Research Article

Hybrid ACO Routing Protocol for Mobile Ad Hoc Networks

D. Rupérez Cañas, 1 A. L. Sandoval Orozco, 1 L. J. García Villalba, 1 and P.-S. Hong 2

1 Group of Analysis, Security and Systems (GASS), Department of Software Engineering and Artificial Intelligence (DISIA), School of Computer Science, Office 431, Universidad Complutense de Madrid (UCM), Calle Profesor José García Santémases s/n, Ciudad Universitaria, 28040 Madrid, Spain
2 School of Information Technology, Sungshin Women’s University, 249-1 Dongseon-dong 3-ga, Seoul 136-742, Republic of Korea

Correspondence should be addressed to L. J. García Villalba; javiergv@fdi.ucm.es

Received 19 April 2013; Accepted 7 May 2013

Academic Editor: Sabah Mohammed

Copyright © 2013 D. Rupérez Cañas et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile ad hoc networks (MANETs) are formed by wireless mobile devices that do not require a predefined infrastructure and where all nodes are at the same level without the need of a central coordinator. The routing is an important task in this kind of networks because of highly dynamic environments and other characteristics. Routing consists of sending the information from sender nodes to destination nodes in a multihop manner. There are different techniques to solve the routing problem. We rely on bioinspired algorithms, those that take into account the behavior of certain animals to solve it. Specifically we design a novel routing algorithm based on Ant Colony Optimization (ACO) which uses the concepts of Swarm Intelligence and analyzes the behavior of ants at the time of obtaining the food. In this work, we present Hybrid ACO Routing (HACOR), a bio-inspired protocol based on ACO. In the simulation results, we observe how this protocol performs significantly better compared to a state-of-the-art routing protocol, according to the analyzed metrics.

1. Introduction

Nowadays the wireless networks are having a relevant importance due to the services offered by the high proliferation of the devices. Mobile ad hoc networks (MANETs) [1] are constituted by wireless mobile devices/nodes distributed without the need of predefined infrastructure; that is, each node has the same level in the network and they can act as client or server. The nodes may join or depart the network at any time, and thus wireless links are constantly created and destroyed.

The main problems of this kind of networks are autoconfiguration and routing. The autoconfiguration of mobile ad hoc networks requires specific protocols [2]; that is, traditional solutions are invalid. Similarly, routing in these networks requires particular solutions, like in wireless sensor networks [3]. Some protocols [4] combine autoconfiguration and routing aspects to optimize the operation of such networks.

Considering the limited communication range of wireless networks, information is usually transmitted in a multihop fashion. Data routing is an essential task in networking and it has the mission of finding routing paths from sender to destination nodes. In highly dynamic environments, such as MANETs, this task becomes particularly challenging, and routing protocols must be designed to cope with the time-varying topologies.

There are several routing approaches for MANETs. The so-called traditional algorithms (OLSR [5], AODV [6]), even though they are widely used for routing, usually do not meet the expectation in terms of performance. For this reason, the researchers have focused on natural behavior of some animals (the most of them are insects) to solve complex problems. These kinds of techniques are typically referred to as bioinspired algorithms.

Ant Colony Optimization (ACO) [7, 8] is a particular type of these algorithms which takes inspiration from the behavior of ants at the time of obtaining the food. ACO is a metaheuristic proposed by Dorigo in his thesis [9], and it has been used to solve different type of problems, including network data routing.
In this work we present Hybrid ACO Routing (HACOR) protocol. It is a novel ACO-based algorithm which combines proactive and reactive parts. HACOR proposes several enhancements and additional functionalities in comparison to previous ACO-based algorithms. Within these, a novel buffering technique for control packets, an improved link failure management procedure, and a proactive route exploration mechanism exist. These strategies were carefully designed for MANETs, aiming to reduce the protocol overhead and improve the network performance in terms of end-to-end delays and data throughput.

This paper is divided into 5 sections with the first section being this introduction. In Section 2, we discuss the most relevant hybrid related work in the routing based on ACO. In Section 3, we present our proposal, HACOR, explaining its major characteristics. In the following Section 4, we analyze the simulation results where HACOR and AODV are compared. Finally, we offer conclusions in Section 5.

2. Related Work

The ACO-based routing protocols can be classified, as well as the traditional protocols, in proactive, reactive, and hybrid. Proactive protocols frequently need to exchange packets between nodes and to continuously update their routing tables; therefore they usually suffer from increased overhead. On the other hand, the reactive protocols act on-demand; therefore they establish routing paths and sent control packets only when needed, thus reducing overhead. This reduction comes at the cost of higher latency.

Not until recently, the community started to explore the combination of both approaches, which led to the proposal of hybrid methods. The main motivation behind the hybridization different algorithms is getting the advantages of both approaches and finding a good trade-off between overhead and latency. We discuss some representative hybrid protocols proposed in the literature. Ant-AODV [10], a hybrid routing protocol based on ACO and on the routing protocol AODV, as its name suggests, tries to take advantage of both.

However, this protocol was designed without taking into account techniques to help to find the shortest routes and mechanisms to mitigate the congestion problem.

HOPNET [11] is based on a technique in which ants jump from one zone to another one. The algorithm has characteristics extracted from the ZRP and DSR protocols, being highly scalable, compared with other hybrid protocols. This algorithm consists of a proactive route setup in the area of node vicinity and communication between zones reactively on demand, which is done at the moment of sending packets from a zone to another. When the number of nodes is small, the continuous movement of peripheral nodes constantly triggers attempts to discover new routes, which causes more overhead and transmission delays compared to other hybrid routing protocols. Another protocol is presented in [12], which combines ideas about ACO with Zone-Based Hierarchical (ZHLS) protocol. Its algorithm is similar to HOPNET and it is based on ants which cross from one zone to the next one. The authors claim that their proposal improves the performance in comparison to traditional algorithms, according to the delay, ratio, and overhead metrics. But undoubtedly the most representative is AntHocNet [13, 14]. It constitutes a hybrid, adaptive, and multipath protocol that takes into account the dynamic topology and other characteristics of the MANETs, presenting a hybrid mode of operation; it is reactive because it has agents operating in the route setup to destinations and proactive due to other agents collecting information to discover new routes in the prevention against link failure. It is multipath because it establishes different paths to send the information to the destination. Finally, it is adaptive because it suits the current traffic and network conditions.

Finally, AntOR [15, 16] is a protocol based on AntHocNet but it differs from this in the following characteristics: (i) it is a protocol that works in two separate modes: Disjoint-link and Disjoint-node; (ii) it takes into account the pheromone separation in the diffusion process; (iii) it uses a distance metric in path exploration. In this protocol there are two kinds of routes: Disjoint-node and Disjoint-link. The first corresponds to routes in which nodes are not shared and the latter refers to routes in which links are not shared.

3. Proposed Algorithm

In this work, we present HACOR which is a hybrid bio-inspired protocol. Main characteristic is that HACOR belongs to hybrid algorithms because it is a combination between proactive and reactive parts.

(i) Reactive. It acts on-demand sending reactive agents or ants for routing setup process, when there are available data packets to be sent towards destination.

(ii) Proactive. In the path exploration process, the source node of the data session sends proactively, in time intervals, agents for creation of alternatives routes. This process only occurs when the communication between source and destination has succeeded during the reactive process.

Moreover, it operates in a multipath way because it establishes different paths to send the information to the destination. Also, it is adaptive because it suits the traffic and network conditions. To this end, the main functional features are as follows.

3.1. Control Packet Buffering. The motivation of this technique is the reduction of overhead. Thus, we send the control packets from the buffer synchronously. This strategy employs a buffer in which the control packets are temporarily stored. All the control packets, which are ready to send, are first stored in the control buffer. The main core of this technique consists on sending the enqueued packets to corresponding destinations in every time interval (it is established to 100 ms). The information included in each entry which is stored in this buffer is (a) socket to send the packet, (b) the control packet which is the particular message of the protocol, and (c) destination address (it could be a broadcast address or a unicast address sent to a determined node). We have utilized
time intervals of 100 ms because we got the best results in the simulations. With this technique, there is not the overlapping of control packets at the time of sending.

3.2. Outdated Route Management. This routing approach utilizes a new technique to control the outdated routes. It is a method that replaces pheromone evaporation. This technique avoids the conflicts of the routes, which may be created in an unnecessary manner. To achieve this goal, an event is triggered every 2 s. The technique is described as follows. A time-stamp field with the current time is established, every time a register in the routing table is updated or created. If the associated time-stamp field of each route from routing table is minor than current time minus a time limit (it is established to 5 s in the experiments), the particular register of the route is deleted from the routing table. This time limit can vary according to implementations. If it has low values, then the system converges slowly to premature routes, but with the drawback that the system may erase routes to active destinations. On the other hand, if it has high values, it implies a fast convergence in the creation of the routes but with disadvantage of keeping outdated routes in the system.

3.3. Data Packet Management. This technique consists of checking periodically the buffer for delayed data packets. Every node, at the time of forwarding the data packet, checks if it has a route to destination. If such route does not exist, it enqueues the corresponding packet. By periodically checking the buffer and resending packets, we obtain a better delivered data packet ratio, without altering other metrics, such as delay or jitter. The time period for checking the buffer is around 100 ms. Also, we want to note that the enqueued data packets are always sent while there is a valid route to its corresponding destination. We use this technique aiming to improve the packet delivery ratio, trying not to lose any packet. To avoid affecting other metrics as average end-to-end delay or jitter, we employ a small execution time as previously mentioned.

3.4. Link Failure Management. In this management, we control two processes. (a) The first one is related to fault tolerant, and (b) the second one is related to the way of neutralization the link failure management.

A common way to handle faults in control messages (an ant agent in ACO-based protocols) is to trigger a mechanism of neutralization process. This process involves the sending of agents to repair the route or to notify the precursors of the node about the failure. These agents are propagated (by broadcasting) throughout the network until reaching the source of the data session indicating that the route has become invalid. In highly dynamic environments, such neutralization processes are executed very often, inducing a large overhead in terms of packets and bytes. To fix this weakness, we employ a new technique that checks if a route exists (i.e., regular pheromone value greater than zero). If a path exists, we send the control message (unicast); otherwise the agent does not send.

Moreover, we try to neutralize the node/link failure as follows. The first event that occurs when there is a node failure is that the node that perceives the failure updates its neighbors table and deletes all routes which have the failure node as next hop. Then we proceed in this way. If there is no route at source node, a reactive forward ant is sent. If there is no route at an intermediate node and a data packet was being forwarded when the failure occurred, a route repair forward ant is sent to every destination of all affected data sessions. If there is no route at the intermediate node and a control packet (HELLO message is not consecutively received at every certain interval) was being forwarded, no neutralization message is sent. If there is no route at the intermediate node and a unicast control packet (different to HELLO) was being forwarded, we send a unicast message to precursor node. This process is repeated as often as needed until the source node is reached. Another functionality is the next. At the moment the data packet is ready to be forwarded to next hop, our algorithm checks if there is or not a valid route to destination of the current data session. In the case there is not valid route, our algorithm enqueues the current data packet and sends a local route repair ant to fix the problem. Also, in this method, the current node sends a unicast message to all reachable neighbors. The neighbors which receive this message sent the same message to its precursors if they have it. Otherwise, they stop sending.

The basic functionality is described in Algorithm 1.

3.5. Route Exploration Management. In our algorithm, we use proactively a method in the route exploration using
the concepts of Simple-ACO [8]. With this approach, we are able to reduce even more the protocol overhead. The modifications in Simple-ACO are the following.

(i) We do not use evaporation process.

(ii) We use free-loop method when the route with all visited nodes is created.

(iii) We do not need to set initial pheromone value toward every one-hop neighbor and we do not update pheromone value anymore. We utilize the message HELLO to this end. Every node that receives a HELLO message from another one-hop neighbor updates its route to such a neighbor with the new pheromone value.

(iv) We use the probabilistic method like Simple-ACO to send the proactive Ant.

(v) We have in consideration the Disjoint-Link route in this process. Thus all alternative routes, created in this process, are disjoint. The advantage of Disjoint-Link routes are (a) a failure in one node only affects a path, not the entire network, and (b) load balancing is better because there are not repeated routes on the disjoint property.

4. Simulation Results

We have performed several tests with the Network Simulator NS-3 [17]. The most important performance metrics that were used in the experimentation are

(i) **average end-to-end delay**: measures the accumulative effectiveness of experienced delays by packets going from source to destination;

(ii) **jitter**: performance metric that measures the delay variation between consecutive packets received;

(iii) **overhead in number of bytes**: relationship between the average of transmitted control bytes and the average of delivered data bytes;

(iv) **overhead in number of packets**: relationship between the average of transmitted control packets by the nodes of network and the average of delivered data packets to their destinations;

(v) **delivered data packet ratio**: the fraction of correctly delivered data packets versus sent;

(vi) **throughput**: volume of work or information flowing through a system. It is calculated by dividing the total number of bits delivered to the destination by the packet delivery time.

We compare our proposal with purely reactive protocol AODV. For the evaluation, we use two kinds of experiments.

4.1. **Experiment A.** We consider 100 nodes, randomly distributed in the area, with transmission range of 300 m. The nodes are moved according to the Random Way Point (RWP) pattern. We vary the pause time from 0 s to 240 s using a time interval of 60 s. The scenario was rectangular with dimensions 3000 m × 1000 m. The speed was set to constant with value of 5 m/s. Each simulation run uses 10 random data sessions using the application protocol Constant Bit Rate (CBR) beginning to send data at random from 0 s to a maximum of 180 s. The sending rate is 512 bit/s, that is, sending a packet of 64 bytes per second. The maximum simulation time is established to 900 s. Each test consisted of 10 runs.

Figure 1 shows how our proposal HACOR has a better performance in terms of delay than AODV at all pause times considered. We can note that in HACOR the delay never reaches the value of 100 ms. Increasing the pause time has two different effects on the general properties of the scenario that are relevant for routing. Firstly, there is a decrease in node mobility; since nodes stay still for longer periods, they are less mobile, and the network becomes less dynamic. As a consequence, the scenario becomes less difficult. Secondly, it is related to the distribution of nodes over the network area when the RWP mobility model is used.
Figure 2 shows that the curve which represents the jitter in HACOR has a behavior more uniform than the curve in AODV. Also we can appreciate how the jitter is never more than 110 ms in our approach.

With regard to overhead in bytes, in Figure 3, we appreciate how this property in HACOR is lower than in AODV at all pause times. Also we show that the curve of HACOR does not have extreme behavior and it is practically a straight line.

Figure 4 shows a slightly similar behavior in comparison to Figure 3. In this plot, we can appreciate how the two curves are less steep. We can also observe that this overhead in number of packets is less in HACOR than in AODV.

According to Figure 5, the ratio is very distinguishing performance metric, and we observe how this metric is much higher in HACOR than in AODV at all pause times. We also check that the ratio is never less than 86%. Thus, we can affirm that we get very good results with our proposal.

Figure 6 has a behavior similar to Figure 5, but this plot uses another scale of values.

4.2. Experiment B. This experiment was designed to assess the scalability of our proposal. We vary the number of nodes in the network from a minimum of 50 nodes up to maximum of 150 nodes with a constant node density. To this end, we changed the scenario dimensions as follows: 750 m × 750 m, 875 m × 875 m, 1000 m × 1000 m, 1125 m × 1125 m, and 1250 m × 1250 m for number of nodes 50, 75, 100, 125, and 150, respectively. These nodes were randomly distributed with a transmission range of 300 m and are also moved according to the Random Way Point (RWP) pattern, which has a pause time of 2 s and speed of 5 m/s. It also uses 10 random data sessions using the application protocol Constant Bit Rate (CBR) beginning to send data at random from 0 s to a maximum of 180 s. The sending rate was 2048 bit/s, that is, sending 4 packets of 64 bytes per second. The maximum simulation time was established to 900 s. It employs a total of 10 runs in the experiment.

Figure 7 shows that delay is better in HACOR than AODV according to this complex scenario. We can observe that both curves are increasing but the curve of AODV is more accentuated than the curve of HACOR.
Figure 8 shows how the jitter in HACOR is better than AODV, and we can appreciate the curve in HACOR having a uniform behavior than AODVs. In AODV the curve is more irregular.

Figure 9 shows the overhead in number of bytes. We appreciate that it is very similar in both approaches.

The overhead in number of packets is slightly higher in HACOR than in AODV, but in dense networks around 150 nodes (according to Figure 10) the overhead is the same. We can check a good performance with regard to this overhead in number of packets with no dense networks.

Figures 11 and 12 show the ratio and throughput, respectively. In both plots, we can appreciate the scalability of the two protocols. In Figure 11, we observe how the ratio decreases strongly in AODV than in our approach. In fact, it implies that in HACOR we obtain a better delivery ratio with the increment of nodes. Figure 12 shows the throughput but using another scale. We can appreciate a similar behavior than Figure 11.

5. Conclusion

Mobile ad hoc networks are formed by wireless devices without a predefined infrastructure. Due to highly dynamic environments, forwarding the data from a source to a destination is a critical and difficult task. This data routing can be treated using several techniques, but we rely on the bioinspired algorithms to solve this complex problem. Ant Colony Optimization (ACO) algorithms are our inspiration. ACO takes into account the behavior of ants at the time of obtaining the food. This behavior is based on Swarm Intelligence, that is, considers the collective behavior of some animals, in our case ants. In this paper, we present HACOR, a hybrid ACO-based routing protocol which has novel characteristics and employs enhanced techniques compared to its predecessors. The experimentation results show that HACOR has a better performance than AODV according to analyzed metrics. Overall, we can appreciate a significant improvement with regard to overhead in number of packets.
Figure 11: Delivered data packet ratio (Experiment B).

Figure 12: Throughput (Experiment B).

Acknowledgment

This work was supported by the Agencia Española de Cooperación Internacional para el Desarrollo (AECID, Spain) through Acción Integrada MAEC-AECID MEDITERRÁE-NEO AI/037528/11.

References

[1] R. Ramanathan and J. Redi, “A brief overview of ad hoc networks: challenges and directions,” Communications Magazine, IEEE, vol. 40, no. 5, pp. 20–22, 2002.
[2] L. J. G. Villalba, J. G. Matesanz, A. L. S. Orozco, and J. D. M. Díaz, “Auto-configuration protocols in mobile ad hoc networks,” Sensors, vol. 11, no. 4, pp. 3652–3666, 2011.
[3] L. J. G. Villalba, A. L. S. Orozco, A. T. Cabrera, and C. J. B. Abbas, “Routing protocols in wireless sensor networks,” Sensors, vol. 9, no. 11, pp. 8399–8421, 2009.
[4] L. J. G. Villalba, J. G. Matesanz, A. L. S. Orozco, and J. D. M. Diaz, “Distributed dynamic host configuration protocol (D2HCP),” Sensors, vol. 11, no. 4, pp. 4438–4461, 2011.
[5] T. Clausen and P. Jacquet, Optimized Link State Routing Protocol, vol. 3626, RFC, 2003.
[6] C. E. Perkins and E. M. Royer, “Ad-hoc on-demand distance vector routing,” in Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA ’99), pp. 90–100, February 1999.
[7] M. Dorigo, “Ant colony optimization: a new meta-heuristic,” in Proceedings of the IEEE Congress on Evolutionary Computation (CEC ’99), pp. 1470–1477, IEEE Press, 1999.
[8] G. A. Di Caro, Ant colony optimization and its application to adaptive routing in telecommunication networks [Ph.D. thesis], Faculté des Sciences Appliquées, Université Libre de Bruxelles, Brussels, Belgium, 2004.
[9] M. Dorigo, Optimization, learning and natural algorithms [Ph.D. thesis], Politecnico di Milano, Italy, 1992.
[10] X. Zheng, W. Guo, and R. Liu, “An ant-based distributed routing algorithm for ad-hoc networks,” in Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS ’04), vol. 1, pp. 412–417, June 2004.
[11] J. Wang, E. Osagie, P. Thulasiraman, and R. K. Thulasiram, “HOPNET: a hybrid ant colony optimization routing algorithm for mobile ad hoc network,” Ad Hoc Networks, vol. 7, no. 4, pp. 690–705, 2009.
[12] M. K. Rafsanjani, S. Asadinia, and F. Pakzad, “A hybrid routing algorithm based on ant colony and ZHLS routing protocol for MANET,” Communications in Computer and Information Science, vol. 120, no. 2, pp. 112–122, 2010.
[13] G. A. Di Caro, F. Ducatelle, and L. M. Gambardella, “AntHocNet: an ant-based hybrid routing algorithm for mobile ad hoc networks,” in Proceedings of the of Parallel Problem Solving from Nature (PPSN’04), vol. 3242, pp. 461–470, Springer, 2004, Lecture Notes in Computer Science.
[14] F. Ducatelle, Adaptive routing in ad hoc wireless multi-hop networks [Ph.D. thesis], Universita della Svizzera Italiana, Istituto Dalle Molle di Studi sull’Intelligenza, Lugano, Switzerland, 2007.
[15] L. J. G. Villalba, D. R. Cañas, and A. L. S. Orozco, “Bio-inspired routing protocol for mobile ad hoc networks,” IET Communications, vol. 4, no. 18, pp. 2187–2195, 2010.
[16] D. R. Cañas, A. L. S. Orozco, L. J. G. Villalba, and T. H. Kim, “A comparison study between antor-disjoint node routing and antor-disjoint link routing for mobile ad hoc networks,” in Proceedings of the FGIT-MulGraB, pp. 300–304, 2011.
[17] G. F. Riley and T. R. Henderson, The Ns-3 Network Simulator, 2010.