TWO ANT COLONY ALGORITHMS FOR BEST-EFFORT ROUTING IN DATAGRAM NETWORKS

GIANNI DI CARO and MARCO DORIGO*

IRIDIA, Université Libre de Bruxelles
50, av. F. Roosevelt, CP 194/6
1050 - Brussels, Belgium
{gdicaro,mdorigo}@ulb.ac.be

ABSTRACT

In this paper we present two versions of AntNet, a novel approach to adaptive learning of routing tables in wide area best-effort datagram networks. AntNet is a distributed multi-agent system inspired by the stigmergy model of communication observed in ant colonies. We report simulation results for AntNet on realistically sized networks using as performance measures throughput, packet delays and resources utilization. Our tests show that both instances of AntNet show superior performance with respect to the current Internet routing algorithm (OSPF), some improved old Internet routing algorithms (SPF and distributed adaptive Bellman-Ford), and recently proposed forms of asynchronous online Bellman-Ford (Q-routing and Predictive Q-routing).

KEYWORDS: Adaptive routing, ant colony optimization, distributed multi-agent systems.

1 INTRODUCTION

In this paper we consider the problem of adaptive routing in communications networks: we focus on routing for wide area datagram networks with irregular topology and best-effort service, the most remarkable example of such networks being the Internet.

The goal of every routing algorithm is to direct traffic from sources to destinations optimizing at the same time several measure of network performance as throughput (correctly delivered bits per time unit), packet delays and resources utilization. The general problem of determining an optimal routing algorithm can be stated as a multi-objective optimization problem in a non-stationary stochastic environment. Information propagation delays, and the difficulty to model the whole network dynamics under arbitrary traffic patterns, make the general routing problem intrinsically distributed. Routing decisions can only be made on the basis of local and approximate information about the current and the future network states.

The adaptive routing algorithms we propose in this paper, called *AntNet*, are distributed and mobile multi-agent systems well matching the above characteristics of the general routing problem. The design of our algorithms has been inspired by previous works on ant colonies and, more generally, by the notion of *stigmergy* [1, 2], that is, the indirect communication taking place among individuals through local, persistent (or slowly changing) modifications induced in their environment. Real ants have been shown to be able to find shortest paths using a stochastic decision policy based only on local information represented by the pheromone trail deposited by other ants [3].

Algorithms that take inspiration from ants' behavior in finding shortest paths have recently been successfully applied to several discrete optimization problems [2, 4, 5, 6, 7, 8, 9]. In ant colony optimization each one of a set of concurrent artificial ants makes use of a stochastic local search strategy to build a solution to the combinatorial problem under consideration. The whole set of ants collectively search for high quality solutions by a cooperative effort mediated by indirect communication of information on the problem structure they collect while building solutions.

Similarly, in AntNet, artificial ants (agents) collectively solve the routing problem by a cooperative effort in which stigmergy, mediated by the network nodes, plays a prominent role. By using a stochastic routing policy based on local (public) and private information ants concurrently and asynchronously explore the network and collect useful information. While exploring, the ants adaptively build probabilistic routing tables and local models of the network status using indirect and non-coordinated communication of the information they collect.

We report on the behavior of two different versions of AntNet, compared to the following routing algorithms: Open Shortest Path First (OSPF) [10], Shortest Path First (SPF) [11], distributed adaptive Bellman-Ford (BF) [12], Q-routing [13], and PQ-routing [14]. We consider realistic experimental conditions tested on the Japanese NTT private backbone and on a set of 100 and 150-node randomly generated networks. In all cases AntNet algorithms show the

*This work was supported by a Madame Curie Fellowship awarded to Gianni Di Caro (CEC TMR Contract N. ERBFMBICT-961153). Marco Dorigo is a Research Associate with the Belgian FNRS.

best performance and the most stable behavior. Among the competitors SPF and BF are the best performing.

2 THE COMMUNICATION NETWORK MODEL

We have developed (in C++) a discrete-event realistic simulator of IP-like datagram networks. The instance of the communication network is mapped on a directed weighted graph with N nodes with limited buffer space. All the links are viewed as bit pipes characterized by a bandwidth (bits/sec) and a transmission delay (sec), and are accessed following a statistical multiplexing scheme. All the traveling packets are subdivided in two classes: data and routing packets. All the packets in the same class have the same priority, so they are queued and served only on the basis of a first-in-first-out policy, but routing packets have a higher priority than data packets. Service times for the two classes are generated following different probabilistic models. Packets can be discarded on node arrival because of lack of incoming buffer space and/or expired time-to-live. No arrival acknowledgment or error notification packets are generated back to the source and only a very simple flow control mechanism is implemented.

Because of the tight coupling between adaptive routing and flow and congestion control components we chose not to implement a "real" transport layer. In fact, we wanted to check the behavior of our algorithm and of its competitors in conditions which minimize the number of interacting components that should be designed together to well match each other.

3 THE ANTNET-CL AND ANTNET-CO ALGORITHMS

AntNet is composed of two sets of homogeneous mobile agents [15], called in the following *forward* and *backward* ants. In AntNet we retain the core ideas of the ant colony optimization paradigm, but we have translated them to match a distributed, dynamic context, different from combinatorial optimization[1]. Ants communicate in an indirect way according to the stigmergy paradigm, through the information they concurrently read and write in two data structures stored in each network node k:

(i) an array $\mathcal{M}_k(\mu, \sigma^2)$ of data structures defining a simple parametric statistical model of the traffic distribution for all destinations d, as seen by the local node k,

(ii) a routing table, organized as in distance-vector algorithms [16]; the table stores for each pair (d, n) a probability value P_{dn}

$$\sum_{n \in \mathcal{N}_k} P_{dn} = 1,\ d \in [1, N],\ \mathcal{N}_k = \{neighbors(k)\}$$

which expresses the goodness of choosing n as next node when the destination node is d.

We present two versions of AntNet, that in the following we call respectively AntNet-CL and AntNet-CO. AntNet-CL (the same algorithm as presented in [17], where was called simply AntNet) is our first implementation of AntNet and it has been developed to manage routing in connection-less best-effort networks, while AntNet-CO (presented for the first time in this paper) is a more reactive version of AntNet-CL, that better matches the requirements of routing in high-speed connection-oriented networks where best-effort services are provided concurrently with Quality-of-Service sessions[2].

We first informally describe the behavior of AntNet-CL, while AntNet-CO will be described in the following, highlighting the way it differs from AntNet-CL.

- At regular intervals, from every network node s, a forward ant $F_{s \to d}$, is launched, with a randomly selected destination node d. Destinations are chosen to match the current traffic patterns.
- Each forward ant selects the next hop node using the information stored in the routing table. The next node is selected, following a random scheme, with a probability proportional to the goodness of each not still visited neighbor node and to the local queues status. If all neighbors have been already visited a uniform random selection is applied considering all the neighbors.
- In case the selected link is not currently available, the forward ant waits its turn in the low-priority queue of the data packets, where it is served on the basis of a FIFO policy.
- The identifier of every visited node k and the time $T_{s \to k}$ elapsed since its launching time to arrive at this k-th node are pushed onto a memory stack $S_{s \to d}(k)$ carried by the forward ant.
- If a cycle is detected, that is, if an ant is forced to return to an already visited node, the cycle's nodes are popped from the ant's stack and all the memory about them is destroyed.
- When the ant $F_{s \to d}$ reaches the destination node d, it generates a backward ant $B_{d \to s}$, transfers to it all of its memory, and then it dies.
- The backward ant makes the same path as that of its corresponding forward ant, but in the opposite direction and making use of high-priority queues. At each node k along the path it pops its stack $S_{s \to d}(k)$ to know the next hop node.

[1] AntNet is not the only algorithm based on the ant colony metaphor that has been applied to routing. Schoonderwoerd et al. [9] have considered the routing problem in connection-oriented networks. Their approach is different from ours because their network and their algorithm was modeled after a very specific type of telephone network. Because of this it was impossible to re-implement and compare their algorithm with ours in the context of datagram best-effort routing we consider in this paper.

[2] AntNet-CO is currently under development and testing to manage fair-share connection-oriented networks. In this paper we present its simplest implementation, applied to best-effort connection-less networks.

- Arriving in a node k coming from a neighbor node f, the backward ant updates $\mathcal{M}_k$ and the routing table for all the entries corresponding to every node i on the path $k \rightarrow d$ followed by ant $F_{k\rightarrow d}$ starting from the current node k.
  - The sample means and variances of the model $\mathcal{M}_k(\mu, \sigma^2)$ are updated with the trip times $T_{k\rightarrow i}$ stored in the stack memory $S_{s\rightarrow d}(k)$.
  - The routing table is changed by incrementing the probabilities P_{if} associated with node f and the nodes i, and decreasing (by normalization) the probabilities P_{in} associated with the other neighbor nodes n. Trip times $T_{k\rightarrow i}$ experienced by the forward ant $F_{s\rightarrow d}$ are used to assign the probability increments.

In AntNet-CO the above basic behavior is identical except that (i) forward ants make use of high-priority queues as backward ants, (ii) they do not carry in the memory stack any information about their experienced times $T_{s\rightarrow k}$, and (iii) backward ants update the routing tables in the visited nodes using estimates of ants' trip times. These estimates are computed at each node k, using a local statistical model $\mathcal{L}_k^l$ capturing the depletion dynamics of each of the local links l. In this paper we use the simplest type of model $\mathcal{L}$, that is, given the number of bits q_l of the data packets waiting in the queue of l, the virtual trip time to reach the desired neighbor is computed as $d_l + (q_l + s_a)/B_l$, where B_l is the bandwidth of the link, s_a is the size of the ant packet and d_l is the link's propagation delay. In both AntNet-CO and AntNet-CL forward ants apply a stochastic policy to discover a feasible good path: in AntNet-CL forward ants behave exactly like data packets and the delays they experience are used by backward ants to score the quality of the paths they crossed, while, in AntNet-CO forward ants quickly discover a path that is scored by backward ants using trip times locally estimated by means of the models $\mathcal{L}$.

In both AntNet algorithms, $T_{k\rightarrow d}$ is the only explicit feedback signal we have: it gives an indication about the goodness r of the followed route because it is proportional to its length from a physical point of view (number of hops, transmission capacity of the used links, processing speed of the crossed nodes) and from a traffic congestion point of view. The problem is that $T_{k\rightarrow d}$ can only be used as a reinforcement signal. In fact, it cannot be associated with an exact error measure, given that we do not know the optimal trip times, which depend on the net load status. The values stored in the model $\mathcal{M}_k$ are used to score the trip times by assigning a goodness measure $r \equiv r(T_{k\rightarrow d}, \mathcal{M}_k), r \in \ (0, 1]$ (r is such that the smaller $T_{k\rightarrow d}$, the higher r). This dimensionless value takes into account an average of the observed values and of their dispersion: $(1 - W_{k\rightarrow d}/T_{k\rightarrow d}) + \Delta(\sigma, W)$, where $W_{k\rightarrow d}$ is the best trip time experienced over an adaptive time window, and $\Delta(\sigma, W)$ is a correcting term (the rationale behind this choice for r is discussed in [17]); r is used by the current node k as a positive reinforcement for the node f the backward ant $B_{d\rightarrow s}$ comes from. The probability P_{df} is increased by the computed reinforcement value r: $P_{df} \leftarrow P_{df} + (1 - P_{df})r = P_{df}(1 - r) + r$. In this way, the probability P_{df} will be increased by a value proportional to the reinforcement received and to the previous value of the node probability (that is, given a same reinforcement, small probability values are increased proportionally more than big probability values). Probabilities P_{dn} for destination d of the other neighboring nodes n implicitly receive a negative reinforcement by normalization. That is, their values are reduced so that the sum of probabilities will still be 1: $P_{dn} \leftarrow P_{dn}(1 - r)$.

It is important to remark that every discovered path receives a positive reinforcement in its selection probability. In this way, not only the (explicit) assigned value r plays a role, but also the (implicit) ant's arrival rate. An important aspect of the AntNet algorithm is that the routing tables are used in a probabilistic way not only by the ants, but also by the packets. This mechanism allows an efficient distribution of the data packets over all the good paths and has been observed to significantly improve AntNet performance. A node-dependent threshold value avoids the choice of low probability links.

As a last consideration, note the critical role played by ant communication. In fact, each ant is complex enough to solve a single sub-problem but the global routing optimization problem cannot be solved efficiently by a single ant. It is the interaction between ants that determines the emergence of a global effective behavior from the network performance point of view. The key concept in the cooperative aspect lies in the indirect and non-coordinated way communication among ants happens (stigmergy [1]). We used stigmergy as a way of recursively transmitting, through the nodes' data structures, the information associated with every "experiment" made by each ant.

4 ROUTING ALGORITHMS USED FOR COMPARISON

The following algorithms, belonging to the various possible combinations of static and adaptive, distance vector and link state classes [16], have been implemented and used to run comparisons. OSPF (static, link state) is our implementation of the official Internet routing algorithm [10] (since we did not consider failure conditions the algorithm reduces to static shortest path routing). SPF (adaptive, link state) is the prototype of link-state algorithms with dynamic metric for link costs evaluations. A similar algorithm was implemented in the second version of ARPANET [11]. We implemented it with state-of-the-art flooding algorithms and link cost metrics [18]. Link costs are evaluated over moving windows using a link usage metric based on the fraction of time the link has been used during the last observation window. This metric was the most effective among the several we considered. BF (adaptive, distance-vector) is an adaptive implementation of the distributed Bellman-Ford algorithm with dynamic metrics [12]. Link costs are evaluated as in SPF above. Q-R (adaptive, distance-vector) is the Q-routing algorithm as proposed in [13]. This is an online asynchronous version of the Bellman-Ford algorithm. PQ-R (adaptive, distance-vector) is the Predictive Q-routing algorithm [14], an extension of Q-routing.

5 EXPERIMENTAL SETTINGS

Networks - In our experiments we used NTTnet, the Japanese NTT fiber-optic corporate backbone, and a set of randomly generated networks of 100 and 150 nodes. NTTnet has 57 nodes and 162 bi-directional links, link bandwidth is of 6 Mbit/sec, while propagation delays range between 1 to 5 msec. NTTnet is not well balanced. The distance between a pair of nodes in term of hops ranges from 1 to 20, while the ratio between the mean connectivity degree and the number of nodes is about 0.05. The random networks have mean connectivity degree slightly greater than 3 and the number of links per node ranges from 2 to 9. Link bandwidths are set to 1.5 Mbit/sec and propagation delays range from 1 to 10 ms. All nets have null link and node fault probabilities, local buffers of 1 Gbit, and packets maximum time to live set to 15 sec.

Traffic patterns - Traffic is defined in terms of open sessions between a pair of active applications situated on different nodes. We considered a Poisson (P) distribution to shape the arrival of new sessions on each node, that is, inter-arrival times are negative exponentially distributed. The Poisson process can be identical over all the nodes (Uniform Poisson, UP), or it can be different for each node (Random Poisson, RP). For all the session types, packet sizes, packet inter arrival times and the total number of generated bits follow a negative exponential distribution.

Performance metrics - We used throughput (delivered bits/sec), data packets delay (sec) and routing overhead (routing bandwidth utilization / available bandwidth). For packets delay we report the whole empirical data distribution, that takes into account the intrinsic variability of packet delays.

Algorithms parameters - In AntNet, the generation interval of the forward ants is set to 0.3 (sec) and the size of the ant packet is 24 +8N_h bytes, where N_h is the incremental number of hops made by the forward ant. In OSPF, SPF, and BF, the length of the time interval between two consecutive routing information broadcasting and the length of the time window to average link costs are the same, and they are set to 0.8 or 3 seconds, depending on the experiment. In Q-R and PQ-R the transmission of routing information is data-driven. For SPF and OSPF, the size of the routing packet sent from a generic node n is 64 + 8$|\mathcal{N}_n|$ bytes, where$|\mathcal{N}_n|$ is the number of neighbors of node n. For Q-R and PQ-R, the routing packet size is 12 bytes, while for BF is set to 24 + 12N bytes, with N equal to the number of nodes in the network. For all the algorithms processing time for routing packets have been assigned depending on the network size and on raw estimates of available processing power.

6 RESULTS AND DISCUSSION

Experiments reported in this section compare AntNet-CL and AntNet-CO with the previously described routing algorithms. All experiments are averaged over 10 trials. Parameters values for traffic characteristics are given in the figures' captions with the following meaning: MSIA is the mean of the sessions inter arrival time distribution and MPIA is the mean of the packet inter arrival time distribution. For all the experiments the mean of the packet size distribution is set to 4096 bit, and the mean of the total number of bits produced by each session is set to 2 Mb.

In all the reported experiments, the traffic load was chosen to be “heavy”, that is, we set the values of the traffic patterns parameters to values that caused the network to reach a state very close to saturation, to better evidentiate the differences among competing algorithms. In fact, when the traffic load is low, almost all the algorithms perform similarly. On the other hand, if the traffic load is too high, then a reasonable assumption is that it is a temporary situation. If it is not, structural changes to the network characteristics should be in order.

In figure 1 are reported experimental results on the NTTnet for a non-uniform Poisson traffic load (RP) distribution. The throughput's curve in the outer graph of figure 1 shows that all the algorithms but OSPF are able to deliver approximately the same throughput, while the small, inner graph, shows how the two AntNet algorithms keep packets' delays at the same level, which is much lower than that of all their competitors.

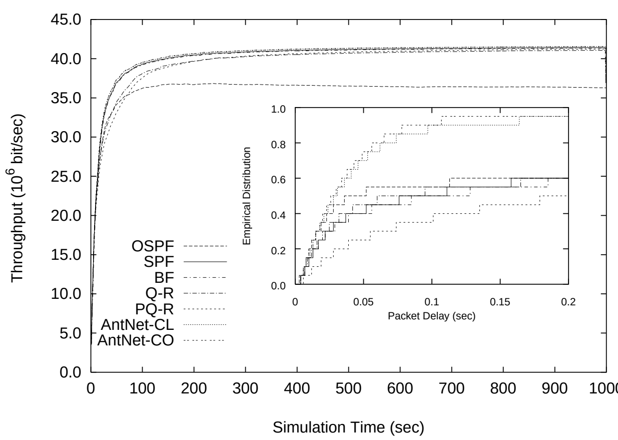

Figure 1: *NTTnet*: A COMPARISON OF ANTNET WITH FIVE COMPETING ALGORITHMS FOR A HEAVY NON-UNIFORM POISSON TRAFFIC (RP). AVERAGE OVER 10 TRIALS. MSIA=2.8, MPIA=0.05.

Figure 2 shows results for a set of 100-node randomly generated networks with UP load. Reported data are the average over 10 trials, where for each trial a different random network has been used. In this case all the algorithms were able to deliver the same throughput, while, once again, differences in the distribution of packet delays are striking. AntNet-CO is by far the best one, followed by AntNet-CL, while all the competitors perform around 30%-40% worse.

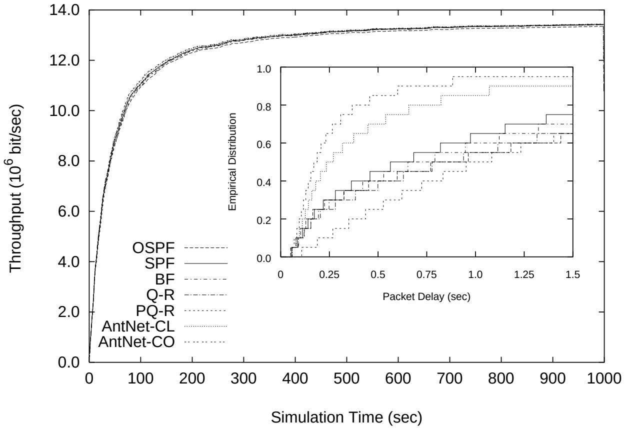

Figure 2: *100-NODE RANDOM NETS*: A COMPARISON OF ANTNET WITH FIVE COMPETING ALGORITHMS FOR A HEAVY UNIFORM POISSON TRAFFIC (UP). AVERAGE OVER 10 TRIALS USING A DIFFERENT RANDOMLY GENERATED 100-NODE NETWORK IN EACH TRIAL. MSIA=15.0, MPIA=0.005.

As a last experiment, in figure 3 we present results for a set of 150-node randomly generated networks with a very heavy RP load. Reported data are the average over 10 trials, where for each trial a different random network has been used. In this case, there are differences among the

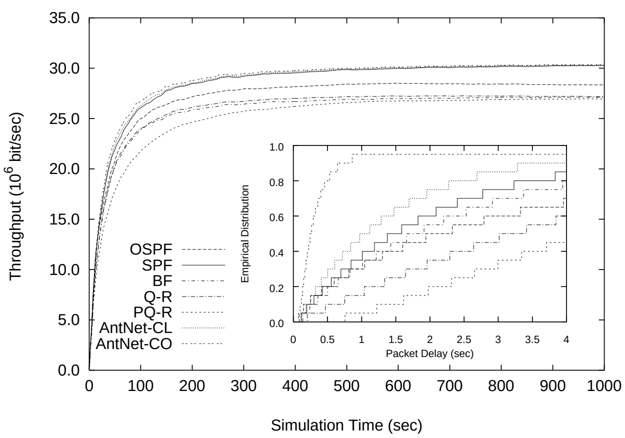

Figure 3: *150-NODE RANDOM NETS*: A COMPARISON OF ANTNET WITH FIVE COMPETING ALGORITHMS FOR A HEAVY NON-UNIFORM POISSON TRAFFIC (RP). AVERAGE OVER 10 TRIALS USING A DIFFERENT RANDOMLY GENERATED 150-NODE NETWORK IN EACH TRIAL. MSIA=10.0, MPIA=0.005.

algorithms' behavior both for throughput and packet delays. Only AntNet-CL, AntNet-CO and SPF are able to follow the generated throughput without losses, OSPF behaves only slightly worse, while all the other algorithms can deliver a throughput about 35% lower. Concerning packet delays, AntNet-CO is still by far the best performing algorithm. AntNet-CL is the second best one, but it keeps delays much higher than AntNet-CO, about four times higher considering the 90th percentile. SPF keeps delays much higher than AntNet-CO, and about 60% higher than AntNet-CL on the 85th percentile. BF follows, but it had much worse performance on throughput. OSPF, Q-R and PQ-R perform rather poorly.

In table 1 are reported results concerning resources utilization by the routing algorithms as measured by the routing traffic overhead, that is, the ratio between the generated routing traffic and the total available bandwidth. Each row in the table refers to a previously discussed experiment (figs. 1-3). Although AntNet's overhead is higher than that of some of its competitors, it must be considered that (i) the relative weight of the routing packets on the net resources is still negligible, (ii) this slightly higher resources consumption is compensated by the much higher performance it provides, and (iii) increasing the routing packet production rate for the best competitors SPF and BF, does not increase, in general, their performance, because of their sensitivity to oscillations.

Table 1: *ROUTING OVERHEAD*: RATIO BETWEEN THE BANDWIDTH OCCUPIED BY THE ROUTING PACKETS AND THE TOTAL AVAILABLE NETWORK BANDWIDTH. ALL DATA ARE SCALED BY A FACTOR OF 10^{-3}.

	NTT - RP	100 - UP	150 - RP
AntNet-CL	4.41	33.33	62.89
AntNet-CO	3.35	54.63	106.5
OSPF	0.14	1.66	2.43
SPF	3.02	14.78	21.46
BF	1.18	11.23	106.7
Q-R	3.36	2.39	5.90
PQ-R	6.37	4.16	8.88

From these results it is clear that AntNet algorithms perform better than both classic and recently proposed algorithms. Among the competitors SPF shows the best performance, followed by BF, while OSPF has the global worse performance. AntNet-CO showed always the best performance and the difference with the performance obtained by all the other algorithms increases with the size of the test networks. AntNet-CL behaves always in a satisfying way but its performance appears to quickly degrade with respect to AntNet-CO as the size of the networks increases.

In general, differences among algorithms performances can be understood on the basis of the different degree of adaptivity and of speed with which the different algorithms respond to traffic conditions changing in the space and in the time (e.g., the very low performance of OSPF is mainly due to the lack of use, differently from all the others, of an adaptive metric).

We identified some aspects of the AntNet algorithms that make them successful (discussed in detail in [17]): (i) AntNet is the only algorithm exploring the whole network, concurrently with the data flow (Q-R and PQ-R do some exploration, but strictly data driven, while BF and SPF do not explore at all), (ii) the information AntNet uses and stores at each node is richer and organized in a less critical way than that of its competitors, (iii) AntNet uses probabilistic tables, that allow for a better redistribution of the traffic and provide a built-in exploration mechanism, (iv) node local estimates are not directly propagated to other nodes, as on the contrary done by all the competitors, making AntNet very robust to locally wrong estimates, (v) AntNet experimentally shows to be very robust to the frequency with which routing tables are updated, that is, to the ants launching rate; on the contrary, this is a very critical aspect in SPF and BF, for which there is no simple way to set the related parameters, being traffic and topology dependent.

The better performance of AntNet-CO with respect to its predecessor AntNet-CL can be understood in terms of its higher reactivity. In fact, AntNet-CO's forward ants do not wait in the data queues. In this way, the information is collected and propagated faster, and it is more up-to-date with respect to the current network status, even if it is based on raw estimates. These characteristics become more and more important as network sizes grow and paths become longer and longer. In fact, in these cases, AntNet-CL can present very long delays in gathering and releasing traffic information across the network, making completely out-of-date the information it uses to update the routing tables.

7 CONCLUSIONS

In this paper we proposed AntNet-CL and AntNet-CO, two versions of AntNet, a novel approach for adaptive routing in communications networks inspired by previous work on artificial ants colonies in combinatorial optimization. Using a realistic simulator of best-effort datagram networks, we compared AntNet to a set of state-of-the-art algorithms and we used the NTT Japanese backbone network as well as two sets of randomly generated networks of 100 and 150 nodes, as benchmark problems. In all the experiments we ran, both AntNet algorithms had by far the best distribution of packet delays, and they were among the best algorithms as far as throughput was concerned. AntNet-CO performance was significantly better than AntNet-CL's, with the difference between the two algorithms increasing with network size. AntNet algorithms were observed to be very robust under the different traffic conditions we tested and they were able to quickly reach a stable behavior. AntNet-CO and AntNet-CL, as well as all the competitor algorithms, had a negligible impact on the use of network bandwidth.

REFERENCES

[1] P. P. Grassé, La reconstruction du nid et les coordinations interindividuelles chez *bellicositermes natalensis et cubitermes sp.* La théorie de la stigmergie: essai d'interprétation du comportement des termites constructeurs, *Insectes Sociaux*, 6, 1959, 41–81.

[2] E. Bonabeau, M. Dorigo, and G. Theraulaz, *From Natural to Artificial Swarm Intelligence* (Oxford University Press, in press).

[3] R. Beckers, J. L. Deneubourg, and S. Goss, Trails and U-turns in the selection of the shortest path by the ant Lasius Niger, *Journal of Theoretical Biology*, 159, 1992, 397–415.

[4] D. Costa and A. Hertz, Ants can colour graphs, *Journal of the Operational Research Society*, 48, 1997, 295–305.

[5] M. Dorigo, *Optimization, Learning and Natural Algorithms (*in Italian*)*, PhD thesis, Dipartimento di Elettronica e Informatica, Politecnico di Milano, IT, 1992.

[6] M. Dorigo and L. M. Gambardella, Ant colony system: A cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computation*, 1(1), 1997, 53–66.

[7] M. Dorigo, V. Maniezzo, and A. Colorni, Positive feedback as a search strategy, Technical Report 91-016, Dipartimento di Elettronica e Informatica, Politecnico di Milano, IT, 1991.

[8] M. Dorigo, V. Maniezzo, and A. Colorni, The ant system: Optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics–PartB*, 26(1), 1996, 29–41.

[9] R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz, Load balancing in telecommunications networks, *Adaptive Behavior*, 5(2), 1996, 169–207.

[10] J. T. Moy, *OSPF Anatomy of an Internet Routing Protocol*, (Addison-Wesley, 1998).

[11] J. M. McQuillan, I. Richer, and E. C. Rosen, The new routing algorithm for the ARPANET, *IEEE Transactions on Communications*, 28, 1980, 711–719.

[12] D. Bertsekas and R. Gallager, *Data Networks*, (Prentice-Hall, 1992).

[13] J.A. Boyan and M.L. Littman, Packet routing in dinamically changing networks: A reinforcement learning approach, In *Advances in Neural Information Processing Systems 6 (NIPS6)*, 1994, 671–678.

[14] S.P. Choi and D.-Y. Yeung, Predictive Q-routing: A memory-based reinforcement learning approach to adaptive traffic control, In *Advances in Neural Information Processing Systems 8 (NIPS8)*, 1996, 945–951.

[15] P. Stone and M. M. Veloso, Multiagent systems: A survey from a machine learning persective, Technical Report CMU-CS-97-193, Carnegie Mellon University, Pittsburgh, PA, 1996.

[16] M. E. Steenstrup, editor, *Routing in Communications Networks*, (Prentice-Hall, 1995).

[17] G. Di Caro and M. Dorigo, AntNet: Distributed stigmergetic control for communications networks, Technical Report 98-01, IRIDIA, Université Libre de Bruxelles, 1998, (Accepted for publication in the *Journal of Artificial Intelligence Research (JAIR))*.

[18] A. U. Shankar, C. Alaettinoğlu, K. Dussa-Zieger, and I. Matta, Performance comparison of routing protocols under dynamic and static file transfer connections, *ACM SIGCOMM Computer Communication Review*, 22(5), 1992, 39–52.