Ant Colony Optimization Routing Mechanisms with Bandwidth Sensing

João M. Mapisse, Pedro Cardoso, and Jânio M. Monteiro

***Abstract*—The study and understanding of the social behavior of insects has contributed to the definition of some algorithms that are capable of solving several types of optimization problems. In 1997 Di Caro and Dorigo developed the first routing algorithm for wired networks, called AntNet, using an approach which was inspired in the behavior of ant colonies. At each node, AntNet, similar to others Ant Colony Optimization (ACO) based algorithms, forward ants based in the amount of pheromones present in the links and in response to the node's queue lengths. In this paper, an adaptation of the ϵ-DANTE algorithm for discrete problems, as an IP based routing mechanism, was implemented. We also propose the inclusion of a new parameter for the computation of paths for both the AntNet and the newly proposed algorithm: the available bandwith. Those methods were tested in ns-2 using two dense network architectures and their efficiency is compared with the original AntNet and a Link-State routing algorithm, when considering the transmission of competing traffic flows between distinct nodes.**

***Index Terms*—Swarm Intelligence, Ant Colony Optimization, Bandwidth, Routing.**

I. Introduction

In recent years there has been a significant change in the Internet traffic trends [1] with a significant increment in video streaming and Peer-to-Peer (P2P) communications. P2P applications for instance not only contribute to nearly half of the traffic in the Internet [1] but also introduce a very dynamic behavior due to peer churn [2], constantly changing congestion points in IP networks.

In these type of scenarios, the constant monitoring of the congestion level in nodes and links can potentially distribute load in different paths, trying to identify the better ones among all the available solutions, which would minimize the end-to-end delay of packets and increase the end-to-end bandwidth.

The social behavior of insects, particularly ants, has inspired a number of optimization techniques. One of those is called Ant Colony Optimization (ACO) [3] (a subclass of the Swarm Intelligence (SI) methods) that are metaheuristics which use artificial ants/agents to look for the best solution in discrete optimization problems[1] [3], [5].

The first ACO algorithm designed for network routing was called AntNet [6] and implemented an adaptive routing of packets in wired networks. In AntNet, packets are forwarded in an intelligent way, making a balance between the queue lengths and pheromones. Using experimental tests this algorithm got the highest level of performance when compared with the OSPF, SPF, BF, Q-R and PQ-R algorithms [7].

However, the AntNet algorithm does not directly consider the available bandwidth of the links, forwarding packets based on queues lengths and pheromones values, which contribute to the reduction of the end-to-end delay, but doesn't react to the available bandwidth in a set of alternative paths.

On the other hand, classical implementations of the SI algorithms tend to require a large amount of computational time or resources to be competitive, specially when an earlier obtained solution needs to be improved [3]. Those circumstances make the use of hybrid algorithms a possible solution to refine the results obtained with the SI methods. In [8] a new ACO algorithm for discrete problems, called ϵ-DANTE was defined and evaluated, which is a hybridization of the Ant Colony Optimization Algorithm with a depth search procedure, putting together an oriented/limited depth search.

In this paper we have adapted the ϵ-DANTE algorithm as an IP routing mechanism and have extend it, together with the AntNet, to include an instant bandwidth measurement, resulting in two new protocols: the AntNetBw and ϵ-DANTENetBw. In these protocols the available bandwidth analysis is combined with the information about the queue length and pheromones, trying to avoid the congestion of some links and distribute traffic among distinct nodes of the wired networks. These new mechanisms are afterward compared with the original ones when applied to different network architectures and traffic patterns.

The rest of the paper is organized as follows. AntNet and the ϵ-DANTENet algorithms will be presented in Section II. In section III we will introduce the newly proposed AntNetBw and ϵ-DANTENetBw routing algorithms. Using these routing mechanisms, in section IV we present the experimental results achieved by these new protocols when compared with previous solutions. Finally in section V we will conclude the paper and present some future work.

II. Ant Colony Based Routing

In order to describe the new algorithms we will start by introducing the most relevant mechanisms of the AntNet and the considered adaptation of the ϵ-DANTE algorithm to obtain a method which we will call ϵ-DANTENet.

A. AntNet algorithm

AntNet was created to find the minimum delay path to forward packets. This was inspired by the behavior of ant

J. M. Mapisse, P. Cardoso, and J. M. Monteiro are with the Institute of Engineering, University of Algarve, Faro, Portugal, e-mails: jmapisse@hotmail.com, {pcardoso, jmmontei}@ualg.pt

J. M. Monteiro is also with INOV, Lisbon, Portugal

Manuscript received November 15th, 2010

[1]We should notice that although ACO are in general applied to discrete problems, some were the cases where they were applied, with satisfactory results, to continuous problems [4].

colonies making it an active algorithm, which depend on the traffic demand in a specific path, measured at periodic intervals.

To describe the AntNet algorithm, we consider a packet being sent from a source node (s) to a destination node (d), in a wired network. The two types of ants used by this algorithm are:

- **forward ant**, is an agent represented by $F_{s\to d}$, that travels from source node s to destination node d, and seek the minimum delay path between them;
- **backward ant**, is an agent represented by $B_{s\to d}$, created in destination node, d, by $F_{s\to d}$ and travels backwards to the source node, in the same path but in the opposite direction, updating the pheromone and routing tables in all nodes of the path.

$F_{s\to d}$ packets carry a set of information that allows the pseudo random finding of paths between source and destination and $B_{s\to d}$ packets enable the updating of the pheromone and routing tables.

From each network source node, s, agents $F_{s\to d}$ are launched towards specific destination nodes, d, at regular intervals, Δt, and concurrently with the data traffic. The forward ant generation processes happen without any form of synchronization among the nodes. The destination node, d, of each agent generated at node s, is chosen according to equation (1).

$$P_d = \frac{bits_{sd}}{\sum_{d'\in N} bits_{sd'}} \tag{1}$$

where P_d gives the probability to choose the destination node d, and $bits_{sd}$ is the number of bits that are being transmitted from the source node s to the destination node d.

The forward ant travels from one node to an adjacent one towards its destination and, while moving, it collects information about the delay, together with the node identifiers of the followed path.

At each intermediate node, k, a stochastic decision policy is applied to select the next hop, n, according to equation (2).

$$P_{nd} = \frac{\tau^k_{nd} + \alpha l_n}{1+\alpha(|N_k|-1)}, \quad \forall n \in N_k \tag{2}$$

where:

- N_k is the set of neighbor nodes of k.
- τ^k_{nd} represent the pheromone between k and n (with $\sum_{n\in N_k} \tau^k_{nd} = 1$);
- $l_n \in [0,1]$ represent the heuristic coefficient determined by equation
$$l_n = 1 - \frac{q_n}{\sum_{n'=1}^{|N_k|} q_{n'}} \tag{3}$$
where q_n represents the number of packets in the queue between nodes k and node n;
- $\alpha \in [0,1]$ represents the coefficient of l_n; and

The information about the traveling time and the node identifiers collected by the forward ants along its path are saved in the ant's packet. To avoid loops, when the pseudo-randomly selection of a next hop results in a node that was already visited by the ant, then the agent is silently discarded.

Once arrived at the destination, the forward ant is converted to a backward ant ($B_{s\to d}$). The forward ant transfers its memory to the backward ant, which travels back to the source node using the same path as the $F_{s\to d}$, but using a higher priority in routers, such that the pheromone information in the nodes is quickly updated.

At each visited node, k, the backward ant updates the local pheromone and routing information, with the data collected between nodes s and k. Once they have returned to their source node, the agent is removed from the network.

See [6] for a detailed description of the AntNet method.

B. *ϵ-DANTENet algorithm*

The ϵ-Depth ANT Explorer Network (ϵ-DANTENet) algorithm is a hybrid method for routing in wired networks, that combines ACO with a depth search. The ϵ-DANTENet is our propose of adapted of the ϵ-DANTE method which is an hybridization of the ACO algorithm (see [8], [9] for detailed description of the method along with computational results for several multiple objective optimization problems).

The ϵ-DANTENet algorithm was designed with the aim of further exploring good solutions from intermediate nodes k to destination node d, within the set of solutions obtained by the AntNet method. As we will see, this is done without the need to rebuild each solution completely from source nodes, i.e., at certain nodes, the backward ant generate new forward ants that will use part of the solution that was built by the original forward ant. To understand this algorithm we begin introducing the following concepts:

a) Identical with the AntNet:

- **forward ant**, is an agent represented by $F^{Ant}_{s\to d}$, that travel from source node s to destination node d, and seek the minimum delay path between them;
- **backward ant**, is an agent represented by $B^{Ant}_{s\to d}$, created in destination node, d, by $F^{Ant}_{s\to d}$ and travel to the source node, in the same path but in the opposite direction, updating the pheromone tables in all the nodes of the path.

b) Different from the AntNet:

- **forward DANTE**, is an agent represented by $F^{DANTE}_{k\to d}$ that is generated by $B^{Ant}_{s\to d}$ at an intermediate node k and aim to implement a limited depth search. The $F^{DANTE}_{k\to d}$ travels from the node k to node d, looking for alternative paths with minimum delays different from the one found by the $F^{Ant}_{k\to d}$. As mentioned, the creation of this intermediate agents favors the exploration of the search space without the need to rebuild the solutions from the origin node, s, which *per se* will induce a smaller computational effort to obtain different solutions.
- **backward DANTE**, is an agent represented by $B^{DANTE}_{s\to d}$, created in destination node, d, by $F^{DANTE}_{k\to d}$. The $B^{DANTE}_{s\to d}$ aims to use information about the time and the node identifiers collected by the $F^{DANTE}_{k\to d}$ to update the pheromone and routing tables from node d to node s. The main difference between $B^{DANTE}_{s\to d}$ and $B^{Ant}_{s\to d}$ is the fact that former it will not generate any king of forward ants.

The memory structure of $B_{s\to d}^{DANTE}$ is the same as the $B_{s\to d}^{Ant}$. However, the memory structure of $F_{k\to d}^{DANTE}$ also includes information about its trip time.

In conclusion, the "AntNet part" of this algorithm uses the $F_{s\to d}^{Ant}$ and $B_{s\to d}^{Ant}$ ants, while the "depth search part" uses the $F_{k\to d}^{DANTE}$ and $B_{s\to d}^{DANTE}$.

The condition to create the $F_{k\to d}^{DANTE}$ is that its next hop, calculated by equation (2), has to be different from the $(k-1)$-th and $(k+1)$-th nodes of the path (which is in the memory of the $F_{s\to d}^{Ant}$). The $F_{k\to d}^{DANTE}$ is then forwarded as the $F_{s\to d}^{Ant}$. This will enforce the production of new solutions which will further explore the the search space, as aimed. As for the $F_{s\to d}^{Ant}$, the $F_{k\to d}^{DANTE}$ transfers the collected information about the traveling time and the node identifiers to the $B_{s\to d}^{DANTE}$ at destination node. The $B_{s\to d}^{DANTE}$ is then forwarded to s as the $B_{s\to d}^{Ant}$.

III. AntNetBw and ϵ-DANTENetBw: ACO routing algorithms with available bandwidth

The aim of forward agents in AntNet and ϵ-DANTENet is to find optimal paths for routing packages from a source node to a destination node. At the intermediate node k a stochastic decision rule is applied for the forward ants/DANTEs deciding the next hop. The decision rule considers the size of the queue and the pheromones level (equation (2)). In order to evaluate the response of these algorithms to the congestion level of some nodes, we have included an heuristic associated with the available bandwidth between the links to choose. In this way, the AntNetBw and ϵ-DANTENetBw were created, wich are improvements of previous AntNet and ϵ-DANTENet procedures. In these new algorithms the forward ants/DANTEs, $F_{s\to d}^{Ant}$ and $F_{k\to d}^{DANTE}$, are forwarded according to the probability:

$$P_{nd} = \left(\frac{\tau_{nd}^{k} + \alpha l_n}{1 + \alpha(|N_k| - 1)} \right) \cdot (1 - Util_n^k), \quad \forall n \in N_k \qquad (4)$$

where $Util_n^k$ represents the used bandwidth of the link between nodes k and n, and the $1 - Util_n^k$ is the available bandwidth between these nodes (the remaining parameters were explained along with equation (2)).

According with equation (4), the routing decision of the ant (or DANTE) now responds to the pheromones level, the size of queues, the memory and the available bandwidth. The use of the available bandwidth tries to avoid forwarding the ant or DANTEs through links where the bandwidth is more limited.

IV. Experimental Results

In order to be able to compare the efficiency of these protocols when distributing traffic in different nodes of a wired network, we have modified the implementation of AntNet by Lima [10], and implemented the ϵ-DANTENet, the AntNetBw, and the ϵ-DANTENetBw in the Network Simulator 2 (ns-2) [11].

Firstly, the AntNet method was adapted by implementing the depth search mechanism of the DANTEs algorithm, resulting in ϵ-DANTENet. Finally, the routing mechanisms with bandwidth sensing was implemented by the introduction of equation (4) in AntNet and ϵ-DANTENet code (replacing equation (2)), resulting in AntNetBw and ϵ-DANTENetBw described in section III.

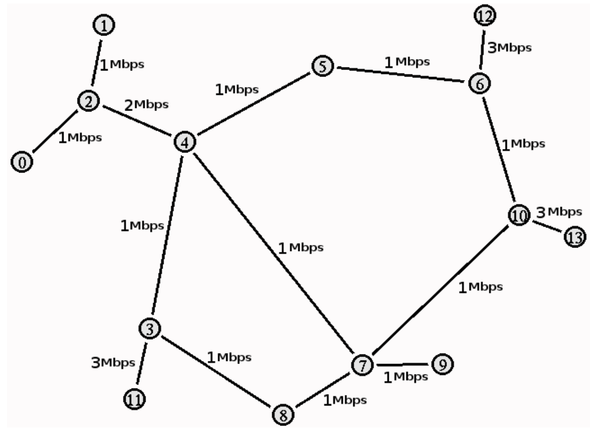

Fig. 1. Topology 1: 14 nodes and 15 links.

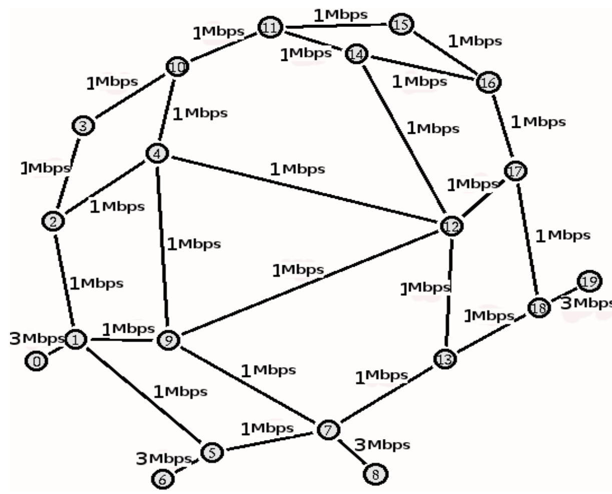

Fig. 2. Topology 2: 20 nodes and 28 links.

TABLE I
Traffic for Topology 1.

UDP flows with a rate of 900 kbps		
Transmitter	Receiver(s)	Time Interval of each transmission (seconds)
0	8	from 0.0 to 30.0
1	9	from 1.5 to 30.0
9	0 and 3	from 0.0 to 30.0
11	10	from 0.75 to 30.0
12	3	from 0.5 to 30.0
13	1 and 11	from 1.0 to 30.0

In the following we compare these protocols using two different topologies represented in Figures 1 and 2. The traffic used for each of the topologies are represented in Tables I and II, respectively.

For each of the network architectures and traffic patterns, 25 simulations were run, with 30 seconds each one. We have measured the average number of lost UDP packets for each of the traffic sources. Parameters Δt (time between ants) and α were set to 0.05 and 0.45, respectively.

TABLE II
TRAFFIC FOR TOPOLOGY 2.

UDP flows with a rate of 900 kbps		
Transmitter	Receivers	Time Interval of each transmission (seconds)
0	11, 16 and 19	from 0.0 to 30.0
6	10, 15 and 17	from 0.0 to 30.0
8	3, 11 and 16	from 0.0 to 30.0
19	0, 3 and 11	from 0.0 to 30.0

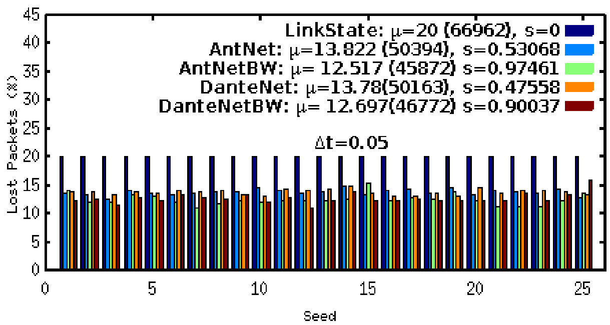

Fig. 3. Percentage's means of total number of UDP lost packets for algorithms Link-State, AntNet, ϵ-DANTENet, AntNetBw and ϵ-DANTENetBw for Topology 1 (total number of losses is in parenthesis).

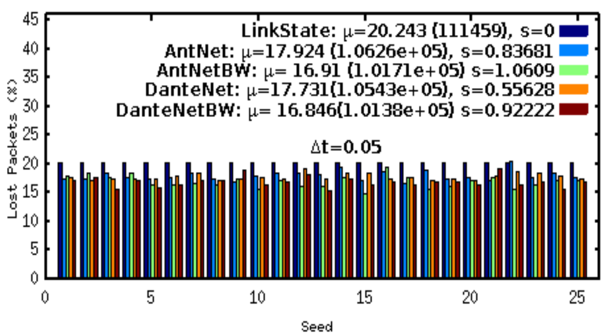

Fig. 4. Percentage's means of total number of UDP lost packets for algorithms Link-State, AntNet, ϵ-DANTENet, AntNetBw and ϵ-DANTENetBw for Topology 2 (total number of losses is in parenthesis).

The results of the average percentage of UDP lost packets for topology 1 and 2 are shown in Figures 3 and 4, respectively.

Given the results shown in Figure 3, the protocols were sorted in terms of average number of lost UDP packets (from lower to higher) resulting in the following list:

1^{st} *AntNetBw* (loss ratios reduction of 9,4% and 39,4% when compared respectively with the AntNet and Link-State algorithms);

2^{nd} *ϵ-DANTENetBw* (loss ratios reduction of 8,1% and 36,5% when compared respectively with the AntNet and Link-State algorithms);

3^{rd} *ϵ-DANTENet* (loss ratios reduction of 0,7% and 31,1% when compared respectively with the AntNet and Link-State algorithms);

4^{th} *AntNet;*

5^{th} *Link-State.*

The same process was performed for the results shown in Figure 3 resulting in the following rating:

1^{st} *ϵ-DANTENetBw* (loss ratios reduction of 6,0% and 16,8% when compared respectively with the AntNet and Link-State algorithms);

2^{nd} *AntNetBw* (loss ratios reduction of 5,7% and 16,5% when compared respectively with the AntNet and Link-State algorithms);

3^{rd} *ϵ-DANTENet* (loss ratios reduction of 1,1% and 12,4% when compared respectively with the AntNet and Link-State algorithms);

4^{th} *AntNet;*

5^{th} *Link-State.*

The results in Figures 3 and 4 show that the algorithms AntNetBw, ϵ-DANTENetBw and ϵ-DANTENet have a better performance in terms of packet losses when compared with the AntNet and Link-State protocols. When comparing the results of the ϵ-DANTENet and the AntNet, we verify that their differences are not very significative. We can conclude that, for the studied cases, the use os the available bandwidth parameter has imposed a better distribution of traffic among links resulting in lower packet loss.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have described two new ACO routing algorithms that include the *available bandwidth* parameter between the links in the forwarding. This new parameter has caused a reduction in the packet losses of wired networks when compared with the AntNet and Link-State protocols.

From the presented work, some future developments can also be proposed, as: to perform a better tuning of the proposed methods, to study their behaviour on other topologies and traffic patterns, and to adapt the methods such that they can distinguish between different types of traffic priorities (a multiple objective model).

REFERENCES

[1] S. Corporation, *Analysis of Traffic Demographics in Broadband Networks*, May 2008.

[2] A. Binzenhofer and K. Leibnitz, "Estimating churn in structured P2P networks," in *International Teletraffic Congress 2007*, June 2007, pp. 630–641.

[3] M. Dorigo and T. Stutzle, *Ant Colony Optimization.* MIT Press, Massachusetts, 2004.

[4] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *European journal of Operational Research*, vol. 185, no. 3, pp. 1155–1173, 2008.

[5] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computer and Operations Research*, vol. 13, pp. 533–549, May 1986. [Online]. Available: http://portal.acm.org/citation.cfm?id=15310.15311

[6] G. Di Caro and M. Dorigo, "Adaptive learning of routing tables in communication networks," in *Italian Workshop on Machine Learning*, Torino-Italy, December 1997.

[7] G. Di Caro and M. Dorigo, "Antnet: Distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Research*, pp. 317–365, 1998.

[8] P. Cardoso, M. Jesus, and A. Márquez, "ϵ-DANTE: an ant colony oriented depth search procedure," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, pp. 1–34, 2010. [Online]. Available: http://dx.doi.org/10.1007/s00500-010-0543-9

[9] P. Cardoso, M. Jesus, and A. Márquez, "Dante: The combination between an ant colony optimization algorithm and a depth search method," in *Hybrid Intelligent Systems, 2008. HIS '08. Eighth International Conference on Hybrid Intelligent Systems*, September 2008, pp. 36–41.

[10] R. Lima, "Aco routing algorithm in practice," November 2009. [Online]. Available: http://antnet.wordpress.com

[11] M. Greis, "Tutorial for the network simulator-ns," 2009. [Online]. Available: http://www.isi.edu/nsnam/ns/tutorial/