2010 17th International Conference on Telecommunications

Ant-DYMO: A Bio-Inspired Algorithm for MANETS

José Alex Pontes Martins
Universidade Vale do Acaraú (UVA)
Sobral, Brazil 62.040–370
Email: alexmartins@larces.uece.br

Sergio Luis O. B. Correia
Computer Science Department
State University of Ceara (UECE)
Fortaleza, Brazil 60.740–000
Email: sergio@larces.uece.br

Joaquim Celestino Júnior
Graduate Program in Computer Science
State University of Ceara (UECE)
Fortaleza, Brazil 60.740–000
Email: celestino@larces.uece.br

***Abstract*—Mobile ad hoc networks are a set of wireless mobile devices that communicate without fixed infrastructure, forming temporary networks dynamically. Each node in such a network is more than a data receiver/sender, it is also a router that forwards data packets to its proper destination. The main characteristics of ad hoc networks are frequent change in the network topology, limited power of its links and restriction on the bandwidth. A routing protocol for ad hoc networks is composed of a routing algorithm with a set of rules that monitor the operation of the network. Thus, the nodes participating in the network have an important role in the management of resources in ad hoc networks. Ant-based routing is an efficient routing scheme based on the behavior of foraging ants. The study of the collective behavior of ants shows that they are able to find the shortest path from the nest to a food source, using a particular mode of communication by the means of a chemical substance called pheromones. This work uses a mechanism from collective intelligence applied to ad hoc networks, in particular the application of ants for routing in ad hoc networks. We have brought some characteristics from the Dynamic MANET On-demand Routing protocol and other MANET protocols in order to propose the new routing algorithm, called Ant-DYMO. We compare Ant-DYMO with DYMO, and we show that our proposition has improved the packet loss and the end-to-end delay.**

I. INTRODUCTION

A commonly used way to differentiate routing protocols for mobile ad hoc networks (MANETs) is based on how the routing information is acquired and maintained by the mobile nodes [1]. This classification system divides the routing protocols into three categories: proactive (DSDV [2]), reactive (AODV [3], DSR [4], DYMO [5]) and hybrid (ZRP [6]) protocols.

The former two groups – proactive and reactive – have clear limitations. Proactive protocols maintain updated information on the whole network topology. If on the one hand routes are always available, on the other hand the costs to make it so can be too high, as a lot of control traffic is required, hence reducing the effective network capacity for data communication. As opposed to proactive protocols, the reactive ones do not suffer with the control traffic problem, as they work on-demand, maintaining only needed routes. When a route for an unknown destination is needed, a route discovery mechanism is then started. The problem with this approach is that the route request process might take a long time, making this kind of protocol not suitable for certain existing applications, such as real-time multimedia transmissions.

Mobile agents making use of the ant colony optimization metaheuristic (ACO) [7] have been shown to be both suitable for and efficient in the task of mapping the network topology, as well as improving its connectivity. The present paper proposes the application of the ACO technique to the Dynamic MANET On-demand Routing (DYMO) protocol, in order to create the Ant-DYMO.

The remainder of this paper is divided as follows: section II presents the related work; section III gives a brief description of the DYMO protocol; section IV explains our Ant-DYMO proposal; section V details the simulation environment, such as scenario, metrics and the Ant-DYMO settings, as well as presents and discusses the obtained simulation results; and finally, section VI concludes this paper, giving insights into the envisioned future works with Ant-DYMO.

II. RELATED WORK

Ant-Colony Based Routing Algorithm (ARA) [8] is a reactive MANET routing protocol that uses the ACO metaheuristic [9]. ARA's artificial pheromone model is probabilistic and based on the number of hops. Through a search procedure, a *forward ant* (FANT) will create a pheromone trail from the source to the destination. A *backward ant* (BANT) does the reverse path, i.e. after a FANT reaches its destination, it is destroyed and a BANT is created, is sent to the source node, and finds its way to the destination by the means of a procedure similar to that of the FANT. It then creates the pheromone trail, or if it already exists, it increases the pheromone level on the path. The data packets also enforce the pheromone level on the path they are passing through.

Ant On-Demand Distance Vector Routing (Ant-AODV) [10] is a hybrid routing algorithm for MANETs based upon AODV, not changing any of its native characteristics. AODV does the reactive part and an ant-based approach does the proactive one. The main goal of the ant algorithm here is to continuously create routes in the attempt to reduce the end-to-end delay and the network latency, increasing the probability of finding routes more quickly, when required. Ant-AODV's artificial pheromone model is based on the number of hops and its

978-1-4244-5247-7/09/$26.00 ©2009 IEEE

goal is to discover the network topology, without any other specific functions, as opposed to most ACO algorithms [7].

Ant Dynamic Source Routing (Ant-DSR) [11] is a reactive protocol that implements a proactive route optimization method through the constant verification of cached routes. This approach increases the probability of a given cached route express the network reality.

III. THE DYMO PROTOCOL

Dynamic MANET On-demand Routing (DYMO) [5] is a reactive routing protocol under development by the Mobile Ad hoc Networks Working Group from IETF and is intended for use by mobile routers in wireless, multi-hop networks. It offers adaption for topology changes and determines unicast routes on-demand. Its main activities are the route discovery and route maintenance mechanisms, achieved through its *route requisition* (RREQ), *route reply* (RREP) and *route error* (RRER) messages. It employs sequence numbers to ensure loop freedom [12] and also to avoid the dissemination of ancient routing information, and its basic operation consists of sending RREQ messages through the network for finding the routes it needs. The intermediate nodes receiving a RREQ message store the route for the node that originated it and then reforward the RREQ. The destination node, upon receiving a RREQ message, replies with a unicast RREP one. The same way, every intermediate node receiving a RREP message stores the route for the node that originated it. To adapt to topology changes, after receiving a packet that should be sent to a link that is no longer available, the node notifies the message sender by sending back a RERR message. IF the source node still wants to send packets to given destination, a new route discovery process is to be initiated. The DYMO protocol works with source routing, meaning nodes read the routing messages to acquire knowledge on the paths involved in the search process, as well as write in the search packet the necessary hops needed to reach its destination. This methods clearly increases the size of the routing packets, with the intention of reducing the number of retransmissions.

IV. ANT-DYMO PROPOSAL

Ant-DYMO is a hybrid protocol that uses an ant-based approach in its proactive phase while DYMO [5] is the basis for the reactive one. That way it was possible to improve the latency and increase the network connectivity.

A. Definitions

Definition 1. Ant-DYMO defines two types of artificial ants: *explorer ant* (EANT), responsible for creating routes to its source and *search ant* (ARREQ), responsible for searching for a specific destination. Similar to the BANTs of other algorithms [8], [10], the EANTs carry the information on the destination node and create (or enforce) pheromone trails along the way. The EANTs carry the address of the source node and also a list containing every intermediate node it has passed by. Algorithm 1 describes in a simplistic way how the EANTs are processed.

Algorithm 1: Simplified processing of a received EANT

```
if EANT TTL $> 0$ then
    • create/update entry for EANT's last hop in node's pheromone table
    • add node address to EANT's list of hops
    • broadcast EANT to neighbors
else
    • discard EANT
end
```

ARREQ has characteristics from other algorithms' FANTs and RREQs [5], [8]. Its main goal is to search for a specific destination, and it inherits the format of DYMO's RREQ, adding a probabilistic search mechanism that takes into account the level of pheromones on the paths.

Definition 2. The transition probability of the ant located at the node i to the node j as the next hop is defined by Equation (1).

$$p(i,j) = \begin{cases} \dfrac{\tau(i,j)}{\sum\limits_{s \in \mathcal{N}_i} \tau(i,s)} & if \ s \in \mathcal{N}_i \\ 0 & otherwise \end{cases} \tag{1}$$

In Equation (1), $\mathcal{N}_i$ is the set of 1-hop neighbors of i and $\tau(i,j)$ is the pheromone level on the link $e(i,j)$. The transition probability $p(i,j)$ is part of Equation (2).

$$\sum_{j \in \mathcal{N}_i} p(i,j) = 1 \tag{2}$$

Definition 3. The pheromone update rules. When an EANT pass by a link $e(i,j)$, the pheromone level on the link $\tau(i,j)$ is updated according to Equations (3) and (4).

$$\tau(i,j) = \tau(i,j) + \Delta\tau(i,j) \tag{3}$$

$$\Delta\tau(i,j) = \frac{\alpha}{Distance(i,j)} \tag{4}$$

$$\tau(i,j) = (1-q) \cdot \tau(i,j) \tag{5}$$

In Equation (3), $\Delta\tau(i,j)$ is the increment to the pheromone level on the link from i to j, and it is calculated according to Equation (4).

In Equation (4), α is an adjustable parameter that represents the influence of the distance – in number of hops – from i to j over the pheromone level.

Equation (5) models the pheromone evaporation scheme, similar to the real ants, where the pheromone level reduces with time.

Ant-DYMO initializes parameter $\alpha = 1$. q is configured through the $evaporation\_factor\_$ parameter, described in section V-C.

B. Basic protocol design

Ant-DYMO is a hybrid and multi-hop algorithm. Nodes acquire information on their neighborhood by the limited flooding of Hello messages, as defined in [5]. Based on the received responses, each node creates its routing probability table [13], [14], similar to ACO's pheromone table, which replaces the traditional node routing table.

C. Routes exploration

After the beginning of the simulations, each node containing an EANT will broadcast it to its neighbors. A node receiving an EANT for the first time will create an entry in its routing table. Here, the essential information is mostly taken from the received EANT, and are:

- destination – the node that generated the EANT;
- next hop – the last node visited by the EANT, taken from its list of hops;
- pheromone level – the amount of pheromone over the link $e(current\ node,\ destination)$, calculated according to Equations (3) and (4).

After adding the entry, the EANT is then broadcasted to the node's neighbors. This process is illustrated in Algorithm 1, and the EANT's cycle lasts during the whole simulation time.

D. Route discovery

When a node $\mathcal{S}$ wants to send packets to a destination $\mathcal{D}$ not present in its routing table, it creates an ARREQ with its address and broadcasts it to its neighbors, and the scheme described in Figure 1 will be followed.

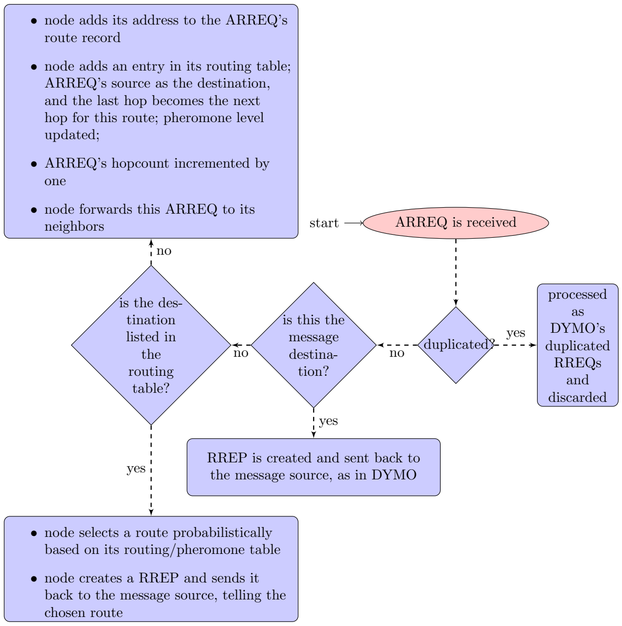

Fig. 1. ARREQ processing

E. Route maintenance

The route maintenance phase is responsible for improving the quality of the existing routes. Some algorithms generate specific packets for this task, but Ant-DYMO needs no special packets for this, and the reasons are two:

- the EANTs keep providing routes all the time, increasing the probabilities of quickly finding an alternate path in case of route errors;
- the data packets – mimicking the behavior of real ants – will enforce the pheromone trail of the selected path. This mechanism is similar to the one in [8].

The other maintenance procedures are natives of DYMO.

TABLE I
SIMULATION CONFIGURATION PARAMETERS

	Scenario	
Parameter	(a)	(b)
MAC	802.11	802.11
Mobility model	R. Waypoint	R. Waypoint
Simulation time	600s	600s
Transport protocol	UDP	UDP
Traffic application	CBR	CBR
Scenario dimension	1000m x 1000m	1000m x 1000m
Number of nodes	50	20
Number of connections	10	5
Pause time	[0, 600]	[0, 600]
Speed	10m/s, 20m/s	10m/s, 20m/s
Packet size	512 bytes	512 bytes
Transmission interval	0,25s	0,25s

V. SIMULATION AND RESULTS

A. Simulation Environment

The simulation environment chosen for evaluating the DYMO and Ant-DYMO protocols was the ns-2 network simulator [15] in its version 2.34, and the DYMO implementation used as basis for implementing Ant-DYMO was DYMOUM [16].

B. Scenario and metrics

Ant-DYMO and DYMO were evaluated in two scenarios and had their performance compared with respect to four metrics: end-to-end delay, delivery rate, loss rate and routing overhead. Table I shows the settings for the two proposed scenarios, and the results are discussed in section V-E.

Each simulation was repeated 10 times and a confidence interval of 95% was used.

C. Ant-DYMO configurable parameters

1) eants_percentage_: This parameter defines the percentage of ants to be inserted into the network, taking into account the number of nodes in the simulation. The ants are then placed into random nodes – selected by a uniform random number generator. For each simulation run we have the ants placed in different nodes.

The simulations carried out showed that a number of ants bigger than 30% increases considerably the overhead without improving the descovery of destinations at the same rate.

TABLE II
ANT-DYMO SETTINGS

Parameter	Value
eants_percentage_	0.30 (30%)
eants_history_	12 hops
evaporation_factor_	0.5
eants_route_expiration_time_	$10s$
eant_interval_	$1s$

2) eants_history_: This parameter defines the number of nodes an ant can visit and store into its list of hops – working pretty much like the TTL of a packet – before being discarded.

It allows control of the quality of the transmitted information, mainly regarding topology changes, as very long paths have a bigger probability of containing invalid routes.

3) evaporation_factor_: This parameter is responsible for controlling the simulation of the evaporation process of real ants. See parameter q in Equation (5).

4) eants_route_expiration_time_: This parameter indicates how long an inactive route discovered by an EANT takes to expire.

5) eant_interval_: This is the periodic interval for the evaporation mechanism to be performed.

D. Parameters settings

For the simulations, the Ant-DYMO protocol was configured as shown in Table II.

E. Simulation Results

1) End-to-end delay: End-to-end delay is related to the average time it takes from when the packet is in the router buffer waiting for a route until its delivery to the destination. That way, it comprises every network delay, such as latency, buffer delay and retransmissions.

According to Figure 2, Figure 3, Figure 4 and Figure 5, the Ant-DYMO protocol takes less time, in average, to deliver its packets.

Figure 2 shows a gain of about 16.5% of Ant-DYMO over DYMO, with a max gain of 35% in the pause time 100s. In Figure 3, the average gain was of about 10%, with max value of 17% in the pause time of 600s. Most of the individual gains happened in the first 400 seconds of pause, which shows that the Ant-DYMO protocol performs better than DYMO in networks with more mobility. This result was expected, as the EANTs are working on mapping the network topology before routes are requested, which increases the probability of finding needed routes in less time.

2) Delivery rate: This is one of the most important metrics while evaluating a routing protocol, as it shows how it performs in terms of effective data delivery.

Figure 6, Figure 7, Figure 8 and Figure 9 show that the Ant-DYMO protocol performs better than DYMO at delivering data packets in the proposed scenarios, mainly in scenarios with up to 400 seconds of pause. The biggest gains of Ant-DYMO over DYMO are of 11%, 8%, 13.4% and 12.3%, and happen respectively at Figure 6 with $0s$ of pause time, Figure 7 with $200s$ of pause time, Figure 8 with $100s$ of pause time and finally Figure 9 with $100s$ of pause time.

3) Loss rate: The loss rate describes the sum of all packets discarded for any reason, such as error, collision and exceeded time, for instance.

The extra overhead generated by the Ant-DYMO control traffic can be clearly observed in Figure 10, Figure 11, Figure 12 and Figure 13.

When the scenario has less nodes, as in Figures 10 and 11, the loss rate in Ant-DYMO is nearly the same observed with DYMO. However, when the number of nodes is larger, as in Figure 12 and Figure 13, the loss rate increases in most cases, situation caused by an also larger number of ants and the extra overhead they add to the network. The loss rates of Ant-DYMO in relation to DYMO are in average $18,6\%$, $9,2\%$, $4,5\%$ and $17,45\%$, observed respectively in Figure 10, Figure 11, Figure 12 and Figure 13.

4) Routing overhead: This is the most vulnerable point of the Ant-DYMO protocol, as can be observed in Figure 14, Figure 15, Figure 16 and Figure 17. Two are the main reasons for this increased overhead, the extra traffic generated directly by the EANTS and the retransmissions created indirectly by the caching mechanism that may provide outdated or inexisting paths, due to the changes in the network topology.

Ant-DYMO had in average a routing overhead of about 15% greater than DYMO.

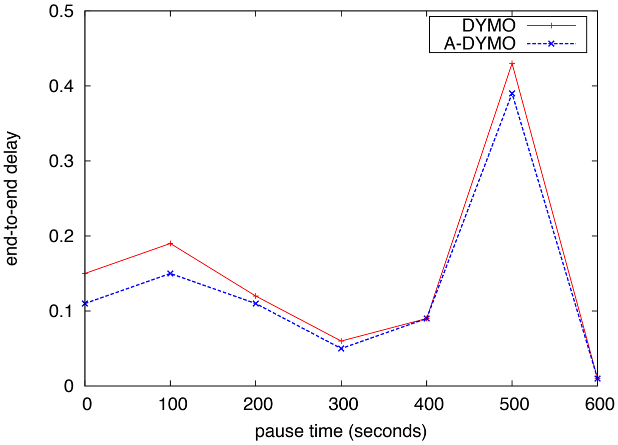

Fig. 2. Delay with 20 nodes x 5 connections x 10m/s

VI. CONCLUSION AND FUTURE WORK

A. Final Analysis

The figures from section V confirmed the following results, already expected by our study:

- Ant-DYMO had a better delivery rate, in average;
- it also delivered the data in less time;
- extra overhead was expected in the Ant-DYMO protocol and its effects were also confirmed in both the delivery rate and the traffic flow generated – above DYMO's but still less than a purely proactive protocol, as the number of ants making the proactive phase is somewhat controlled.

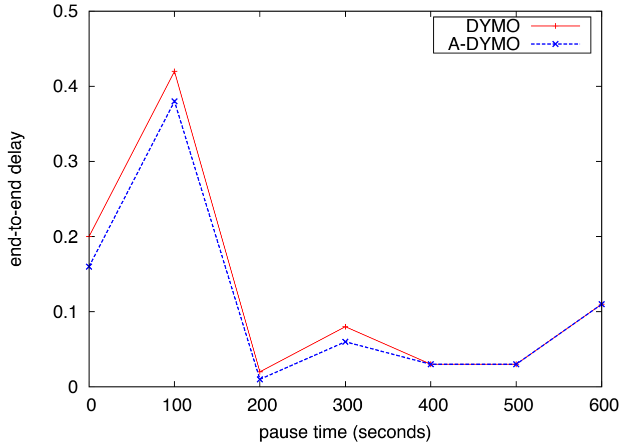

Fig. 3. Delay with 20 nodes x 5 connections x 20m/s

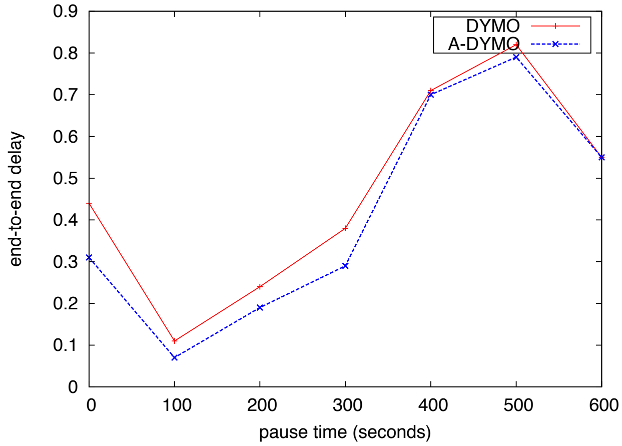

Fig. 4. Delay with 50 nodes x 15 connections x 10m/s

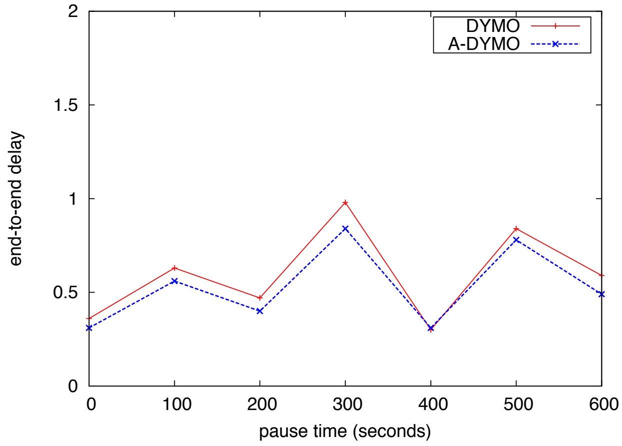

Fig. 5. Delay with 50 nodes x 15 connections x 20m/s

In overall, the Ant-DYMO protocol has been shown to be superior to DYMO regarding the effective packet delivery in a smaller amount of time. It was also shown that it is possible to directly influence its performance by tuning its configuration parameters.

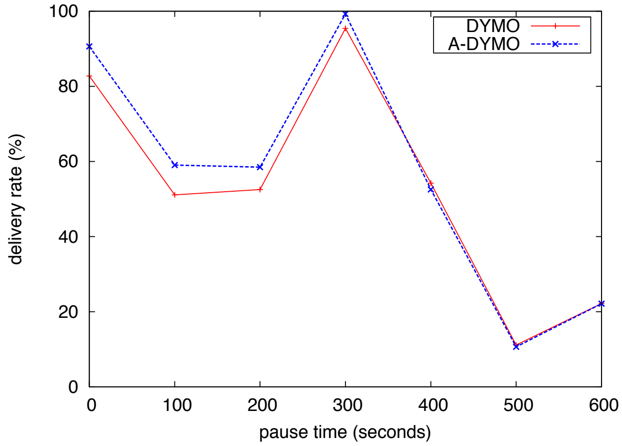

Fig. 6. Delivery with 20 nodes x 5 connections x 10m/s

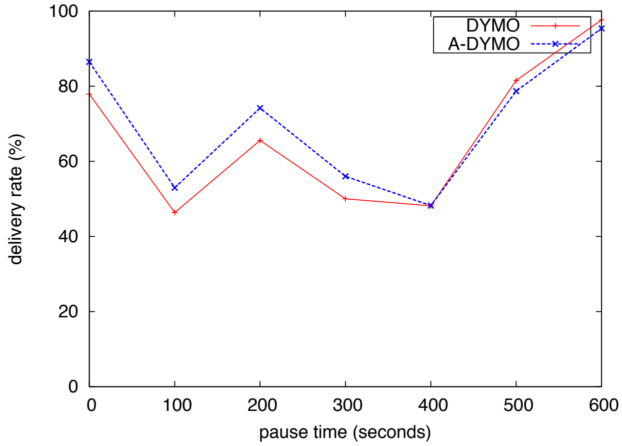

Fig. 7. Delivery with 20 nodes x 5 connections x 10m/s

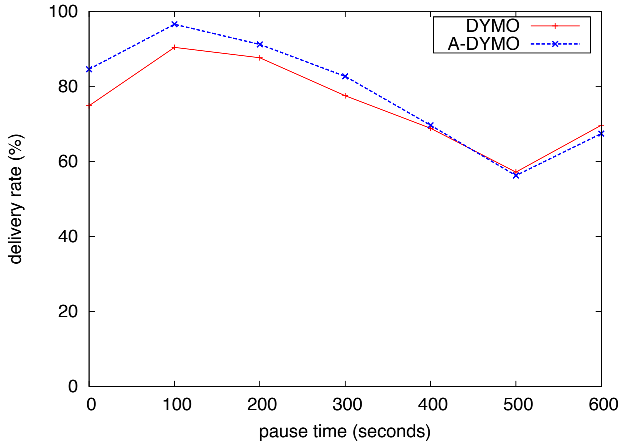

Fig. 8. Delivery with 50 nodes x 15 connections x 20m/s

B. Future work

The design and implementation of the Ant-DYMO protocol had the distance – measured in number of hops – as the only quality metric. The authors envision to improve their current Ant-DYMO proposal by making it work in a multiobjective way, adding also other metrics to work together, such as link quality – that could be measured by the power prediction –,

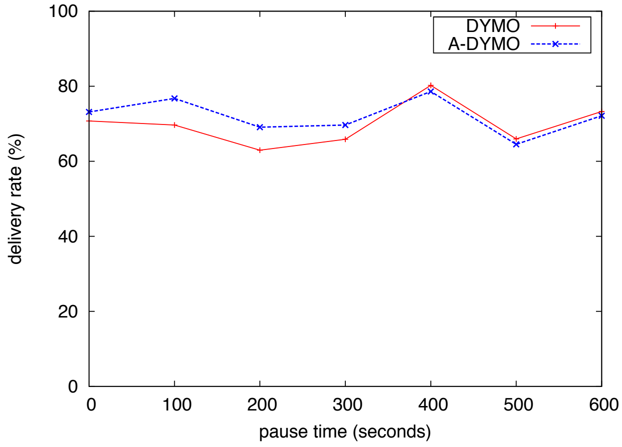

Fig. 9. Delivery with 50 nodes x 15 connections x 20m/s

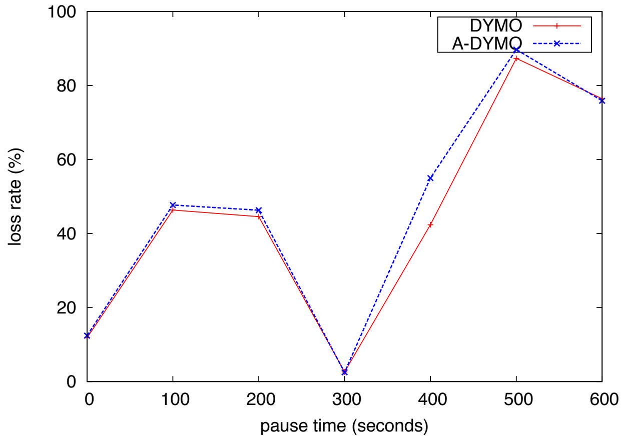

Fig. 10. Loss with 20 nodes x 5 connections x 10m/s

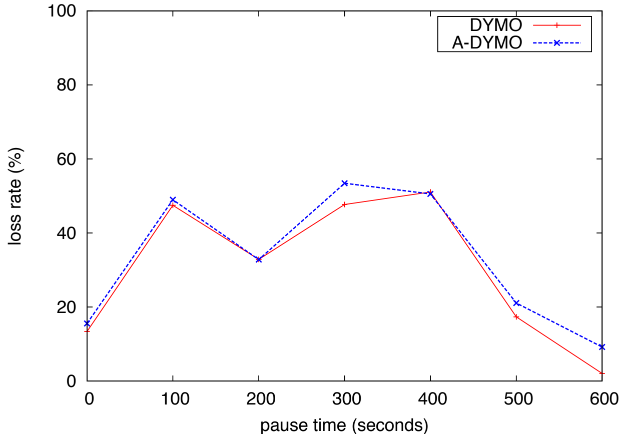

Fig. 11. Loss with 20 nodes x 5 connections x 10m/s

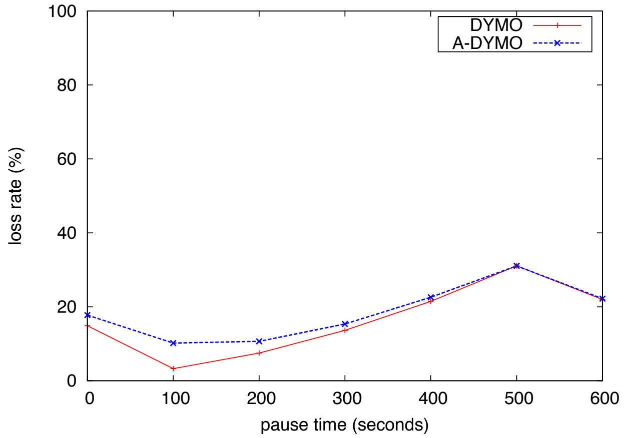

Fig. 12. Loss with 50 nodes x 15 connections x 20m/s

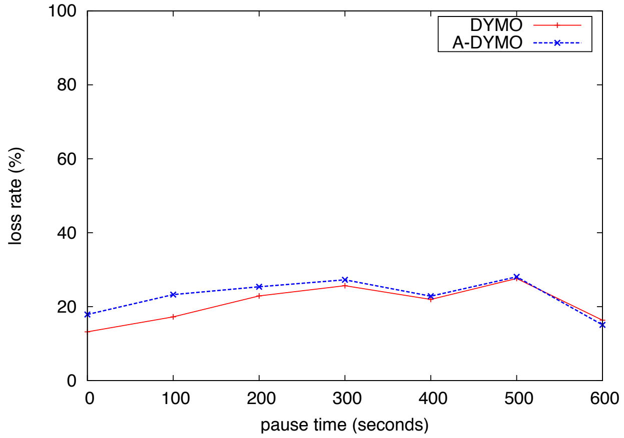

Fig. 13. Loss with 50 nodes x 15 connections x 20m/s

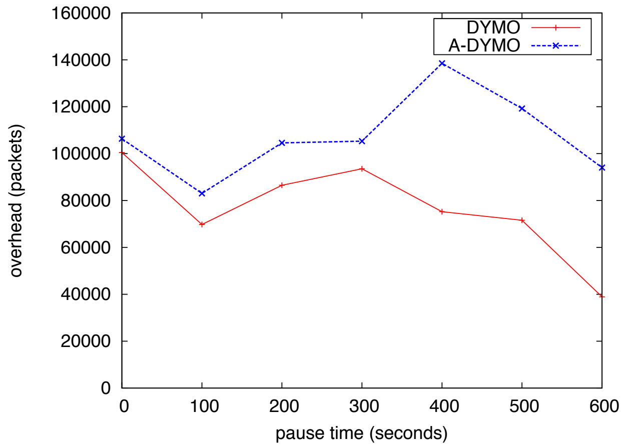

Fig. 14. Routing overhead with 20 nodes x 5 connections x 10m/s

buffer size or congestion prediction, as an example.

REFERENCES

[1] C. Liu and J. Kaiser, “A survey of mobile ad hoc network routing protocols,” *University of Magdeburg*, 2005.

[2] C. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers,” *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, pp. 234–244, 1994.

[3] C. Perkins, E. Belding-Royer, and S. Das, “RFC3561: ad hoc on-demand distance vector (AODV) routing,” *Internet RFCs*, 2003.

[4] D. Johnson, Y. Hu, and D. Maltz, “RFC4728: The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4,” 2007.

[5] I. Chakeres and C. Perkins, “Dynamic MANET On-demand (DYMO) Routing draft-ietf-manet-dymo-17,” Dynamic MANET On-demand (DYMO) Routing draft-ietf-manet-dymo-17, Internet Engineering Task Force, Mar. 2009. [Online]. Available: http://tools.ietf.org/html/draft-ietf-manet-dymo-17

[6] Z. Haas, M. Pearlman, and P. Samar, “The zone routing protocol (ZRP)

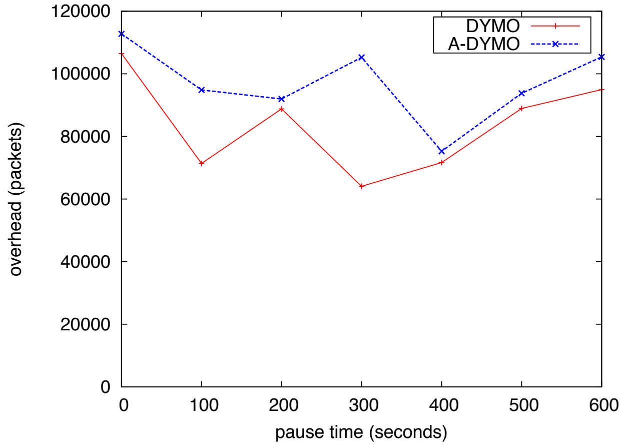

Fig. 15. Routing overhead with 20 nodes x 5 connections x 10m/s

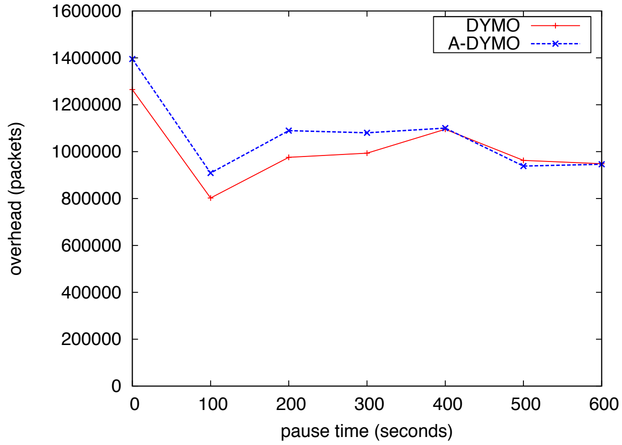

Fig. 16. Routing overhead with 50 nodes x 15 connections x 20m/s

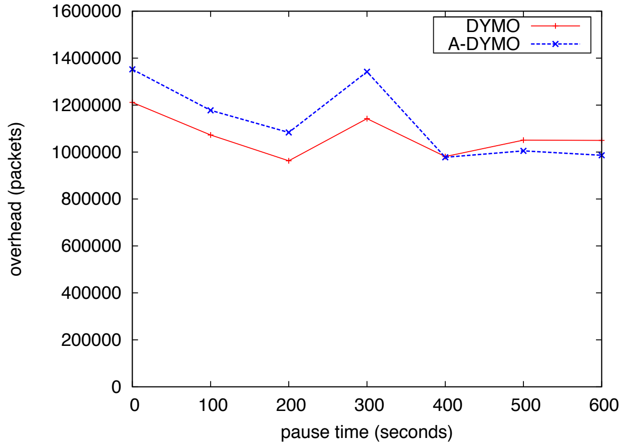

Fig. 17. Routing overhead with 50 nodes x 15 connections x 20m/s

for ad hoc networks," *draft-ietf-manet-zone-zrp-04.txt*, 2002.

[7] M. Dorigo and T. Stützle, *Ant Colony Optimization (Bradford Books)*. The MIT Press, July 2004.

[8] M. Gunes, U. Sorges, and I. Bouazizi, "ARA-the ant-colony based routing algorithm for MANETs," in *proceedings of the ICPP international Workshop on ad hoc networks (IWAHN)*. Citeseer, 2002.

[9] C. Blum and D. Merkle, *Swarm intelligence: introduction and applications*. Springer-Verlag New York Inc, 2008.

[10] S. Marwaha, C. Khong, and T. Srinivasan, "Ant-Aodv - Mobile agents based routing protocol for mobile ad hoc networks," *In in Proceedings of IEEE Globecom*, 2002.

[11] M. Aissani, M. Fenouche, H. Sadour, and A. Mellouk, "Ant-DSR: cache maintenance based routing protocol for mobile ad-hoc networks," in *Telecommunications, 2007. AICT 2007. The Third Advanced International Conference on*, 2007, pp. 35–35.

[12] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, vol. 2. Citeseer, 1999, pp. 90–100.

[13] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*. Oxford University Press, USA, 1999.

[14] X. Yang, L. Layuan, and C. Chuanhui, "Application Research Based Ant Colony Optimization for MANET," in *Wireless Communications, Networking and Mobile Computing, 2006. WiCOM 2006. International Conference on*, 2006, pp. 1–4.

[15] N. Simulator, "2 (NS2)," *URL: http://www. isi. edu/nsnam*, 2005.

[16] P. Ruiz and F. Ros, "DYMOUM–A DYMO implementation for real world and simulation," *URL http://masimum. dif. um. es*, 2005.