0%
Proposed	+32.6%	+5.8%	-38.4%

VI. CONCLUSION

This paper presents a QoS-Aware ACO procedure which launches ants at a rate proportional to the current traffic load. This aids the discovery of new paths when the optimal path

becomes congested and thus minimizes the overheads at low traffic conditions. The Verification procedure manages to detect damaged links immediately and thus minimizes packet. Furthermore, the proposed scheme adopts priority based scheduling where higher priorities are assigned to time critical applications. This procedure has achieved a reduction in average delay of 38.4% relative to the other methods for time critical applications such as VoIP and Videoconferencing applications. In addition, the smart initialization procedure adopted by this proposed solution provides a convergence similar to the Smart-Ant scheme. The Smart-Ant scheme adopts variable ant sizes and thus may not be practical for large networks. On the other hand, the proposed scheme adopts a fixed ant size with minimal loss in performance.

The simulation results presented here clearly demonstrate the superiority of the proposed system. The increase in delay relative to tradition schemes is minimal and is most of the time smaller than the delay provided by the other ACO schemes. Moreover, the proposed system outperforms all the other mechanisms considered in this paper, where throughput gain of up to 8% relative to the traditional routing procedures was achieved.

Based on the simulation results presented in this paper it can be concluded that a commercial implementation of this algorithm is feasible and can be used for large backbone networks. The proposed algorithm is particularly suitable for networks requiring a high throughput, varying traffic oscillations, and where resilience to network perturbations, such as link failures is essential. However due to the slower response to the network changes when compared with LS and DV, it is suggested to adopt LS or DV in transient state.

REFERENCES

- [1] D. Bertsekas, and R. Gallager, *Data Networks*, New Jersey, Prentice-Hill, 1987, pp. 1-30.
- [2] A. S. Tanenbaum, *Computer networks*, Prentice-Hall, Thrid edition, 2003, pp. 1-50.
- [3] B. A. Forouzan, *Data Communication and Networking*, New York, McGraw-Hill, 2007, pp. 59-39.
- [4] Nectcordia, "Static Routing", [Online]. Available at: <http://www.netcordia.com/resources/books/chap-3-static-routing.asp>
- [5] S. Bak, J. Cob, and E. Leiss, *Load Balancing Pouting via Randomization*, XXV Information Latinoamerican Conference, Paraguay, 1999, pp. 999-1010.
- [6] R. Schoonderwoerd, O. Holland, and J. Rothkrantz, "Load Balancing in Telecommunications Networks", *Adaptive Behaviour*, vol. 5, no. 02, pp. 169-207.
- [7] S. Guérin, "Optimisation multi-agents en environnement dynamique: application au routage dans le réseaux de Telecommunications", DEA, dissertation, University of Rennes, 1997.
- [8] M. Dorigo, and G. Di Caro, "Ant Algorithms for Discrete Optimization", *Artificial Life*, vol. 5, pp. 137-172
- [9] B. Baran, "Improved AntNet routing", *ACM SIGCOMM Computer Communication Review*, Vol.31.Issue 2 Supplement, 2001, pp. 42-48.
- [10] L. Barolli, A. Koyama, K. Matsumoto, T. Suganuma and N. Shitorori, "A Genetic Algorithm Based Routing Method using Two QoS Parameters," in *Proc. of Int. Workshop on Database and Expert Systems Applications*, 2002, pp. 2-5.
- [11] NSF, "The Internet Changing the way we communicate", [Online]. Available at <http://www.nsf.gov/about/history/nsf0050/internet/launch.htm>
- [12] X. Cheng, Y.B. Hou, "A Study of Genetic Ant Routing Algorithm," in *Proc. Int. conf. on Machine Learning and Cybernetics*, 2003, pp. 2-5.