An analysis of the different components of the AntHocNet routing algorithm

Frederick Ducatelle, Gianni A. Di Caro, and Luca M. Gambardella*

Istituto "Dalle Molle" di Studi sull'Intelligenza Artificiale (IDSIA)
Galleria 2, CH-6928 Manno-Lugano, Switzerland
{frederick,gianni,luca}@idsia.ch

Abstract. Mobile ad hoc networks are a class of highly dynamic networks. In previous work, we developed a new routing algorithm, called AntHocNet, for these challenging network environments. AntHocNet has been designed after the Ant Colony Optimization (ACO) framework, and its general architecture shares strong similarities with the architectures of typical ACO implementations for network routing. On the other hand, AntHocNet also contains several elements which are new to ACO routing implementations, such as the combination of ant-based path sampling with a lightweight information bootstrapping process, the use of both reactive and proactive components, and the use of composite pheromone metrics. In this paper we discuss all these elements, pointing out their general usefulness to face the multiple challenges of mobile ad hoc networks, and perform an evaluation of their working and effect on performance through extensive simulation studies.

1 Introduction

Mobile Ad Hoc Networks (MANETs) [1] are networks in which all nodes are mobile and communicate with each other via wireless connections. Nodes can join or leave at any time. There is no fixed infrastructure. All nodes are equal and there is no central control or overview. There are no designated routers: nodes serve as routers for each other, and data packets are forwarded from node to node in a multi-hop fashion. MANETs are useful to bring wireless connectivity in infrastructureless areas or to provide instantaneous connectivity free of charge inside specific user communities and/or geographic areas. However, the control of a MANET is very challenging. Its topology and traffic patterns are defined by the present users, their positions and radio ranges. The effectively available bandwidth is defined by the characteristics of the wireless signal between the nodes, and by the amount of simultaneous contention to access the shared wireless medium. Due to the mobility and the constant arrival/departure of users,

* This work was supported by the Future & Emerging Technologies unit of the European Commission through project "BISON: Biology-Inspired techniques for Self-Organization in dynamic Networks" (IST-2001-38923) and by the Swiss Hasler Foundation through grant DICS-1830.

all these characteristics, which make up the mode of the network, change over time, and different modes can coexist in different parts of the network.

Routing is at the core of the functioning of a MANET, and the challenges mentioned above call for a fully adaptive, multi-modal routing controller. We believe that the multidimensional complexity of the task makes it necessary to include multiple learning, adaptive, and behavioral components in the design of the routing algorithm. This is the approach we followed in *AntHocNet* [2–6]. It combines *Monte Carlo path sampling and learning using ant agents*, which is characteristic of the ACO framework [7], with an *information bootstrapping process*, which is typical for dynamic programming and some reinforcement learning approaches [8]. Operating the two learning mechanisms at different speeds allows to obtain an adaptivity, robustness and efficiency which neither of the subcomponents could offer on its own. Moreover, the use of both proactive and reactive behaviors allows to both anticipate and respond in timely fashion to sudden disruptive events. AntHocNet's design also includes the use of multiple metrics (e.g., number of hops and signal quality) in the definition of the pheromone variables used to guide ant decisions.

AntHocNet was inspired by previous work on ACO routing for wired networks [9], but its composite design represents a departure from previous instances of ACO algorithms for routing. The effective integration of a bootstrapping-based mechanisms within a typical ACO architecture is in fact an approach which is innovative in general terms, and not only concerning the application of ACO to network problems. Furthermore, the way we integrated the two mechanisms in AntHocNet is rather general and could be applied with success also to different application domains. Moreover, while the combined use of reactive and proactive ant generation, as well as the use of a composite pheromone metric, are not totally a novelty in ACO routing algorithms, the way these schemes are used is original and general (e.g., see [10–12] for examples of other ACO algorithms for MANETs).

The purpose of this paper is to report an experimental analysis of the role and effect of these different design components of the algorithm. In particular, we show the effect on performance of using bootstrapping and proactive components, of adopting different choices for the composite pheromone metric used to guide ant activities, and of selecting different levels of exploration. Even if this *sensitivity analysis* is specific for AntHocNet, we believe that to a certain extent the reported results can also provide general insights about all the considered issues, and in particular about the integration of ant-based path sampling with pheromone bootstrapping mechanisms.

The general effectiveness of AntHocNet's integrated approach was assessed in a number of papers [2–6] and a technical report [13]. Over a wide range of scenarios with different characteristics in terms of mobility, data traffic load, modality, etc., AntHocNet always showed excellent performance compared to state-of-the-art MANET routing algorithms such as AODV [14] and OLSR [15]. In this paper we focus on the sensitivity analysis and we do not report any further results concerning AntHocNet's general performance.

The rest of the paper is organized as follows. In Section 2 we provide a concise description of AntHocNet (for more details the reader can consult the mentioned references). Section 3 describes the experimental methodology and the general characteristics of the simulation environment. In Section 4 and its subsections, we report the results of the experimental analysis and discuss them.

2 The AntHocNet routing algorithm

In MANET jargon AntHocNet is termed a *hybrid* algorithm since it makes use of both reactive and proactive strategies to establish routing paths. It is *reactive* in the sense that a node only starts gathering routing information for a specific destination when a local traffic session needs to communicate with the destination and no routing information is available. It is *proactive* because as long as the communication starts, and for the entire duration of the communication, the nodes proactively keep the routing information related to the ongoing flow up-to-date with network changes. In this way both the costs and the number of paths used by each running flow can reflect the actual status of the network, providing an optimized network response. The reactive component of the algorithm deals with the phase of *path setup* and is totally based on the use of *ACO ant agents* to find a good initial path. Routing information is encoded in node *pheromone tables*. The proactive component implements *path maintenance and improvement*, proactively adapting during the course of a session the paths the session is using to network changes. Path maintenance and improvement is realized by a combination of ant path sampling and slow-rate *pheromone diffusion*: the routing information obtained via ant path sampling is spread between the nodes of the MANET and used to update the routing tables according to a bootstrapping scheme that in turn provide main guidance for the ant path exploration. *Link failures* are dealt with using a local path repair process or via explicit notification messages. *Stochastic decisions* are used both for ant exploration and to distribute data packets over multiple paths.

In the following we provide a concise description of each of these components. The component dealing with link failures is not described since it is neither central for the algorithm nor relevant for the sensitivity analysis reported here.

2.1 Metrics for path quality and pheromone tables

Paths are implicitly defined by tables of pheromone variables playing the role of routing tables. An entry $\mathcal{T}^i_{nd} \in \mathbb{R}$ of the pheromone table $\mathcal{T}^i$ at node i contains a value indicating the estimated goodness of going from i over neighbor n to reach destination d. Since AntHocNet only maintains information about destinations which are active in a communication session, and the neighbors of a node change continually, the filling of the pheromone tables is sparse and dynamic. Several different metrics, such as *number of hops, end-to-end delay, signal quality, congestion, etc.*, can be used to define the goodness of a path. AntHocNet makes use of a combination of these metrics to define the pheromone variables.

The effect of different combinations of metrics is studied in Subsection 4.1.

2.2 Reactive path setup

When a source node s starts a communication session with a destination node d, and it does not have routing information for d, it broadcasts a *reactive forward ant*. At each node, the ant is either unicast or broadcast, according to whether or not the current node has pheromone information for d. If information is available, the ant chooses its next hop n with the probability P_{nd} which depends on the relative goodness of n as a next hop, expressed in the pheromone variable $\mathcal{T}^i_{nd}$:

$$P_{nd} = \frac{(\mathcal{T}^i_{nd})^\beta}{\sum_{j \in \mathcal{N}^i_d} (\mathcal{T}^i_{jd})^\beta}, \quad \beta \geq 1, \tag{1}$$

where $\mathcal{N}^i_d$ is the set of neighbors of i over which a path to d is known, and β is a parameter which controls the exploratory behavior of the ants. If no pheromone is available, the ant is broadcast. Since it is quite likely that somewhere along the path no pheromone is found, in the experiments we normally use a high value of β to avoid excessive ant proliferation. Due to subsequent broadcasts, many duplicate copies of the same ant travel to the destination. A node which receives multiple copies of the same ant only accepts the first and discards the other. This way, only one path is set up initially. During the course of the communication session, more paths are added via the proactive path maintenance and exploration mechanism discussed in the next subsection.

Each forward ant keeps a list of the nodes it has visited. Upon arrival at the destination d, it is converted into a *backward ant*, which travels back to the source retracing the path. At each intermediate node i, coming from neighbor n, the ant updates the entry $\mathcal{T}^i_{nd}$ in the i's pheromone table. The way the entry is updated depends on the path quality metrics used to define pheromone variables. For instance, if the pheromone is expressed using the number of hops as a measure of goodness, at each hop the backward ant increments an internal hop counter and uses the inverse of this value to locally assign the value τ^i_d which is used to update the pheromone variable $\mathcal{T}^i_{nd}$ as follows: $\mathcal{T}^i_{nd} = \gamma\mathcal{T}^i_{nd} + (1-\gamma)\tau^i_d,\ \gamma \in [0,1]$. For different metrics, the calculation of τ^i_d is more complex but follows the same logic. For instance, if delay is used, the ant needs to incrementally calculate at each node a robust estimate of the expected delay to reach the destination.

2.3 Proactive path maintenance and exploration

During the course of a communication session, source nodes periodically send out *proactive forward ants* to update the information about currently used paths and try to find new and better paths. They follow pheromone and update pheromone tables in the same way as reactive forward ants. Such continuous proactive sampling of paths is the typical mode of operation in ACO routing algorithms. However, the ant sending frequency needed to faithfully keep track of the constant network changes is in general too high for the available bandwidth. Moreover, to find entirely new paths, excessive blind exploration through random walks or broadcasts would be needed. Therefore, we keep ant sending rate low but ant actions are integrated with a lightweight process combining *pheromone diffusion and bootstrapping*. This process provides a second way of updating pheromone on existing paths, and can give information to guide exploratory ant behavior.

Pheromone diffusion is implemented using *beacon messages* broadcast periodically and asynchronously by the nodes to all their neighbors. In these messages, the sending node n places a list of destinations it has information about, including for each destination d its best pheromone value $\mathcal{T}^n_{m^*d}$. A node i receiving the message from n first of all registers that n is its neighbor. Then, for each destination d listed in the message, it derives an estimate of the goodness of going from i to d over n, combining the cost of hopping from i to n with the reported pheromone value $\mathcal{T}^n_{m^*d}$. We call the obtained estimate $\mathcal{B}^i_{nd}$ *bootstrapped pheromone*, since it is built up bootstrapping on the value of the path goodness estimate received from an adjacent node. The bootstrapped pheromone can in turn be forwarded in the next message sent out by n, giving rise to a bootstrapped pheromone field over the MANET. This is the typical way of calculating estimates followed by all approaches based on *dynamic programming* such the class of distributed Bellman-Ford routing algorithms [16] and the reinforcement learning algorithms derived from Q-learning [8]. However, due to the slow multi-step forwarding, bootstrapped pheromone does not provide the most accurate view of the current situation and has difficulty to properly track the distributed changes in the network. Generally speaking, bootstrapping alone is not expected to work effectively in highly non-stationary environments. However, here the bootstrapped pheromone is obtained via a lightweight, efficient process, and is complemented by the explicit Monte Carlo path sampling and updating done by the ants. In this way we have two complementary updating frequencies in the path updating process. Bootstrapped pheromone is used directly for the *maintenance* of existing paths. That is, if an entry $\mathcal{T}^i_{nd}$ is present in the routing table, $\mathcal{B}^i_{nd}$ is used as a replacement of it. For path *exploration*, bootstrapped pheromone is used indirectly. If i does not yet have a value for $\mathcal{T}^i_{nd}$ in its routing table, $\mathcal{B}^i_{nd}$ could indicate a possible new path from i to d over n. However, this path has never been sampled explicitly by an ant, and due to the slow multi-step process it could contain undetected loops or dangling links. It is therefore not safe to use for data forwarding before being checked. This is the task of proactive forward ants, which use both the regular and the bootstrapped pheromone on their way to the destination. This way, promising pheromone is investigated, and can be turned into a regular path available for data. This increases the number of paths available for data routing, which grows to a full mesh, and allows the algorithm to exploit new opportunities in the ever changing topology.

The effect of varying the number of bootstrapping entries in the beacon messages, the sending rate of proactive ants, and their β exploration exponent is studied respectively in Subsections 4.3, 4.2, and 4.4.

2.4 Stochastic data routing

Nodes in AntHocNet *forward data stochastically* according to the pheromone values. When a node has multiple next hops for the destination d of the data, it randomly selects one of them, with probability P_{nd}. P_{nd} is calculated like for reactive forward ants, using Eq. 1. According to this strategy, a mesh of multiple paths is made available to route data. The number of paths to use is automatically and dynamically selected in function of their estimated quality. The effect of varying β in Eq. 1 for data forwarding is studied in Subsection 4.4.

3 Experimental methodology and characteristics of the simulation environment

For the study of the performance of AntHocNet, we use simulation experiments. This is the most common approach in MANETs, since the complexity of this kind of networks makes analytical evaluations difficult and limited in scope, while the high costs involved in purchasing and configuring hardware limit the use of real testbeds. As simulation software, we use QualNet [17], a commercial simulation package which comes with correct implementations of the most important protocols at all layers of the network protocol stack.

All the simulation tests reported in this paper last 900 seconds, and each data point represents the average taken over 10 runs using different random seeds. The tests are carried out in open space scenarios (see [13] for evaluations of AntHocNet in a structured urban scenario). 100 nodes move in an area of $2400 \times 800\ m^2$. The movements of the nodes are defined according to the random waypoint mobility model [18]. Nodes pick a random destination point in the area, and move to that point with a randomly chosen speed. Upon arrival, they stay put for a fixed pause time, after which a new random destination and speed are chosen. In our experiments, the node speed is chosen uniformly between 0 and 10 m/s, unless stated otherwise. The pause time is always 30 seconds.

Radio signal propagation is modeled with the two-ray ground reflection model, which considers both the direct and the ground reflection path [19]. The transmission range of each node is 250 meters. At the physical and medium access control layers of the network protocol stack, we use the IEEE 802.11b protocol in DCF function with 2 Mbits/s bandwidth. At the application layer, data traffic is generated by 20 constant bit rate (CBR) sources, sending packets of 64 bytes. The CBR sessions start at a random time between 0 and 180 seconds after the start of the simulation, and go on until the end of the simulation. The data rate is 4 packets per second, unless stated differently. CBR uses the UDP protocol at the transport layer. All these settings reflect choices widely adopted in MANET research. Concerning the AntHocNet parameters, if not stated differently, the value of β in Eq. 1 is set to 20, the maximum number of entries in the pheromone diffusion messages is set to 10, and the sending interval for the proactive ants is 2 seconds.

To evaluate the performance of the routing algorithms, we measure the *average end-to-end delay* for data packets and the *ratio of correctly delivered versus sent packets* (delivery ratio). These are standard measures of effectiveness in MANETs. Other metrics which we consider are *delay jitter* and *routing overhead*. Delay jitter is the average difference in inter-arrival time between packets. This is an important metric in quality-of-service networks and in MANETs provides also a measure of the stability of the algorithm's response to topological changes. Routing overhead is a measure of efficiency. We calculate it as the number of control packet transmissions (counting every hop) per data packet delivered. Due to space limitations we do not show the results for jitter and overhead. In any case, for all the tests of Section 4, the results for jitter and overhead always follow the same trends of those for delivery ratio and end-to-end delay.

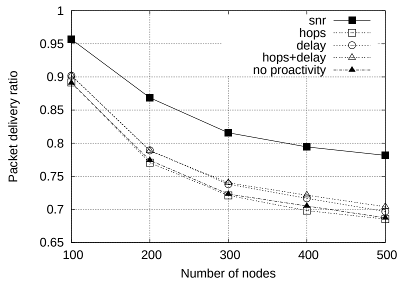
4 Experimental analysis of the different components of AntHocNet

In this section, we study how the different components influence the performance of the algorithm. In 4.1, we investigate the use of different path evaluation metrics for the definition of pheromone values. In 4.2 we study the effect of the frequency with which proactive ants are sent out. In 4.3, we study the importance of the amount of information transmitted in the pheromone diffusion messages and made available for bootstrapping. Finally, in 4.4, we investigate the effect of allowing different levels of exploration in proactive forward ants and for data.

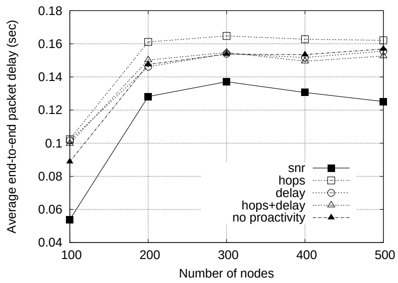
4.1 Using different optimization metrics for pheromone definition

A pheromone value encodes the adaptive estimated goodness of a routing decision. Different path evaluation metrics can be used to measure this goodness. We investigate the use of path length in number of hops (referred to as "hops" in Figure 1), path end-to-end delay ("delay"), number of hops combined with the quality of each hop in terms of signal-to-noise ratio ("snr"), and number of hops combined with end-to-end delay ("hops+delay"). We also plot results for a version of AntHocNet where proactive learning is switched off ("no proactivity") and the "snr" metric is used. Figures 1a and 1b show respectively the delivery ratio and the average end-to-end delay of the different versions of the algorithm.

A first clear result is that the metric combining hops and signal-to-noise ratio is the most effective one. On the other hand, the optimization with respect to the sole number of hops produces the worst results. Even worse than using no proactivity at all with the "snr" metric. Similar results have been reported in the MANET community [20]: paths with a low number of hops usually use long hops, which necessarily have lower signal strength, and can therefore easily loose connection. This diagnosis is confirmed by the fact that combining number of hops and signal-to-noise ratio leads to large improvement in performance. Finally, we point out that also the use of the end-to-end delay alone or together with hops does not give good results. This is partly due to the backoff mechanisms at the MAC layer, that make the single node experiencing fluctuating delays accessing the shared wireless channel. This generates large variations in terms of end-to-end delay. Under these conditions it becomes hard to learn estimates of path latencies which are robust enough to rank the quality of the different paths.



(a)



(b)

Fig. 1. (a) Delivery ratio and (b) average end-to-end delay for AntHocNet using different optimization metrics in scenarios with increasing nodes and constant density.

These results indicate that it is crucial to choose a good path evaluation metric to use in the pheromone model, based on knowledge of the specific network environment. A composite metric is the right way to go.

4.2 Varying the proactive ant send interval

The proactive ant send interval is the time between successive proactive ants in the proactive path maintenance and improvement phase. It defines how often the algorithm looks for path improvements, and therefore how quickly it can adapt to changes. We made tests with send intervals of 0.5, 1, 2, 5, 10, 20, and 50 seconds. The test scenario is the basic scenario described in Section 3. The different curves in Figures 2a and 2b represent tests using different maximum node speeds (so changing the network mobility).

All tests show a similar pattern. A too low ant send interval leads to bad performance, because the network gets flooded by ants. At 2 seconds, there seems to be an optimal send interval. For frequencies lower than that, the performance decays because the algorithm is not sending enough ants to keep up with the changes in the network. For low speeds, this decay is slower since the network changes less fast. However, it is interesting to note that the best send interval value is independent of the node speed. We also did some tests keeping the speed constant on 10 m/s, and varying the data traffic load (not presented here due to space limitations). Although higher traffic load could be expected to leave less space for ants, also there the best ant send interval was always around 2 seconds. The general effect of the use of proactive actions was also shown in previous Figure 1 where it is evident the significant decrease in performance when the proactive mechanisms are switched off (using "snr" metric) with respect to the case of using the same metric but switching proactivity on.

4.3 Varying the number of entries in pheromone diffusion messages

The number of maximum entries in the pheromone diffusion messages defines how much pheromone information is spread at each step of the information bootstrapping process. Concretely, a low number of entries spreads little information

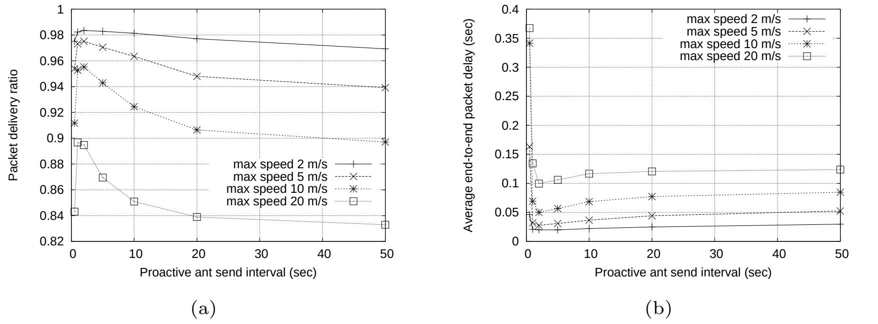


Fig. 2. (a) Delivery ratio and (b) average end-to-end delay for AntHocNet increasing the proactive ant send interval in scenarios with different node speeds.

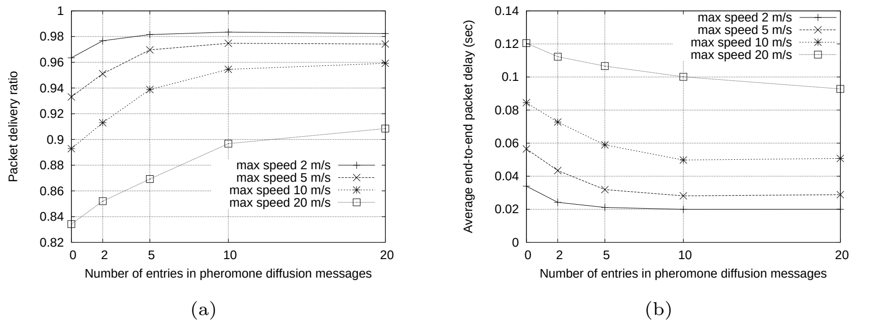


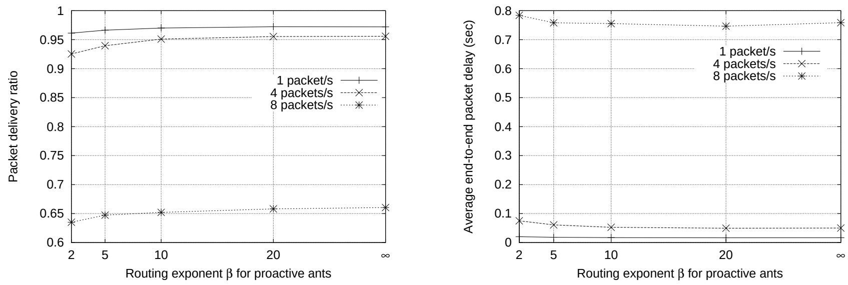
Fig. 3. (a) Delivery ratio and (b) average end-to-end delay for AntHocNet varying the number of entries in pheromone diffusion messages for scenarios with different speeds.

and determines also a slow running of the bootstrapping process. 0 entries is the extreme case where the supporting pheromone diffusion function is disabled and the proactive ants get no guidance. We made tests using 0, 2, 5, 10 and 20 entries. The destinations whose routing information is included in each message are selected randomly out of the known destinations stored in the pheromone table. The test scenario is the basic scenario described in Section 3, with 20 active CBR sessions. Like in Subsection 4.2, we report results for different maximum node speeds. Figure 3 reports delivery ratio and average delay.

The graphs show the importance of the supporting pheromone diffusion process: giving more efficiency to this process allows for better performance. Moreover, for the tested sizes the benefit of the increase in transmitted information is still greater than the negative impact due to the generation of larger messages. As was observed in 4.2, the importance of the efficiency of the learning process decreases for slower changing networks.

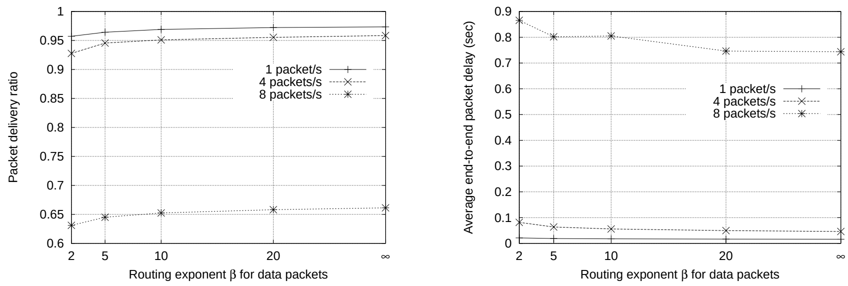
4.4 Varying the routing exponent for ants and data

The exponent β in Equation 1 defines the amount of exploration allowed to the ants during their path search phase. As previously mentioned, for the reactive



(a) (b)

Fig. 4. (a) Delivery ratio and (b) average end-to-end delay for AntHocNet increasing the proactive ant routing exponent in scenarios with different data traffic send rates.



(a) (b)

Fig. 5. (a) Delivery ratio and (b) average end-to-end delay for AntHocNet increasing the data packet routing exponent in scenarios with different data traffic send rates.

forward ants β is set to a quite high value (≈ 20) in order to reduce counter-productive ant proliferation and the establishment of sub-optimal paths at the start of a new data session. On the other hand, proactive ants are meant to explore and check the path improvements suggested by bootstrapped pheromone. Therefore, we studied the effect of varying the degree of exploration allowed to the proactive forward ants by considering β values of 2, 5, 10, 20 and also the case of deterministic choice of the best path. We considered scenarios with data rates of 1, 4 and 8 packets per second for the 20 CBR sessions.

The results reported in Figure 4 shows a large difference in performance for the cases of 1 and 4 packets and that of 8 packets per second. Nevertheless, an equivalent performance response with respect to β is evidenced in all the three different scenarios. Performance increases reducing exploration. The difference between the extreme cases of $\beta = 2$ and deterministic choices (indicated with ∞ in the plots), is small but clear. These results show that in MANETs, exploration at the level of the ants do not really pay back due to constant changes and strong bandwidth limitations that limit the frequency of ant generation and the number of different paths that can be effectively explored. We observed a similar behavior also increasing the degree of exploration in reactive forward ants. On the other hand, in AntHocNet path exploration is implicitly carried out with the low-overhead mechanisms of pheromone diffusion and bootstrapping. New paths are built in a multi-step fashion by bootstrapping on the pheromone information received from the neighbors. Results in Figure 4 says that the best performance is obtained when proactive ants are used to test only the best path indicated by the current combination of regular and bootstrapped pheromone.

Analogous results are reported in Figure 5, that shows the effect of varying the value of the exponent β for data packets. In this case, β controls the amount of multiple paths used to spread data. The results suggest that the best choice is to adopt a deterministic greedy policy for next hop selection. On the other hand, this does not imply that a single path is used to forward the data packets of a same traffic flow. We experimentally observed that multiple paths are actually used even when a deterministic policy is adopted, as the result of the continual proactive updating and addition/removal of paths made available to the running flows. However, the results indicate that an excessive use of multiple paths can easily bring performance down. This is due to the fact that if two paths simultaneously used for the same flow are not radio-disjoint they will interfere with a consequent degradation of performance.

5 Conclusions

MANETs are extremely dynamic network environments. Their multi-modality represents an important challenge for algorithms at all levels of the network protocol stack, and specifically for routing. We addressed these challenges with AntHocNet, a routing algorithm designed after ACO ideas. AntHocNet was introduced in previous work and showed superior performance compared to other state-of-the-art algorithms over a wide range of MANET simulation scenarios [2–6, 13]. In this paper we discussed AntHocNet emphasizing its innovative design, especially with respect to previous ACO algorithms for routing. In particular, we pointed out the fact that AntHocNet is based on the integration of reactive and proactive components, and on the integration of the typical ACO path sampling mechanism with the learning of routing information using an information bootstrapping process. In a detailed experimental study we have investigated the role and the importance of these and other different components of the algorithm, studying their effect on the overall performance. The effectiveness of the use of a composite design to deal with the multiple challenges of MANETs, and in particular the effectiveness of combining ant-based Monte Carlo sampling with pheromone bootstrapping, has been confirmed by the experimental results. Moreover, the experimental analysis has pointed out the need for adopting low-overhead and low-interference strategies for exploration and data forwarding, as well as the importance of defining a composite pheromone metric taking into account different multiple aspects of the network environment.

References

1. Royer, E., Toh, C.K.: A review of current routing protocols for ad hoc mobile wireless networks. IEEE Personal Communications (1999)

2. Di Caro, G., Ducatelle, F., Gambardella, L.: AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks. European Transactions on Telecommunications, *Special Issue on Self Organization in Mobile Networking* **16**(5) (2005) 443–455
3. Ducatelle, F., Di Caro, G., Gambardella, L.: Using ant agents to combine reactive and proactive strategies for routing in mobile ad hoc networks. Int. Journal of Computational Intelligence and Applications (IJCIA), *Special Issue on Nature-Inspired Approaches to Networks and Telecommunications* **5**(2) (2005) 169–184
4. Di Caro, G., Ducatelle, F., Gambardella, L.: Swarm intelligence for routing in mobile ad hoc networks. In: Proceedings of the 2005 IEEE Swarm Intelligence Symposium (SIS). (2005)
5. Ducatelle, F., Di Caro, G., Gambardella, L.: Ant agents for hybrid multipath routing in mobile ad hoc networks. In: Proceedings of the Second Annual Conference on Wireless On demand Network Systems and Services (WONS), St. Moritz, Switzerland (2005)
6. Di Caro, G., Ducatelle, F., Gambardella, L.: AntHocNet: an ant-based hybrid routing algorithm for mobile ad hoc networks. In: Proceedings of Parallel Problem Solving from Nature (PPSN) VIII. Volume 3242 of Lecture Notes in Computer Science., Springer-Verlag (2004) 461–470 (Conference best paper award).
7. Di Caro, G.: Ant Colony Optimization and its application to adaptive routing in telecommunication networks. PhD thesis, Faculté des Sciences Appliquées, Université Libre de Bruxelles, Brussels, Belgium (2004)
8. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press (1998)
9. Di Caro, G., Dorigo, M.: AntNet: Distributed stigmergic control for communications networks. J. of Artificial Intelligence Research (JAIR) **9** (1998) 317–365
10. Shen, C.C., Jaikaeo, C., Srisathapornphat, C., Huang, Z., Rajagopalan, S.: Ad hoc networking with swarm intelligence. In: Proceedings of ANTS 2004, Fourth International Workshop on Ant Algorithms. LNCS, Springer-Verlag (2004)
11. Baras, J.S., Mehta, H.: A probabilistic emergent routing algorithm for mobile ad hoc networks. In: WiOpt03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks. (2003)
12. Günes, M., Kähmer, M., Bouazizi, I.: Ant-routing-algorithm (ARA) for mobile multi-hop ad-hoc networks - new features and results. In: Proceedings of the 2nd Mediterranean Workshop on Ad-Hoc Networks (Med-Hoc-Net'03), Mahdia, Tunisia (2003)
13. Di Caro, G., Ducatelle, F., Gambardella, L.: Studies of routing performance in a city-like testbed for mobile ad hoc networks. Technical Report 07-06, IDSIA, Lugano (Switzerland) (2006)
14. Perkins, C., Royer, E.: Ad-hoc on-demand distance vector routing. In: Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications. (1999)
15. Clausen, T., Jacquet, P., Laouiti, A., Muhlethaler, P., Qayyum, A., Viennot, L.: Optimized link state routing protocol. In: Proceedings of IEEE INMIC. (2001)
16. Bertsekas, D., Gallager, R.: Data Networks. Prentice–Hall (1992)
17. Scalable Network Technologies, Inc. Culver City, CA, USA: QualNet Simulator, Version 3.8. (2005) `http://www.scalable-networks.com`.
18. Johnson, D., Maltz, D.: Dynamic Source Routing in Ad Hoc Wireless Networks. In: Mobile Computing. Kluwer (1996) 153–181
19. Rappaport, T.Wireless communications, principles and practicePrenticeHall (1996)
20. De Couto, D., Aguayo, D., Chambers, B., Morris, R.: Performance of multihop wireless networks: Shortest path is not enough. In: Proceedings of the First Workshop on Hot Topics in Networks (HotNets-I), ACM SIGCOMM (2002)