Available online at www.sciencedirect.com
ScienceDirect
Ad Hoc Networks 6 (2008) 827–859
Ad Hoc Networks
www.elsevier.com/locate/adhoc

On ant routing algorithms in ad hoc networks with critical connectivity

Laura Rosati [a,*], Matteo Berioli [b], Gianluca Reali [a]

[a] *University of Perugia, Department of Electronic and Information Engineering, via G. Duranti 93, 06125 Perugia, Italy*
[b] *German Aerospace Center (DLR), Institute of Communications and Navigation, P.O. Box 1116, Oberpfaffenhofen, Germany*

Received 31 August 2006; received in revised form 23 May 2007; accepted 16 July 2007
Available online 24 July 2007

Abstract

This paper shows a novel self-organizing approach for routing datagrams in ad hoc networks, called Distributed Ant Routing (DAR). This approach belongs to the class of routing algorithms inspired by the behavior of the ant colonies in locating and storing food. The effectiveness of the heuristic algorithm is supported by mathematical proofs and demonstrated by a comparison with the well-known Ad hoc On Demand Distance Vector (AODV) algorithm. The differences and the similarities of the two algorithms have been highlighted. Results obtained by a theoretical analysis and a simulation campaign show that DAR allows obtaining some important advantages that makes it a valuable candidate to operate in ad hoc networks and the same method helps in the selection of the algorithm parameters. Since the approach aims at minimizing complexity in the nodes at the expenses of the optimality of the solution, it results to be particularly suitable in environments where fast communication establishment and minimum signalling overhead are requested. These requirements are typical of ad hoc networks with critical connectivity, as described in the paper. Thus the performance of the proposed algorithm are shown in ad hoc networks with critical connectivity and compared to some existing ad hoc routing algorithms.

Keywords: Ad hoc networks; Routing protocol; Ant routing; Critical connectivity

1. Introduction

Routing protocols for ad hoc networks may be described in terms of the state information characterizing each node and/or in terms of the information exchanged among nodes. *Topology-based* protocols use the principle that each node in a network maintains large-scale topology information. This principle is just the same as what link-state protocols use. *Destination-based* protocols do not maintain large-scale topology information, but only topology information needed to know the nearest neighbors. The best known are distance-vector protocols, which maintain a vector of distances to each destination (hop count or other metrics) for all possible next hops, based on the classical Bellman–Ford routing mechanism.

* Corresponding author. Tel.: +39 075 5853918.
E-mail addresses: laura.rosati@diei.unipg.it (L. Rosati), matteo.berioli@dlr.de (M. Berioli), gianluca.reali@diei.unipg.it (G. Reali).

1570-8705/$ - see front matter
doi:10.1016/j.adhoc.2007.07.003

Another traditional classification is to divide protocols in proactive (table-driven) and in reactive (on-demand). *Proactive* routing protocols maintain tables that store routing information; for any change in network topology, they trigger propagating updates throughout the network in order to maintain a consistent network view. *Reactive* routing protocols are characterized by a path discovery mechanism that is initiated when an information unit needs to get to a given destination. Some of the most known MANET routing protocols are mentioned below.

Destination-Sequenced Distance Vector (DSDV) [1] routing protocol is a proactive destination-based algorithm. The modifications to the Bellman–Ford algorithm include loop avoidance. Also Ad hoc On Demand Distance Vector (AODV) routing protocol [2] is destination-based, it minimizes the number of required broadcast messages by creating routes on an on-demand basis. Dynamic Source Routing (DSR) [3] is reactive and topology based. It uses source routing rather than hop-by-hop routing; each packet is routed according to the routing information carried in its header, which includes the complete, ordered list of mobile nodes through which the packet must pass. Temporally-Ordered Routing Algorithm (TORA) [4] is neither a distance-vector nor a link-state; it belongs to a family of algorithms referred to as "link-reversal" algorithms. It provides multiple routes for any desired source/destination pair and reacts only when all routes to the destination are lost. TORA is reactive in the sense that route creation is initiated on demand. However, route maintenance is done on a proactive basis such that multiple routing options are available in case of link failures.

In this paper we focus on MANETs where the Quality of Service (QoS) requirements consist of a fast communication establishment and a minimum signalling overhead. Such scenarios are referred to as ad hoc networks with critical connectivity. An important instance is represented by ad hoc networks featuring randomly changing topology and potentially sparse and intermittent connectivity with long outages, and thus the unfeasibility to rely on any static or pre-calculated routing information. In such scenarios traditional MANETs routing protocols could work not effectively, since the route discovery process intrinsically relies on the "reachability" of the destination node at the time of route discovery. The objectives in designing an efficient routing protocol for ad hoc networks with critical connectivity should be:

- *low convergence time*: to build routes quickly so that they can be used before the topology changes;
- *robustness*: to react quickly, re-establishing routing when topological changes destroy existing routes;
- minimum routing signaling overhead.

There are many scenarios characterized by critical connectivity, for example military and disaster recovery operations. Another instance is given by Ad hoc Space Networks (ASNs), which have been recently designed for scientific space exploration missions, where the space-borne nodes of the network typically consist of multiple spacecrafts with multiple sensors [5–7]. Space-borne nodes in the future should be self-organizing and able to establish communications with heterogeneous nodes and with pre-existing constellations. Space communication links can be intermittent, with very long propagation delays, and consequently the network may not be connected. In this framework, as well as in other areas (see, e.g., the Saami Network Connectivity (SNC) project [8]), the new paradigm of Delay Tolerant Network (DTN) [9] was proposed. The DTN architecture provides a common method for interconnecting heterogeneous gateways or proxies that employ store-and-forward message routing to overcome communication disruptions. An end-to-end message-oriented overlay called the "bundle layer" is placed on top of the transport layer, it enables the interconnection of different portions of the network where different routing algorithms operate. Thus different routing protocols might be used in this context. The applicability of some of them is constrained by precise assumptions. Another proposed approach is epidemic routing [10]: nodes forward each received datagram to each node in their transmitting range until the datagram reaches the destination. Thus this approach is effective only in very sparse networks.

On the other hand it would be desirable for MANETs with critical connectivity to have one single routing algorithm which could leave aside these assumptions, and that can adapt to the environment, performing better in case of good connectivity and worse in case of very intermittent links. In particular hop-by-hop and ad hoc routing is expected to be the solution, using *incomplete* topology information and probabilistic estimations. Some recent papers have proposed routing algorithms for MANETs based on mobile agents.

An extensive general description of agent-based routing principles and design choices can be found in [11,12]. Agent-based routing algorithms for MANETs have been investigated in [13–16]. They have shown a good dynamic behavior, robustness, and ability to work in a distributed environment. In this paper we contribute on this research trend by proposing a specific routing algorithm which is characterized by reduced signaling load and fast convergence. Our approach may be classified into a specific class of agent-based algorithms which are inspired by the ant colonies foraging behavior.

Recent overview papers on Ant Colony Optimization (ACO) are [17,18]. (ACO) has been successfully applied in many combinatorial optimization problems such as the asymmetric traveling salesman problem [19,20], the vehicle routing problem [21], the quadratic assignment problem [22], the graph coloring problem [23].

A comprehensive description of ant routing algorithms may be found in [24]. An important property which we try to bring to the networking environment is the ability of ants to make use of individuals, implementing very simple rules to self organize and find the optimal path between the nest and the food location. The approach presented in this work consists of using very simple ant-like agents. In fact, we believe that simplicity is a fundamental feature in a difficult MANET environment. In this paper, we analyze the effectiveness of the approach by means of a theoretical analysis and simulations.

In order to achieve a minimum routing signaling overhead, we decided to implement the routing algorithm presented in this paper as *reactive* and *destination-based* for the reasons below. In ad hoc networks with critical connectivity, the long propagation delay and high mobility, could prevent traditional table-driven routing protocols, e.g, DSDV, from performing effectively. Because of limited bandwidth of wireless links, message complexity must be kept low. DSDV generates much more routing traffic than on-demand approaches, due to the fact that DSDV periodically generates routing traffic. Also the overhead of a topology-based algorithm, e.g., DSR, is potentially larger than in destination-based approaches since each DSR packet must carry the complete list of the intermediate mobile node to reach the destination. Moreover a topology-based algorithm is not a distributed approach; as the network becomes larger, control packets and data packets become larger as well. Clearly, this has a negative impact on the limited available bandwidth.

The structure of the paper is as follows. In Section 2 we present the general principle of ant routing algorithms and in Section 3 we define one particular algorithm belonging to this class. In Section 4 we present a simulation approach which enables to verify the effectiveness of the algorithm and the setting of its parameters. In Section 5 we investigate the performance of such approach in comparison with traditional ad hoc networks routing algorithms when operating in condition of critical connectivity. In Section 6 we drive the conclusions of the work.

2. Background on ant routing

Ant colonies are distributed biological systems that, in spite of the simplicity of their components, show highly structured social organization. As a result, ant colonies can accomplish astonishingly complex tasks that could never be performed by a single insect. The basic principle of an ant routing algorithm is that ants deposit on the ground a hormone, the pheromone, while they roam looking for food. Ants can also smell pheromone and tend to follow with higher probability those paths characterized by strong pheromone concentrations. The pheromone trails allow the ants to find their way to the food source (or back to the nest). The same pheromone trails can be used by other ants to find the location of the food sources discovered by their nestmates. It was demonstrated experimentally [25,26] that this pheromone-trail-following behavior gives raise to the emergence of the shortest path.

An ant routing algorithm can be briefly described in the following way (cf. also Fig. 1):

- From each network node, a number of discovery packets (*forward ants*) are sent towards the selected destination nodes. They propagate concurrently and independently.
- In each node routing tables consists of stochastic tables, used to select next hops according to weighted probabilities. These probabilities are calculated on the basis of the pheromone trails left by previous ants.
- While moving, the ants deposit pheromone on the path links, i.e., in the node routing tables they change the probability to select a particular next hop.

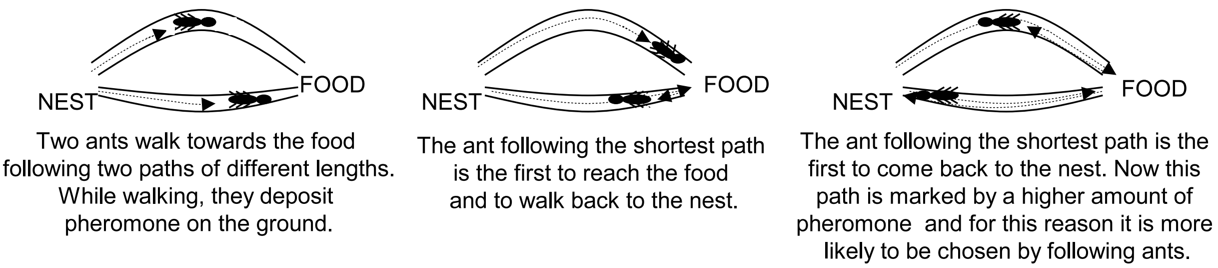

Fig. 1. Basic principle of ant routing paradigm.

- Once a forward ant gets to the destination node, it first generates a *backward ant* and then dies. This way, the new packet created and sent back to the source will propagate through the same path selected by the forward ant.
- On its way back, the backward ant deposits pheromone on the reverse path links. Thus it updates the routing table of the nodes along the path. Once it has returned to the source node, the backward ant dies.

A distributed heuristic solution like the ant routing displays several features making it particularly suitable in ad hoc networks:

- the algorithm is fully distributed ⇒ there is no single point of failure;
- the operations to be performed in each node are very simple;
- the algorithm is based on an asynchronous and autonomous interaction of agents;
- it is self-organizing, thus robust and fault tolerant ⇒ there is no need of defining path recovery algorithms;
- it is intrinsically traffic adaptive without any need for complex and yet inflexible metrics;
- it is inherently adaptive to all kinds of long-term variations in topology and traffic demand, which are difficult to be taken into account by deterministic approaches.

Ant routing algorithms can be classified in different ways, according to how the pheromone is updated, how routing table probabilities are calculated, how often and how many ants are sent per request, and so on. In Fig. 2 we present a possible classification. Using the schematic notation as introduced in the right column therein, the most representative ant routing algorithms to be found in the literature can be listed and categorized as follows (their characteristics are also presented according to the described classification in Fig. 1):

– ABC (Ant-Based Control) [27]: {C3; I2/3; M1; P3}
– ADRA (Ant-based Distributed Route for Ad hoc network) [28]: {C1; I1; M1; P2/3}
– ANB (Ant algorithm for Non-Bifurcated flows) [29]: {C1; I2; M2; P3}
– AntNet [24]: {C4; I2/3; M2; P1/2}
– ARAMA (Ant Routing Algorithm for Mobile Ad hoc Networks) [30]: {C1; I2/3; M2; P3}
– ASGA (Ant System plus Genetic Algorithm) [31]: {C1; I2/3; M2; P3}
– BP-CT (Back Propagation-Cross Target) [32]: {C1; I2/3; M1; P3}
– CAF (Cooperative Asymmetric Forward) [33]: {C1; I2; M2; P1}
– GARA (Genetic Ant Routing Algorithm) [34]: {C4; I2; M2; P2}
– MABR (Mobile Ants Based Routing) [35]: {C3/4; I1; M1; P3}
– PERA (Probabilistic Emergent Routing Algorithm) [36]: {C3/4; I2; M1; P2/3}
– RBA (Routing By Ants) [37]: {C1; I2/3; M2; P1}
– Regular Ant Algorithm [38]: {C3; I2; M1; P2/3}.

It is worth to be noted that some of the algorithms of Table 1 (namely ARAMA, MABR, PERA, ADRA) have been proposed explicitly for MANETs, whereas the others for data communication networks in general. Moreover some of these approaches (namely ABC, ANB, ARAMA, ASGA, GARA, RBA) are connection-oriented, whereas the others connection-less. Other ant-based routing protocols like Ant-AODV Hybrid Routing protocol [39,40] and GPS Ant-Like Routing Algorithm (GPSAL) [41] have not been included in Table 1 because they employ ants only to collect and disseminate up-to-date routing information about the

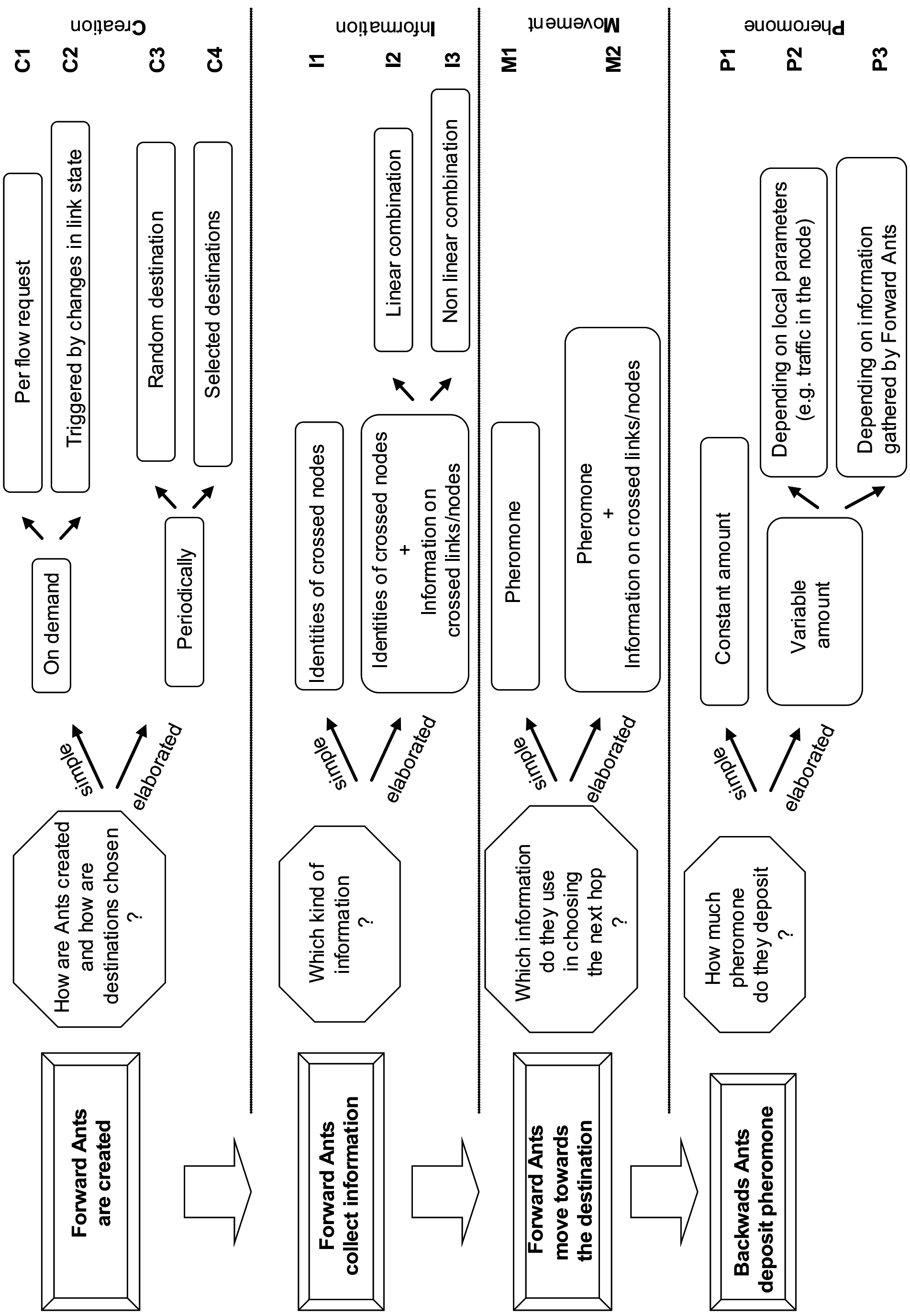

Fig. 2. Ant routing algorithms classification.

Table 1
Some ant routing algorithms

Algorithm	Ants sending	Information collected by Forward Ants	Parameters considered in choosing the next hop	Amount of deposited pheromone
ABC [27]	– Periodic (destinations are randomly selected)	– Identities of the crossed nodes – Launching time	– Pheromone	– Amount depending on the ant age
AntNet [24]	– Periodic (destinations selected according to traffic patterns)	– Identities of the crossed nodes – Time elapsed between ant launch and arrival at each node	– Pheromone – Queue length at current node	– Constant amount (in some versions of the algorithm) – Amount depending on the local traffic model
ADRA [28]	– Triggered by connection requests	– Identities of the crossed nodes	– Pheromone	– Amount function of different parameters (distance from the source node, quality of the link, congestion, velocity of the nodes)
ANB [29]	– Triggered by connection requests	– Identities of the crossed nodes – Bandwidth requirements of the crossed nodes	– Pheromone – Residual capacity of arcs – Distance from the current node to the destination node of the ant	– Amount depending on the length of the ant's route
ARAMA [30]	– Triggered by connection requests	– Identities of crossed nodes – Link costs – Other information related to the crossed links (queuing delay, SNR, bit error rate,...)	– Pheromone – Information on the neighboring node (queuing delay, SNR, bit error rate, remaining battery energy,...)	– Amount depending on the quality of the found path
ASGA [31]	– Triggered by connection requests	– Identities of crossed nodes – Link costs (the link costs are expressed as a function of the link utilization)	– Pheromone – Link costs – Linear combination of the two by means of genetically encoded weights	– Amount depending on the quality of the whole path found
BP-CT [32]	– Triggered by connection requests	– Identities of crossed nodes – Times at which the nodes have been traversed	– Pheromone	– Amount depending on the time length of the found path
CAF [33]	– Triggered by connection requests	– Identities of crossed nodes – Links'costs	– Pheromone[a] (a function used to shape the probabilities in order to favor the best paths)	– Constant
GARA [34]	– Periodic (destinations selected according to traffic patterns)	– Identities of crossed nodes – Times at which the nodes have been traversed	– Path base is maintained in each node for each destination. Each path bases is evolved by genetic algorithm	– Amount depending on the local traffic model
MABR [35]	– Periodic	– Identities of crossed nodes	– Pheromone	– Amount depending on the logical link costs

PERA [36]	– Periodic	– Identities of the crossed nodes – Times at which the nodes have been traversed	– Pheromone[b]	– Function of some metric or a combination of metrics, e.g. delay or the number of hops
RBA [37]	– Triggered by connection requests	– Identities of crossed nodes – Link costs	– Pheromone – Link costs – Linear Combination of the two	– Constant amount
Regular Ant Algorithm [38]	– Periodic (to randomly chosen destinations)	– Identities of crossed nodes – Link costs	– Pheromone	– Non-decreasing function of the links' costs

[a] In [33] the term "learning parameter" is used instead of pheromone.
[b] In [36] the term "reinforcement parameter" is used instead of pheromone.

location of the nodes, and hence, do not make use of the previously described pheromone process for finding shortest paths.

In [42,43] we have proposed and analyzed a very simple connection-less ant routing approach for MANET. We called it Distributed Ant Routing (DAR) algorithm. The definition of this approach was made on the basis of a categorization of the most representative ant routing algorithms, in order to design an approach which requires a low computational complexity.

Specifically, the four main ant routing algorithm characteristics listed in Fig. 2 and Table 1 are implemented in DAR in the simplest possible way, yet preserving the effectiveness of the approach. In DAR routes are created on-demand, in order to have a low routing signalling load with respect to proactive approaches. Forward ants collect information only about the identities of the crossed nodes. Forward ants move towards the destination choosing the next hop only on a pheromone basis. The amount of pheromone deposited by backward ants on each crossed link is constant.

The simplicity of the protocol could be helpful in achieving seamless routing in networks constituted by heterogeneous elements. Moreover, if the routing protocol is simple, the network can be expanded with additional nodes without requiring complex update procedures. For this reason DAR was proposed in [42,44,45] as a powerful mean to enhance communications in meshed regular and irregular satellite constellations; in such complex network scenarios an heuristic approach like DAR is an appealing solution if traditional (deterministic) techniques either fail completely or at least face intractable complexity. As shown in the following, even if DAR is very simple there are several parameters to be set.

With respect to the previously quoted works on DAR, this paper presents a deeper comparison

Table 2
General simulation parameters

Transmission range	100 m
Wireless link shared capacity (C_l)	1 Mb/s
Number of mobile nodes (N_{MN})	30
Traffic type	Constant bit rate
Simulation time	25000 s
Packet rate	4 packets/s
Packet size	500 byte

metric analysis and mathematical insight on the performance of routing in MANETS.

We study in depth the proposed algorithm and compare it with the well-known AODV, which is the most well-known MANET protocol which shares the main DAR features, e.g., the on-demand and hop-by-hop behaviors. Thus we proceed describing their characteristics.

2.1. Ad hoc on-demand distance vector and distributed ant routing

The AODV algorithm can be essentially described in the following way:

- When a node has to find a next hop for a packet to a given destination, it broadcasts Route REQuest packet (RREQs); meanwhile, the packet that can not be forwarded is buffered until a valid next hop is found.
- Each node which receives this (RREQ) stores a reverse route state from itself back to the source. When the reverse routes are timed out, they are deleted.
- Once the (RREQ) reaches a node (eventually the destination) with a sufficient fresh route to the destination, i.e., a route characterized by a sequence number which is higher than the one stored in the packet itself, a Route REPly packet (RREP) is generated and sent back to the source, through the reverse route previously created.
- On its way back, the (RREP) updates the routing table of the nodes along the path.

The DAR algorithm will be described in details in the next section. For the moment it is sufficient to highlight its similarities and differences with respect to AODV. Both DAR and AODV are characterized by the following features:

- They enable dynamic, self-starting and multihop routing in ad hoc networks.
- They are on-demand routing algorithms, thus each route from any source to any destination is searched when data have to be sent.
- Each node maintains a routing table with a routing entry for each possible destination.
- When a routing entry for one destination points to a valid next hop, the routing entry is said to be "available" or "up"; if the information in the routing entry is too old or expired the routing entry is "not available" or "down".
- If a packet has to be sent to a destination for which the routing entry is "down", a route discovery process has to be started to find a good and valid next hop, and the packet is buffered.
- The state created in each node along the path is a hop-by-hop state, meaning that each node does not know the whole path to the destination, but only the next hop node.
- In order to update the topology of the network, periodic "HELLO" packets are sent from each node to its neighbors (that is to the nodes staying within a specified distance).

DAR and AODV differ mainly in the following features:

- While in AODV the routing tables are deterministic, in DAR they are stochastic, meaning that the next hop is selected according to weighted probabilities.
- In AODV each routing entry is associated with a sequence number which indicates how recently the route was used; DAR does not use sequence numbers: routes not recently used are purged by means of pheromone evaporation.
- When a link is not available anymore, in AODV a Route ERRor Packet (RERR) is sent to all the nodes using the link to forward packets, so that relevant routing tables can be changed; in DAR, error messages are not needed. In fact, if the current node cannot forward a datagram due to the lack of a valid routing table entry, then the node starts searching a new route by sending forward ants.

3. The distributed ant routing algorithm

In DAR, in each node the routing tables are stochastic: next hop is selected according to weighted probabilities, calculated on the basis of the pheromone trails left by ants. When a node receives a datagram with destination d, if the routing entry for d is available, then the datagram is forwarded. Otherwise, the datagram is buffered at the node and forward ants are sent out at constant rate r_{ae} (ant emission rate) in order to search a path to d.

Two hop-by-hop routing modes can be implemented: hop-by-bop random routing (nodes *randomly* choose a neighbor to deliver datagrams) and hop-by-hop optimal routing (nodes choose *opti-*

mal next hop to deliver datagrams).[1] Previous results, e.g., [46], show excellent results for hop-by-bop random routing in the case of static networks with relatively small topologies. However, as also stated in [36], this might not be a suitable method for MANETs with rapid topology changes. For this reason DAR adopts hop-by-hop optimal routing. The forward ant is routed at each node according to the probabilities for the next hop in the routing table at the current node. Thus, the forwarding of the forward ant is *probabilistic* and allows exploration of paths available in the network. Datagrams are routed *deterministically* based on the maximum probability at each intermediate node from the source node to the destination node. This process creates a complete global route by using local information.

We have designed this ant routing algorithm according to the principle of a maximum simplicity, thus we have assumed that ants can only deposit a constant amount of pheromone while moving and that they can only be influenced by the presence of the pheromone in the path selection. Thus, the forward ants store only the identities of the visited nodes in order to avoid cycles. Once a forward ant gets to the destination node, it first generates a *backward ant* and then dies. This way, the new packet created and sent back to the source will propagate through the same path selected by the forward ant. As a backward ant travels, it deposits pheromone on the crossed links as described below, updating the routing table of the nodes along the path. Once it has returned to the source node, the backward ant dies.

Being j the current node, i the node the backward ant comes from and τ a constant value, with $0 < \tau < 1$, $\tau_i n$ is the amount of pheromone on the link (j, i) after n backward ants coming back to j. In the process of pheromone update this quantity is multiplied by $(1 - \tau)$ and then τ is added in order to calculate $\tau_i(n + 1)$. The pheromone quantities on the other links are multiplied by $(1 - \tau)$. This simulates the deposit of a constant amount of pheromone:

$$\tau_k(n+1) = \begin{cases} \tau_k(n)(1-\tau) + \tau, & k = i, \\ \tau_k(n)(1-\tau), & k \neq i. \end{cases} \tag{1}$$

The probabilities that a forward ant will select a particular next hop i can be calculated as follows. We will call $p_i(n)$ the probability for a forward ant in node j to choose the node i as the next hop after n backward ants coming back to j. If N is the number of neighbors of j, then we have

$$p_i = \frac{\tau_i(n)}{\sum_{k=1}^{N} \tau_k(n)}. \tag{2}$$

This ensures that the sum of all the probabilities relevant to all the valid neighbors is 1.

Let $y_i(n)$ be a binary variable which is 1 if the nth backward ant crosses the link (i,j), 0 otherwise. Let τ_0 denote $\tau_k(0)$ for $k = 1 \ldots N$. It follows that for a particular next hop i:

$$\tau_i(1) = \tau_0(1-\tau) + y_i(1)\tau, \tag{3}$$

$$\begin{aligned} \tau_i(2) &= [\tau_0(1-\tau) + y_i(1)\tau](1-\tau) + y_i(2)\tau \\ &= \tau_0(1-\tau)^2 + y_i(1)\tau(1-\tau) + y_i(2)\tau \end{aligned} \tag{4}$$

and, in general,

$$\tau_i(n) = \tau_0(1-\tau)^n + \sum_{l=1}^{n} y_i(l)\tau(1-\tau)^{n-l}. \tag{5}$$

The sum of the pheromone on the links departing from j after n backward ants coming back to j is

$$\tau_{\text{tot}}(n) = \sum_{k=1}^{N} \tau_k(n) = N\tau_0(1-\tau)^n + \sum_{l=1}^{n} \tau(1-\tau)^{n-l}. \tag{6}$$

Consequently:

$$p_i(n) = \frac{\tau_0(1-\tau)^n + \sum_{l=1}^{n} y_i(l)\tau(1-\tau)^{n-l}}{N\tau_0(1-\tau)^n + \sum_{l=1}^{n} \tau(1-\tau)^{n-l}}, \tag{7}$$

which can be written as

$$p_i(n) = M(n) + \sum_{l=1}^{n} y_i(l)\Delta p_l(n), \tag{8}$$

where

$$M(n) = \frac{\tau_0}{N\tau(0) + \sum_{t=1}^{n} \tau(1-\tau)^{-t}} \tag{9}$$

and

$$\Delta p_l(n) = \frac{\tau(1-\tau)^{-l}}{N\tau_0 + \sum_{t=1}^{n} \tau(1-\tau)^{-t}}. \tag{10}$$

[1] Similarly, some connection-oriented approaches, e.g., GARA, are capable to provide two path base dependent source routing modes: source random routing (source nodes randomly select a path from path base, and send data packets along it) and source optimal routing (source nodes select the optimal path in path base, and send data packets along it).

After n ants coming back, the term $\Delta p_l(n)$ represents the increment in the probability $p_i(n)$ provided by the l-backward ant, with $l \leqslant n$.

3.1. Pheromone evaporation

In real ant colonies pheromone also evaporates; this process allows selecting new directions without being over-constrained by previous decisions. This is particularly important in case of variable dense topologies and it can be included in ant routing implementations (additional information on pheromone evaporation can be found for instance in [30,37]).

The pheromone evaporation can be simulated by updating the values of the pheromone on every link at regular time intervals Δt_{ev}. For the sake of simplicity below we will call m the generic time instant $m\ \Delta t_{ev}$.

Evaporation is performed simply by multiplying the value of the pheromone on the kth link by a factor smaller than 1.

Thus, being j the current node, we can define $\tau'_k(m)$ as the new quantity of pheromone on link (j,k), which takes into account the evaporation process. Now we suppose that between the time instants m and $m+1$ one backward ant crosses the link (i,j). The new operation of pheromone update, which takes into account evaporation, is performed according to the following two-steps rule:

$$\hat{\tau}_k(m+1) = \begin{cases} \tau'_k(m)(1-\tau)+\tau, & k=i, \\ \tau'_k(m)(1-\tau), & k \neq i, \end{cases} \tag{11}$$

$$\tau'_k(m+1) = [\tau'_k(0) - \hat{\tau}_k(m+1)]k_{ev} + \hat{\tau}_k(m+1). \tag{12}$$

$\hat{\tau}_k(m+1)$ is an auxiliary function only needed as intermediate step to calculate $\tau'_k(m+1)$. $\tau'_k(0)$ is clearly the initial value of pheromone on each link and k_{ev} is a constant <1. For every node $\tau'_k(0)$ has to be a constant $\forall k = 1,\ldots,N$, since at the beginning the probability to select a particular next hop is the same for all neighbors ($p_k(0) = 1/N$, $\forall k = 1,\ldots,N$). We will assume for the sake of simplicity $\tau'_k(0) = 1, \forall k = 1,\ldots,N$.

It can be easily demonstrated that according to this evaporation formula, for every possible value of $\tau'_k(m)$ with $m = N_{ev}n, \tau'_k(m+l)$ tends to $\tau'_k(0)$ if l goes to infinity, that is selection probabilities become uniform if pheromone keeps evaporating without being updated.

If we consider at time m any pheromone quantity $\tau'_k(m)$ and we consider that from that moment the pheromone is not updated by backward ants anymore but it evaporates, then the pheromone will be only changed every N_{ev} timesteps. Thus it will clearly result:

$$\tau'_k(m+l) = \tau'_k\left(m + \left\lfloor \frac{l}{N_{ev}} \right\rfloor N_{ev}\right), \tag{13}$$

and as a consequence:

$$\lim_{l\to+\infty} \tau'_k(m+l) = \lim_{l\to+\infty} \tau'_k\left(m + \left\lfloor \frac{l}{N_{ev}} \right\rfloor N_{ev}\right) = \lim_{p\to+\infty} \tau'_k(m+pN_{ev}), \tag{14}$$

with p integer. It can be demonstrated by induction that, if the pheromone is only changed by evaporation every N_{ev} timesteps, we have

$$\tau'_k(m+pN_{ev}) = k_{ev}\tau'_k(0)\sum_{j=0}^{p-1}(1-k_{ev})^j + \tau'_k(m)(1-k_{ev})^p. \tag{15}$$

Thus it clearly results:

$$\lim_{l\to+\infty} \tau'_k(m+l) = \lim_{p\to+\infty} \tau'_k(m+pN_{ev}) = \tau'_k(0). \tag{16}$$

Thus, before the algorithm starts ($m = 0$), or in case of a long evaporation without updates, we obtain a uniform distribution of the pheromone and for the probabilities: $\tau_i(n) = \tau_i(0) = 1$ and $p_i(n) = p_i(0) = 1/N, \forall i = 1,\ldots,N$.

In the remainder of the section, we use the following notation. We suppose that the first evaporation is performed after f_1 backward ants coming back, the second one after f_2 backward ants coming back and, in general, the pth evaporation is performed after f_p backward ants coming back. For each group f_m of ants coming back to j, let $y_{k,m}(t)$ be a variable which is 1 if the tth ant sent after the $(m-1)$th evaporation crosses the link (k,j), 0 otherwise. We define the parameters $s_{k,y}(f_m)$ and $s(f_m)$ as

$$s_{k,y}(f_m) = \sum_{t=1}^{f_m} y_{k,m}(t)\tau(1-\tau)^{f_m-t} \tag{17}$$

and

$$s(f_m) = \sum_{t=1}^{f_m} \tau(1-\tau)^{f_m-t}, \tag{18}$$

respectively. We define $p'_k(m) = \frac{\tau'_k(m)}{\tau'_{tot}(m)}$ as the probability for a forward ant at node j at the time step

m to choose k as the next node to move to. We denote as k^* the node such that the link (j,k^*) belongs to the shortest path between j and the destination of the flow.

At the beginning each link departing from j is assigned a constant value of pheromone τ_0:

$$\tau_k(0) = \tau_0. \tag{19}$$

Suppose f_1 backward ants come back to node j:

$$\hat{\tau}_k(1) = \tau_0(1-\tau)^{f_1} + s_{k,y}(f_1). \tag{20}$$

If an evaporation is performed:

$$\tau'_k(1) = \tau_0(1-\tau)^{f_1}(1-k_{\mathrm{ev}}) + s_{k,y}(f_1)(1-k_{\mathrm{ev}}) + \tau_0 k_{\mathrm{ev}} \tag{21}$$

Successively f_2 backward ants come back to node j:

$$\begin{aligned}\hat{\tau}_k(2) = {} & \tau_0(1-\tau)^{f_1+f_2}(1-k_{\mathrm{ev}}) \\ & + s_{k,y}(f_1)(1-k_{\mathrm{ev}})(1-\tau)^{f_2} \\ & + \tau_0 k_{\mathrm{ev}}(1-\tau)^{f_2} + s_{k,y}(f_2).\end{aligned} \tag{22}$$

Again an evaporation is performed:

$$\begin{aligned}\tau'_k(2) = {} & \tau_0(1-\tau)^{f_1+f_2}(1-k_{\mathrm{ev}})^2 \\ & + s_{k,y}(f_1)(1-k_{\mathrm{ev}})^2(1-\tau)^{f_2} \\ & + (1-k_{\mathrm{ev}})\tau_0 k_{\mathrm{ev}}(1-\tau)^{f_2} \\ & + (1-k_{\mathrm{ev}})s_{k,y}(f_2) + \tau_0 k_{\mathrm{ev}}.\end{aligned} \tag{23}$$

In general,

$$\hat{\tau}_k(m) = \tau'_k(m-1)(1-\tau)^{f_{\mathrm{m}}} + s_{k,y}(f_{\mathrm{m}}), \tag{24}$$

$$\hat{\tau}_{\mathrm{tot}}(m) = N\sum_{k=1}^{N}\tau'_k(m-1)(1-\tau)^{f_{\mathrm{m}}} + s(f_{\mathrm{m}}), \tag{25}$$

$$\tau'_k(m) = k_{\mathrm{ev}}\tau_0 + \hat{\tau}_k(m-1)(1-k_{\mathrm{ev}}), \tag{26}$$

$$\tau'_{\mathrm{tot}}(m) = N(k_{\mathrm{ev}}\tau_0 + \hat{\tau}_{\mathrm{tot}}(m-1)(1-k_{\mathrm{ev}})). \tag{27}$$

It can be shown that the above equations are equivalent to the following ones:

$$\begin{aligned}\hat{\tau}_k(m) = {} & \tau_0(1-\tau)^{\sum_{l=1}^{m} f_l}(1-k_{\mathrm{ev}})^{m-1} \\ & + \sum_{l=1}^{m} s_{k,y}(f_l)(1-\tau)^{\sum_{v=l+1}^{m} f_v}(1-k_{\mathrm{ev}})^{m-l} \\ & + \sum_{l=1}^{m-1}(1-\tau)^{\sum_{v=l+1}^{m} f_v}(1-k_{\mathrm{ev}})^{m-l-1}\tau_0 k_{\mathrm{ev}},\end{aligned} \tag{28}$$

$$\begin{aligned}\hat{\tau}_{\mathrm{tot}}(m) = {} & N\tau_0(1-\tau)^{\sum_{l=1}^{m} f_l}(1-k_{\mathrm{ev}})^{m-1} \\ & + \sum_{l=1}^{m} s(f_l)(1-\tau)^{\sum_{v=l+1}^{m} f_v}(1-k_{\mathrm{ev}})^{m-l} \\ & + N\sum_{l=1}^{m-1}(1-\tau)^{\sum_{v=l+1}^{m} f_v}(1-k_{\mathrm{ev}})^{m-l-1}\tau_0 k_{\mathrm{ev}},\end{aligned} \tag{29}$$

$$\begin{aligned}\tau'_k(m) = {} & \tau_0(1-\tau)^{\sum_{l=1}^{m} f_l}(1-k_{\mathrm{ev}})^{m} \\ & + \sum_{l=1}^{m} s_{k,y}(f_l)(1-\tau)^{\sum_{v=l+1}^{m} f_v}(1-k_{\mathrm{ev}})^{m-l+1} \\ & + \sum_{l=1}^{m}(1-\tau)^{\sum_{v=l+1}^{m} f_v}(1-k_{\mathrm{ev}})^{m-l}\tau_0 k_{\mathrm{ev}},\end{aligned} \tag{30}$$

which can be equivalently written as

$$\tau'(m) = (1-k_{\mathrm{ev}})^m\left[\tau_0(1-\tau)^{\sum_{l=1}^{m} f_l} + \sum_{l=1}^{m}\left(\frac{(1-\tau)^{\sum_{v=l+1}^{m} f_v}}{(1-k_{\mathrm{ev}})^l}(s_{k,y}(f_l)(1-k_{\mathrm{ev}}) + \tau_0 k_{\mathrm{ev}})\right)\right].$$

The sum of the quantities of pheromone on the links departing from j is:

$$\begin{aligned}\tau'_{\mathrm{tot}}(m) = {} & N\tau_0(1-\tau)^{\sum_{l=1}^{m} f_l}(1-k_{\mathrm{ev}})^{m} \\ & + \sum_{l=1}^{m} s(f_l)(1-\tau)^{\sum_{v=l+1}^{m} f_v}(1-k_{\mathrm{ev}})^{\hat{m}-l+1} \\ & + N\sum_{l=1}^{m}(1-\tau)^{\sum_{v=l+1}^{m} f_v}(1-k_{\mathrm{ev}})^{m-l}\tau_0 k_{\mathrm{ev}}.\end{aligned} \tag{31}$$

Now we are interested in studying the behavior of the algorithm with varying the parameters which characterize the pheromone evaporation. In particular we expect that with increasing k_{ev} and decreasing the evaporation interval, the effect of the process is stronger, in the sense that the probability gets closer to its initial value $1/N$.

For the simulation shown in the remainder of this section, we assume $N = 4.0$, $\tau = 0.3$, $\tau_0 = 1$.

Fig. 3 shows p_{k^*} with $k_{ev} = 0.1$ and with varying m for different values of a constant number f of ants coming back to j between two successive evaporations. For simplicity sake we assume to be in the phase when *all* backward ants cross the link (k^*,j) belonging to the shortest path. From Fig. 3 it can be seen that, for $m > 1$, p_{k^*} increases when increasing f. This fact is reasonable, since bigger values of f mean smaller evaporation rates.

Fig. 4 shows p_{k^*} with varying m for different values of k_{ev}. Again we assume to be in the phase when *all* backward ants cross the link (k^*,j) belonging to the shortest path. The average number f of ants coming back to j between two successive evaporations is set to N. We assume that for $m > 8$ no ants come back to j. From Fig. 4 it can be seen that, for $m > 8$, if evaporation is performed, p_{k^*} decreases when increasing m until the probability reaches its initial value, i.e., $1/N$. Clearly, $\forall m$, the effect of the evaporation increases with increasing k_{ev}.

3.2. Threshold probability

As an extension of existing routing algorithms, we adopted a general criterion to decide whether a routing entry has to be considered either valid or not. This is a common problem in ad hoc networks, since when a route to a particular destination is found, the node never knows how long this information may be kept. Since the node does not know how the topology changes, when the next routing request for the same destination arrives, it will not know whether the old information can be still considered valid.

If *proactive* ant routing protocols are adopted, the routing entry are supposed to be always valid, since the agents are sent periodically to "probe" the network. As far as *reactive* ant routing protocol is concerned, different strategies have been implemented. In some protocols, e.g., ANB, when a backward ant arrives at its destination node, its memory is transferred to a global "daemon" which calculates the best path. In other approaches, e.g., in RBA and in ASGA, if a certain percentage of the previous ants followed the same path, the path is considered valid. An allocator agent is then created to allocate network resources along the best route. We note that the above listed solutions have been applied to connection-oriented protocols.

For connection-less approaches, like DAR, a different solution is required. For instance in ADRA datagrams are sent after the first relevant backward ant has come back to the source of the datagram. This solution could be reasonable if the forwarding

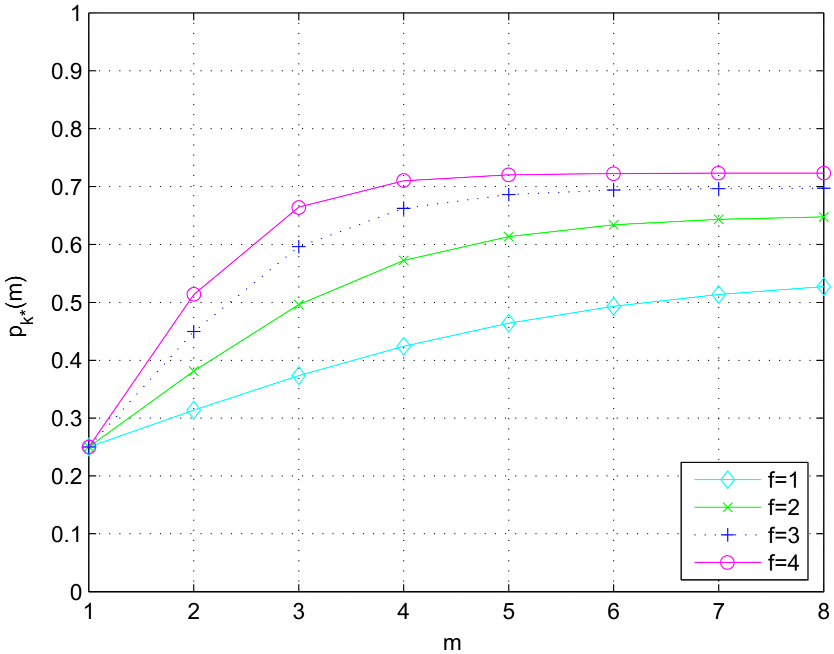

Fig. 3. p_{k^*} with varying m for different values of the average number f of ants coming back to j between two successive evaporations.

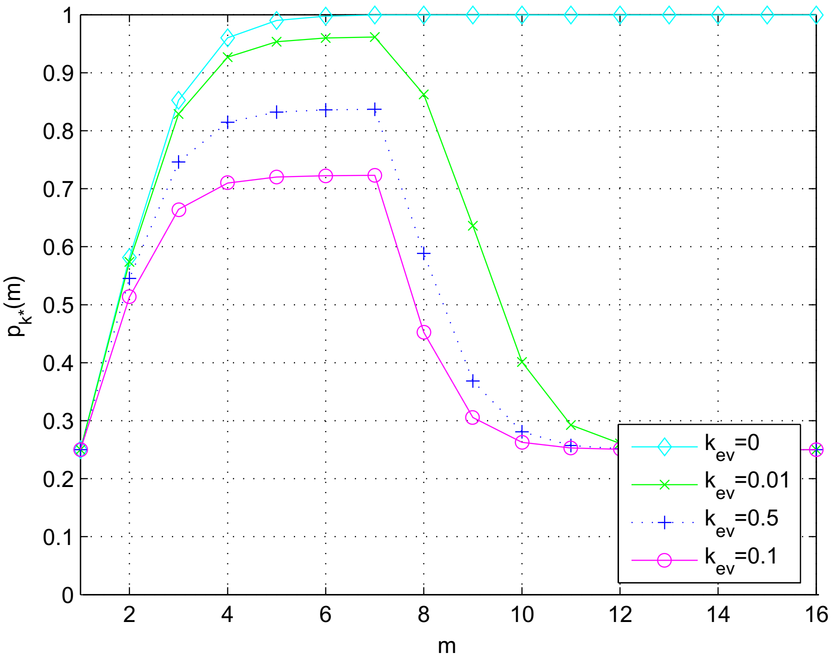

Fig. 4. p_{k^*} with varying m for different values of the average number of k_{ev}.

of all the datagrams on the best path is not the main goal of the algorithm. Nevertheless it might be useful to have the possibility to "tune" somehow the level of optimality of the routes which can be follow by datagrams.

We defined a general criterion for DAR which can be checked by each individual node at each instant in a very simple way without the need of an overall view of the network. For this reason this feature makes this ant routing algorithm fully distributed. In this way local information (next hop probabilities) is used in such a way that global information (a complete route between the source and the destination) emerges from it without direct exchange or synchronization of routing data between the routers.

We set a probability limit, L_p. Every destination d has a routing entry containing the different probabilities to select the different neighbors as the next hop. If in the routing entry at least one neighbor has a selection probability $p_i(n)$ higher than L_p, then this routing entry is labelled with a flag meaning that it is "available". Clearly $L_p > p_i(0) = 1/N$. Thus L_p is a kind of threshold which decides when a routing entry is good enough to be considered as available. This is done since after some ants have come back through one particular neighbor, the probability to choose that link as the next hop increases, meaning that it is a good selection. This brings that probability above the threshold and this will remain only if evaporation does not decrease the probability value again. In principle there might be more than one neighbor with probability $p_i(n) > L_p$, but normally there will be only one, since the ant sending process is stopped when the first good next hop is found. However in case two probabilities are over the threshold, packets are routed to the next hop which presents the higher value. Thus, a certain number of ants coming back from one particular neighbor will be required to increase the probability associated with that neighbor and to make that neighbor the next hop for one destination d.

In Section 4 DAR performance will be assessed with varying algorithm parameters, in particular the threshold probability L_p, which is main novel feature introduced by DAR approach in the framework of ant routing algorithms. In particular the convergence time and the signalling load, as discussed in Section 1, are very important performance parameter in the framework of ad hoc networks with critical connectivity. We expect that the behavior of the convergence time and the signalling load with varying τ and L_p is the same of the minimum number $n_{b,min}$ of backward ants need to be received on a certain link to reach the L_p threshold.

In the initial phase of a route discovery all the links departing from the current node have the same probability to be chosen by the ants. After this initial transient period, the agents will start to converge on the best link. Because of the inherent heuristic nature of ant routing, we are interested in the *minimum* number, $n_{b,min}$, i.e., in the best case, now we calculate this parameter assuming that the best path is already "raised", i.e., the initial transient phase is finished. For simplicity sake we also assume that the pheromone value on the link belonging to the best path is 1 (as a consequence on the other links departing from the current node will be smaller than 1).

It can be easily calculated that:

$$n_{b,\min} = \left\lceil \frac{\ln(N-1) + \ln\left(\frac{L_p}{1-L_p}\right)}{-\ln(1-\tau)} \right\rceil. \tag{32}$$

This minimum value is found by assuming that, during the entire period of time backward ants are coming back, no evaporation takes place and N remains constant (the number of neighbors does not change).

If we call n_m the time instant the mth backward ant moves over the link (i,j), and, as we mentioned previously, we also assume that $\tau_i(n_1 - 1) = 1$, from Eqs. (1) and (2) it easily follows that:

$$p_i(n_1) = \frac{\tau_i(n_1)}{\sum_{k=1}^{N}\tau_k(n_1)} = \frac{1}{1+(N-1)(1-\tau)}, \tag{33}$$

$$p_i(n_2) = \frac{\tau_i(n_2)}{\sum_{k=1}^{N}\tau_k(n_2)} = \frac{1}{1+(N-1)(1-\tau)^2}, \tag{34}$$

and, in general,:

$$p_i(n_m) = \frac{\tau_i(n_m)}{\sum_{k=1}^{N}\tau_k(n_m)} = \frac{1}{1+(N-1)(1-\tau)^m}, \tag{35}$$

which can be written in the following way:

$$m = \frac{\ln(N-1) + \ln\left(\frac{p_i(n_m)}{1-p_i(n_m)}\right)}{-\ln(1-\tau)}. \tag{36}$$

Eq. (32) follows then from the definition of L_p. In Fig. 5 $n_{b,min}$ is plotted as a function of τ with L_p as a parameter. The behavior of the function is reasonable, it can be easily understood that for higher values of τ (or smaller values of L_p), fewer backward ants $n_{b,min}$ are required to the bring a probability value over the threshold L_p.

In Eq. (36) $n_{b,min}$ is the minimum number of backward ants needed to be received on a *certain* link to reach the L_p threshold. Now we try to estimate the minimum number $n_{b,min}$ of backward ants needed to be *totally* received from the node initiator of the route request before the L_p threshold is reached.

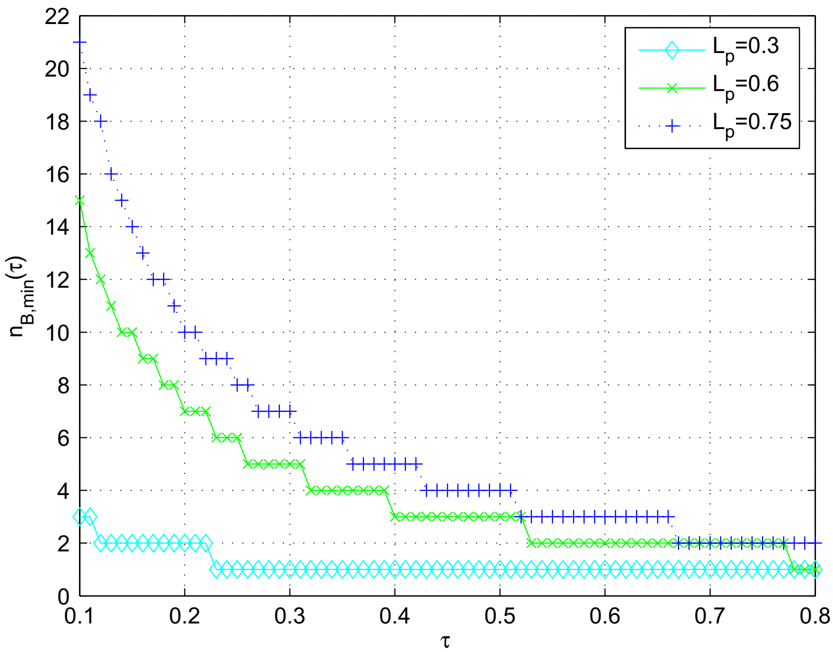

Fig. 5. $n_{B,min}$ versus τ with L_p as a parameter.

We can estimate $n_{b,min}$ from Eq. (7), specifically $n_{b,min}$ can be defined as the minimum n such that $p_i(n) \geqslant L_p$.

In order to calculate $n_{b,min}$ we assume that the pheromone value on the link belonging to the best path is 1, the other links departing from j are

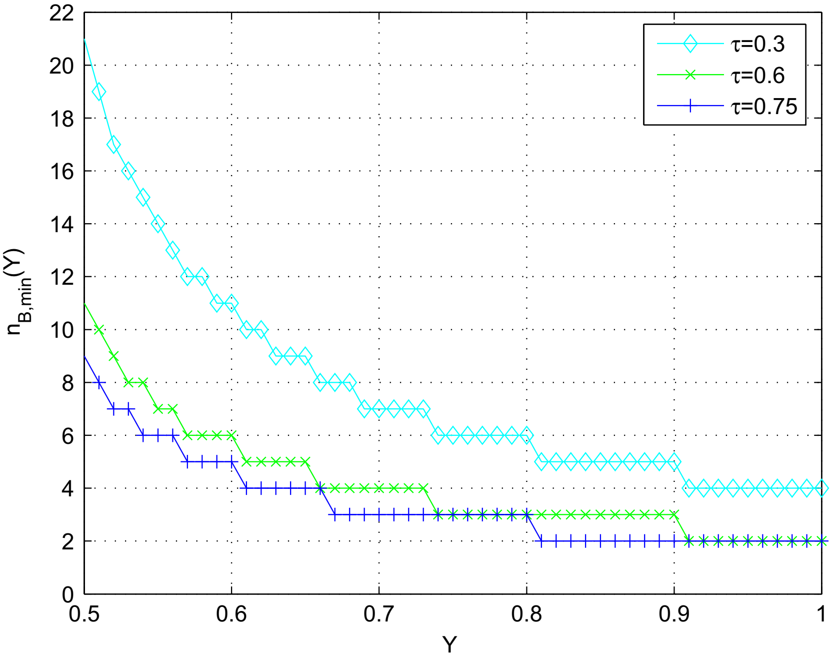

Fig. 6. $n_{B,min}$ versus Y with τ as a parameter ($L_p = 0.4$).

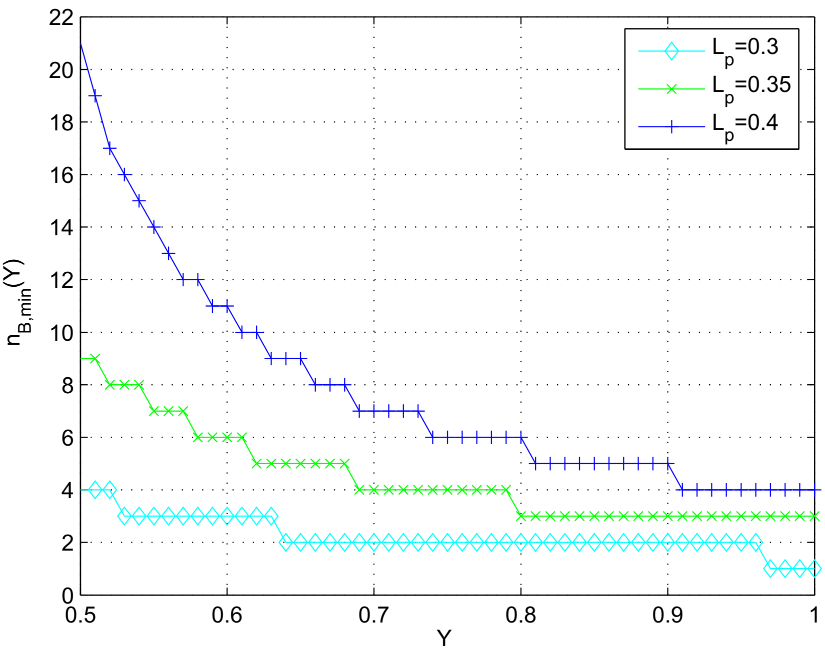

Fig. 7. $n_{b,min}$ versus Y with L_p as a parameter ($\tau = 0.3$).

characterized by a smaller amount of pheromone, no evaporation takes place and N remains constant (the number of neighbors does not change).

We recall that $y_i(n)$ is a binary variable which is 1 if the nth backward ant crosses the link (i,j), 0 otherwise. We can model the fact that not all the ants follow the same path in the following way. We set $y_{k^*}(n) = Y \ \forall n$, where (k^*, i) is the link belonging to the shortest path and Y is a constant, with $0 \leqslant Y \leqslant 1$. Intuitively, if $Y = 1$ it means that all the ants cross the link (k^*, i) (as in Fig. 5), if $Y = 0.5$ it means that half of all the ants cross the link (k^*, i) and so on. In the remainder of the section we assume for simplicity $0.5 \leqslant Y \leqslant 1$, since after the transient phase the ants are supposed to follow with more probability the best path. This is a quite strong assumption, which leads to an underestimate of the signalling load and the time employed to find the path. This fact is particularly true for values of L_{p} close to 1. Nevertheless, the assumption is necessary because, due to the inherent heuristic nature of ant routing, it results quite difficult to exactly estimate such values. The assumption allows us to investigate algorithm convergence towards the solution with varying algorithm parameters.

Fig. 6 shows $n_{\mathrm{B,min}}$ versus Y with τ as a parameter ($L_{\mathrm{p}} = 0.4$).

Fig. 7 shows $n_{\mathrm{B,min}}$ versus Y with L_{p} as a parameter ($\tau = 0.3$). In this case $n_{\mathrm{B,min}}$ is calculated by simulation using MATLAB. The values are determined on the basis of its definition.

From Figs. 6 and 7 it can be seen that $n_{\mathrm{b,min}}$ increases with decreasing Y. This is reasonable since if the number of ants crossing the link belonging to the shortest path decreases, the convergence to the optimal solution slows down. The behavior $n_{\mathrm{B,min}}$ varying L_{p} and τ is the same of $n_{\mathrm{b,min}}$.

4. Setting of the parameters

Even if the DAR algorithm is very simple there are several parameters to be tuned, e.g., τ, r_{ae}. Proper tuning of these parameters becomes more difficult if the ant routing algorithm is not in its simplest version and if it involves several additional parameters and functions. This is considered a common problem for ant routing algorithms.

The approaches commonly used to study ant routing algorithms largely exploit simulation software, due to the inherent heuristic features of the model. Thus, in order to understand the characteristics and the performance of the suggested algorithm, a comprehensive simulation campaign has been conducted. Simulations have been done by using the Network Simulator software NS-2.[2]

Due to the extreme conditions in which the routing algorithms work in case of critical connectivity, for the DAR we need to tune the algorithm parameters in a non-critical scenario. Thus we first considered a traditional non-critical ad hoc network of Mobile Nodes (MN) using omni-directional antennas with a communication range of 100 m. The positions of the MNs are defined by the coordinate values x and y, which are randomly chosen in the range of the grid where the nodes can move. The dimensions of the grid and the number of MNs have been chosen in order to have, for the given value of the communication range of the antenna, a mean value of N equal to 4. This means that on the average each node will have four neighbors during the simulations. We adopted a Poisson generation traffic process between uniformly distributed sources and destinations. In the following, the parameter λ denotes the mean flow generation rate over the entire network and μ the mean flow release rate (thus $1/\mu$ is the average flow duration). We set μ to $1/40\ \mathrm{s}^{-1}$. As a consequence the ratio λ/μ represents the mean number of active flows in the whole network. Nodes generate flows with constant bit rate. Each flow is made of 500-bytes-long datagrams sent out every 0.25 s; thus the flow bit rate is 2 kB/s. This was selected assuming that the communication among nodes is mainly composed of short messages (the average resulting length of one flow is 80 kB). Now we need to make some considerations to evaluate what are reasonable values of λ to load such a network. If we consider that in the network there will be an average number λ/μ of active flows, each one transmitting with a bit rate equal to r_{F}, the overall capacity requested to the network will be $\lambda/\mu r_{\mathrm{F}}$. We will consider that one flow is transmitted over a path with average length l_{p}. We can roughly estimate the maximum theoretical number n_{p} of disjoint paths of average length l_{p} available in a network with N_{l} links, as

$$n_{\mathrm{p}} = \frac{N_{\mathrm{l}}}{l_{\mathrm{p}}}. \tag{37}$$

For us it is $N_{\mathrm{l}} = 4N_{\mathrm{MN}}/2$, since every node can communicate on the average with 4 neighbors by means of a wireless shared link of capacity C_{l} and each

[2] http://www.isi.edu/nsnam [47].

node has one single omni-directional antenna. Thus we can think as if in the network there were $N_l = 2N_{MN}$ bidirectional links of capacity $C_l/4$ each. Since the n_p paths are assumed to be disjoint, the network can offer on each path a capacity equal to $C_l/4$. Thus the overall maximum theoretical capacity the network can offer is $n_pC_l/4$. Hence:

$$\frac{\lambda}{\mu} r_F \leqslant \frac{C_l n_p}{4}. \tag{38}$$

Thus using Eq. (37) in Eq. (38), we can estimate a maximum value for λ to load our network as

$$\lambda \leqslant \lambda_{max} = \frac{C_l N_{MN} \mu}{2 l_p r_F}. \tag{39}$$

Experimental estimations of l_p in our scenario result in a $\lambda_{max} \simeq 10\ s^{-1}$.

We stress that the capacity analysis above has been done only in order to evaluate a maximum value for λ. In the actual network the assumption that the paths are disjoint will not hold. Nevertheless, considering a value for λ which is very small with respect to λ_{max}, we can reasonably expect that the network can satisfy the overall requested capacity.

The simulation time was set to 25000 s, which we have demonstrated to be sufficient to prove our theoretical analysis.

The aim here is to verify analytically the L_p-related theoretical analysis presented in Section 3.2, in particular the calculus of Eq. (32). Since Eq. (32) was found assuming that N is constant, we assumed a fixed topology. As a consequence, evaporation is not needed. These assumptions, which may not be correct in a traditional ad hoc environment, reveal to be right in a critical connectivity scenario. They are simple enough to set the parameters if the algorithm has to be used in environments where the main problem is to find *a* solution, no matter if it is optimal or not (this concept will be better outlined in Section 5). For the same reason we do not need to load the network, and thus we set λ to $1/30\ s^{-1}$, which is small enough with respect to λ_{max}. The analysis in this section also allows us to compare the DAR algorithm with the AODV.

We refer to the convergence time t_{conv} as the time elapsed between the event that a datagram triggers the sending of discovery packets in a node and the time when this datagram is forwarded from the node. This value is a measure of how fast the routing protocol can find a next hop for a route request, or equivalently how long it takes to bring a routing entry up, if required.

We define NRL as the ratio of the routing signaling load (in bytes) and the total number of bytes sent. In DAR the signaling load includes the total number of forward and backward ants; in AODV it includes the total number of RREQ, RREP and RERRs. In order to compare the two algorithms, we do not consider the HELLO messages since the amount of these signaling packets is the same in both approaches. The size of each routing packet in DAR is 146 bytes ($S_a = 146$ bytes); in AODV the size of RREQ, RREP and RERR is 48, 44 and 32 bytes, respectively. This was estimated by simulating the signalling packets with the relevant headers. The DAR packets are larger since each ant has to store the identities of the nodes it passed through. In real implementations these packets could contain a field of fixed length, proportional to the maximum number of hops constituting a loop-free path in the network. This is a waste of resources since some space of the field could often remain unused. This problem can be faced by creating a dynamic list in the packet header, as it happens, for instance, in IPv6 [48].

4.1. Setting of τ and L_p

As a preliminary consideration, we should estimate the range where r_{ae} can vary. Clearly the time between the sending of two forward ants has to be longer than the ant transmission time. Thus it results

$$1/r_{ae} > S_a/C_l; \quad r_{ae} < C_l/S_a, \tag{40}$$

where S_a is the size of each ant routing packet and C_l is the link capacity. In this case it results $r_{ae} < 850\ s^{-1}$.

On the other hand it does not make much sense waiting for the first ant to come back to the source before sending the second one. Referring to Round Trip Time (RTT) as the time needed by the last forward ant to reach the destination from the source plus the time needed by its relevant backward ant to come back, we have

$$1/r_{ae} < \text{RTT}. \tag{41}$$

In order to better understand this, we need to make some further considerations. We can assume that when a discovery process is performed the relevant routing table is set to “up” when n_b ants come back to the source. If $1/r_{ae}$ is greater than RTT, the process can be represented as in Fig. 8a.

In this case the routing load on the network is directly proportional to n_b and it does not depend on r_{ae}, since there are no ants roaming in the network if the routing table is "up".

If $1/r_{ae} > \mathrm{RTT}$ also NRL is almost directly proportional to $n_{b,min}$. We have used the RTT mean value obtained by AODV in the same experimental conditions (0.01 s). Simulation results for $r_{ae} = 10\ s^{-1}$ are shown in Fig. 9, where NRL is plotted as a function of τ and for different values of L_p. As expected, the behavior of NRL with varying τ and L_p is the same as $n_{b,min}$ of Fig. 5. This is reasonable, it can be easily understood that for higher values of τ (or smaller values of L_p), fewer backward ants $n_{b,min}$ are required to the increase a probability value over the threshold L_p. We have ran some simulations under the same experimental conditions by using the AODV protocol and we have obtained

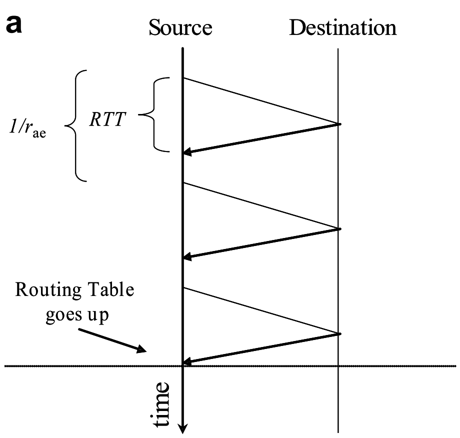

Fig. 8. Scheme of a DAR discovery process ($n_b = 3$) with $1/r_{ae} > \mathrm{RTT}$ (a) and $1/r_{ae} < \mathrm{RTT}$ (b).

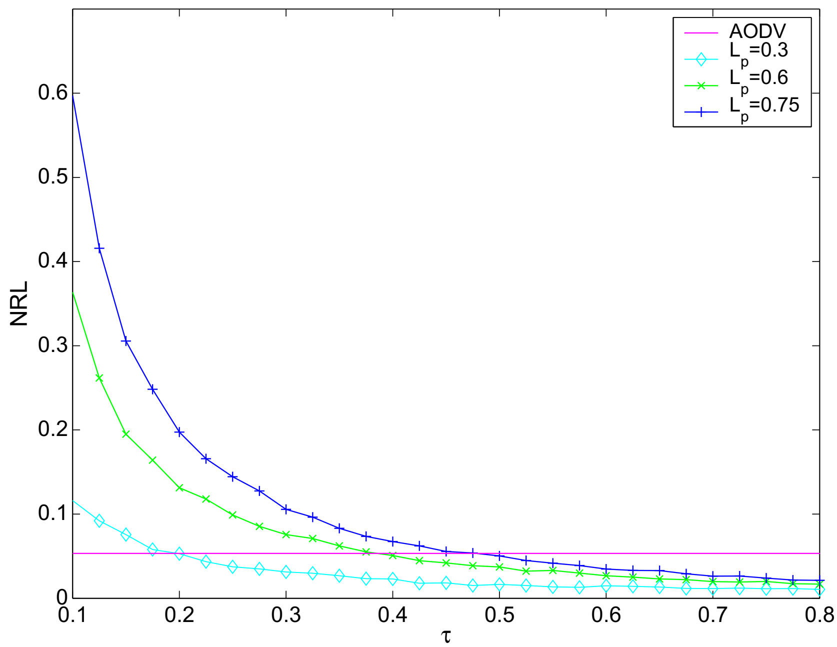

Fig. 9. NRL versus τ with L_p as a parameter.

0.0533 as NRL (as shown in Fig. 9). As it can be clearly seen from the figures, for some values of τ and L_p, DAR outperforms AODV.

In Fig. 10 t_{conv} is plotted as a function of τ with $N = 4$ and with L_p as a parameter. The simulation results are averaged over the simulation time and presented as mean convergence time with the 90% convergence intervals. The behavior of the function is reasonable, again for higher values of τ (or smaller values of L_p), fewer backward ants $n_{b,min}$ are required to the bring next hop selection probability over the threshold L_p. On the basis of the results shown, we can say that a trade-off solution consists of having a reasonable low number of backward ants, for example $n_{b,min} = 6$; in this scenario this can be obtained for values of $\tau = 0.3$ and $L_p = 0.6$.

4.2. Setting of the ant emission rate

Until this point we have considered the case $1/r_{ae} > \mathrm{RTT}$. On the other hand, if the second ant is sent before the first ant comes back to the source, that is if $1/r_{ae} < \mathrm{RTT}$, the situation changes as shown in Fig. 8b. In this case the routing load depends on r_{ae}, since there might be ants roaming in the network even if the routing entry is "up". In fact, before the routing table goes up, $\lfloor \mathrm{RTT}\, r_{ae} \rfloor$ ants are sent out, additionally to the n_b ants which would be strictly necessary. Thus, we can estimate NRL as follows:

$$\mathrm{NRL} = \frac{2 S_a n_{dp} (n_{b,min} + \lfloor \mathrm{RTT}\, r_{ae} \rfloor)}{n_{Bs}}. \tag{42}$$

The factor 2 is due to the fact that each ant coming back to the source is associated with two signaling packets, a forward ant and a backward ant; n_{dp} is the number of discovery processes done in the whole network and n_{Bs} is the total number of data bytes sent.

In order to experimentally verify the validity of Eq. (42), we ran some simulations. The results obtained are shown in Fig. 11, where NRL is plotted as a function of r_{ae} for different values of τ, with $L_p = 0.4$. The results follow expectations, routing signaling load increases almost linearly with r_{ae}, and the values are comparable to AODV performance (0.0533). We can argue that r_{ae} should be small in order to have low signalling load but, on the other hand, having a big r_{ae} would allow having a short convergence time. In fact the minimum possible convergence time $t_{conv\,min}$ can be expressed as a function of r_{ae} in the following way (see also Fig. 8b):

$$t_{conv\,min}(r_{ae}) = \frac{(n_{b,\,min} - 1)}{r_{ae}} + \mathrm{RTT}. \tag{43}$$

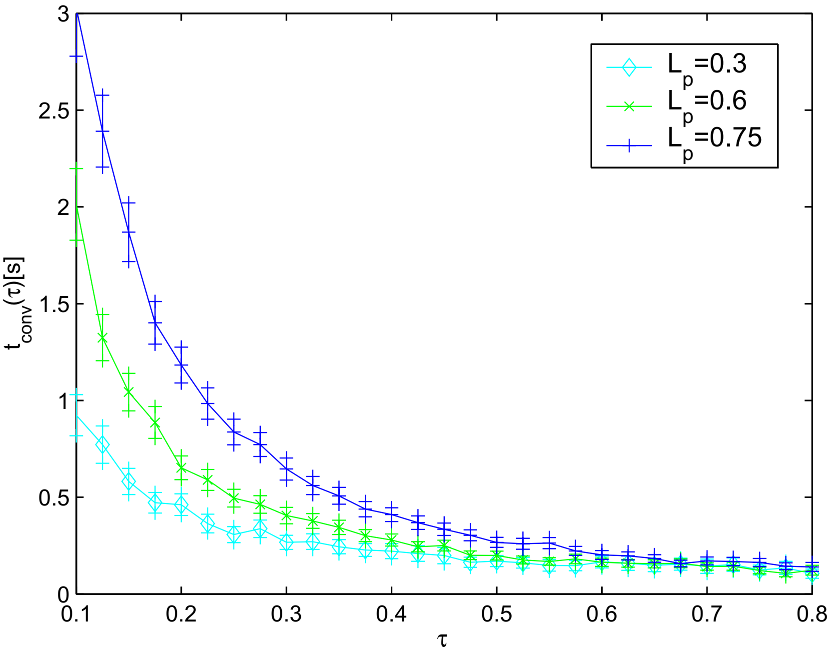

Fig. 10. Convergence time versus τ with L_p as a parameter, together with the 90% confidence intervals.

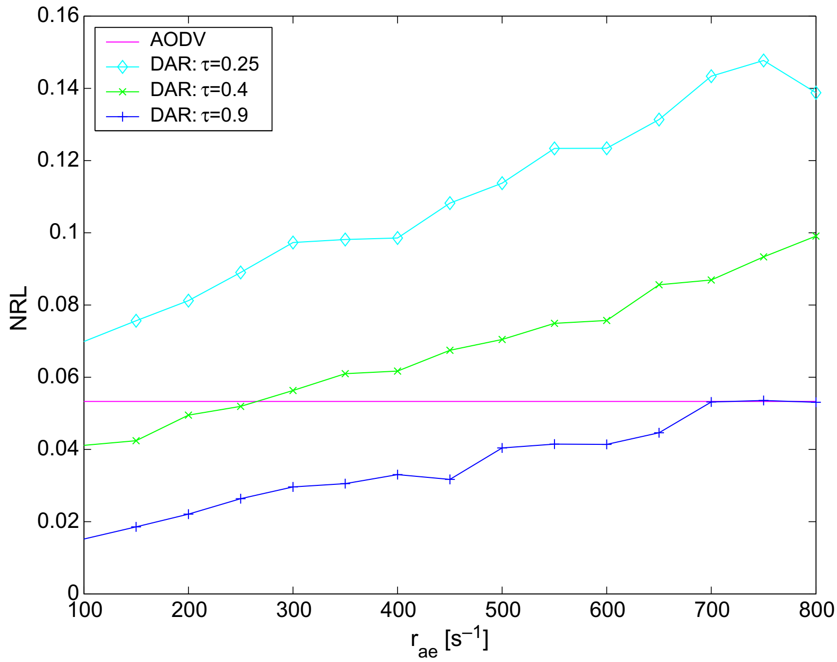

Fig. 11. NRL versus r_{ae} for different values of τ.

By substituting Eq. (32) in Eq. (43) we have

$$t_{\text{conv min}}(r_{ae}) = \left(\left\lceil \frac{\ln(N-1) + \ln\left(\frac{L_p}{1-L_p}\right)}{-\ln(1-\tau)} \right\rceil - 1 \right) \frac{1}{r_{ae}} + \text{RTT}. \tag{44}$$

We stress that Eq. (44) has been calculated by considering $Y = 1$ (see Section 3.2). In Fig. 12 $t_{\text{conv min}}(r_{ae})$ is plotted for different values of τ with $L_p = 0.4$ and $N = 4$.

Fig. 13 shows $t_{\text{conv min}}$ with varying r_{ae} for different values of Y (with $L_p = 0.4$ and $\tau = 0.25$). In this case $t_{\text{conv,min}}$ is calculated by simulation using MATLAB. The values are determined from Eq. (43) substituting $n_{\text{b,min}}$ with $n_{\text{B,min}}$.

From Fig. 13 it can be seen that $t_{\text{conv min}}$ increases with decreasing Y. This is reasonable, because, as also shown in Figs. 6 and 7 of Section 3.2, if the number of ants crossing the link belonging to the shortest path decreases, the convergence to the optimal solution slows down.

In order to experimentally verify the validity of Eq. (44), we ran some simulations and we plotted the measured $t_{\text{conv}}(r_{ae})$ for different values of the τ. The simulation results are averaged over the simulation time and presented as mean convergence time with the 90% convergence intervals. The behavior of the experimental function is the same as the theoretical one. The higher values obtained in Fig. 14 are due to the fact that, in computing Eq. (32), from which Eq. (44) is derived, we optimistically assumed that each forward ant chooses the best path towards the destination.

By comparing Figs. 14 and 13 we can estimate that, on average, 60% of the ants choose the link belonging to the shortest path.

Thus it correctly results that $t_{\text{conv min}}(r_{ae}) \leqslant t_{\text{conv}}(r_{ae}), \forall r_{ae}$.

We have ran some simulations under the same experimental conditions using the AODV protocol and we have obtained a mean convergence time of 0.0175 s. This value is comparable to the ones obtained with DAR, when the values of the parameters τ, L_p and r_{ae} are carefully selected.

The simulation results shown in Figs. 11, 12 and 14 also confirm that for bigger τ we get smaller routing signalling load and convergence time. We can conclude that DAR does not perform worse than AODV if the parameters are correctly chosen. In order to obtain a trade-off in terms of routing load on the network and convergence time, an optimal value for the parameter r_{ae} is around 400 s^{-1}. Actually we can also intuitively understand that in order to have a good route selection, i.e., a low mean end-

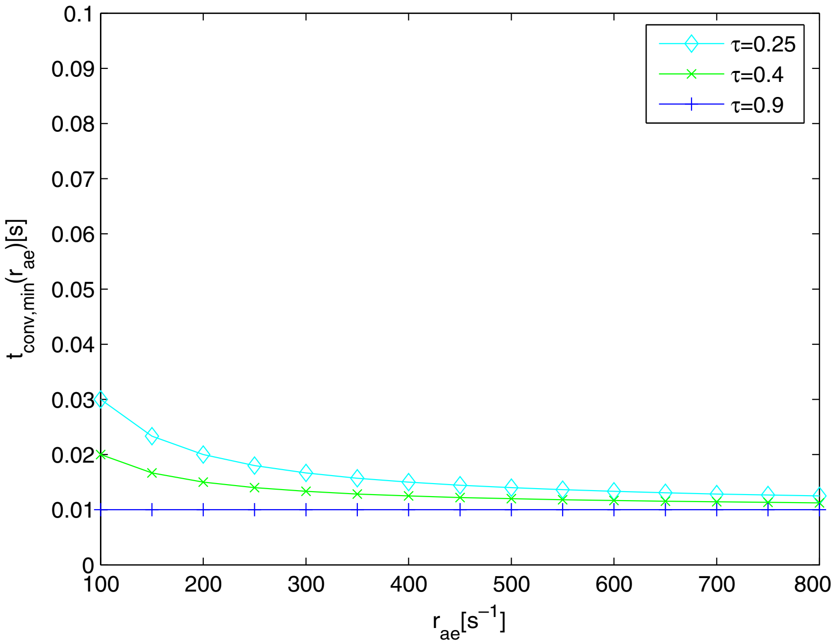

Fig. 12. Convergence time versus r_{ae} with τ as a parameter: analytically estimated minimum mean value.

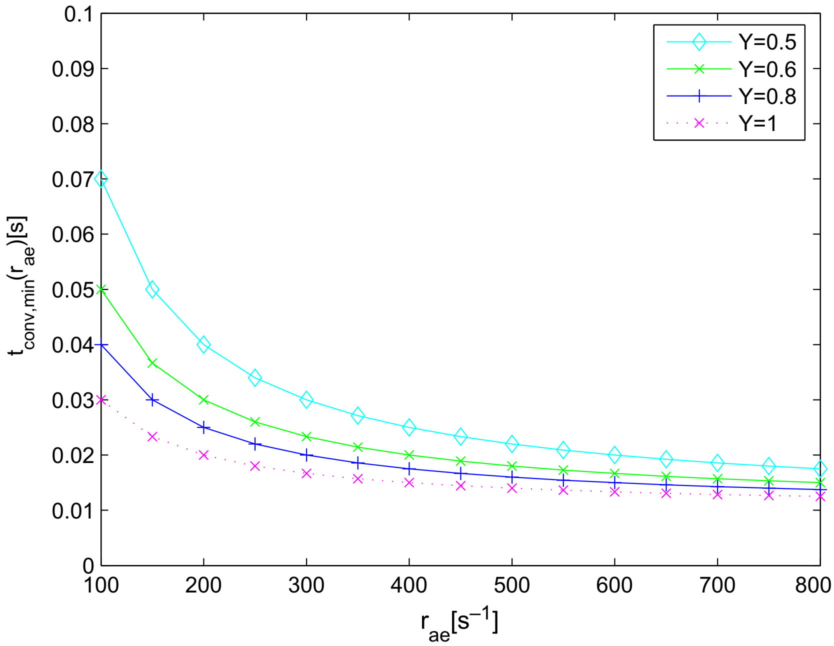

Fig. 13. Convergence time versus r_{ae} with Y as a parameter: analytically estimated minimum mean value.

to-end delay, intended as the time elapsed between a datagram is generated by the source node and it is successfully received at the destination, we need a big number of backward ants coming back, and this means having values of τ close to 0 and values of L_p close to 1.

The DAR performs operations (weighted random selections), which are much simpler than those

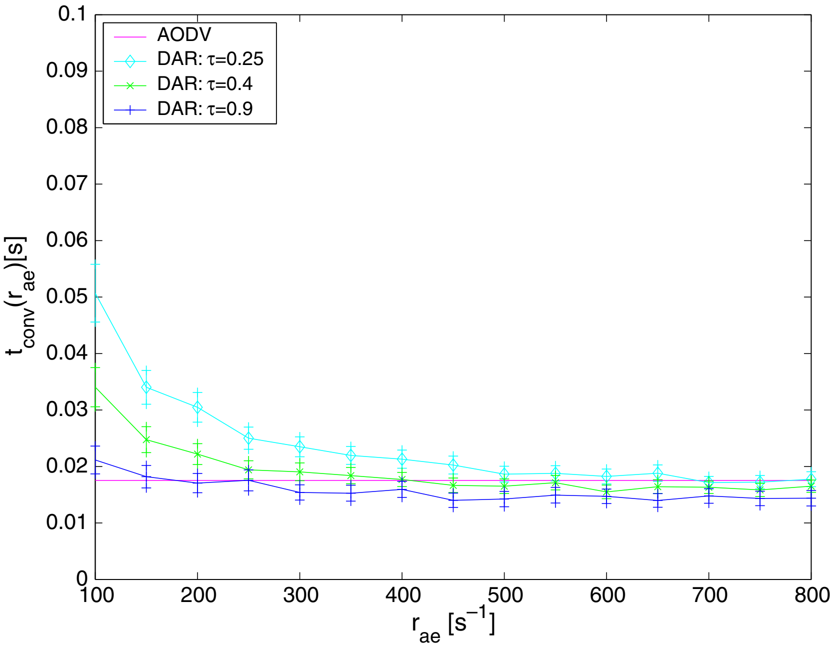

Fig. 14. Convergence time versus r_{ae} with τ as a parameter: simulation results together with the 90% confidence intervals.

performed by conventional algorithms, and this was also not included in the comparison of the protocol "cost". The authors believe that the simplicity of the protocol and the very low complexity largely compensate for the small disadvantage of having one or two more parameters which have to be set.

5. MANET routing in critical connectivity

In this section we investigate the performance of both traditional MANET routing algorithms and DAR, when operating in conditions of critical connectivity and with very intermittent links. The main motivation of using MANET routing in networks with critical connectivity is to be independent from the topology constraints. Clearly a new algorithm specifically designed for a sparse network with a given level of connectivity can perform better; but if the level of connectivity changes new algorithms normally need to be designed. In addition the degree of sparseness of a network may change of some orders of magnitude inside the network itself: in cluster areas many nodes may be close to each others, in other areas there may be isolated nodes with short connection time windows. It would be desirable to have algorithms which operate independently from the local characteristics of the topology, and in particular in the transition region where the connectivity is too low for traditional MANET routing, but it is still too high for specific routing algorithms, i.e., when the average number of neighbors for each node is smaller than some units.

The aim is to optimize the Packet Delivery Ratio (PDR), defined as the ratio of data packets delivered to the destination and those generated by the source nodes. In ad hoc networks with critical connectivity the main goal may be not optimal performance communications; this might be the case if the network is well meshed and the concentration of nodes is high. On the other hand, when the network is very sparse and when it presents long outage periods, it might also be that the main goal is simply to find a way *at all* and *at some time*.

For the simulations of this section we did not implement pheromone evaporation for the following reason. Let us consider the shortest path between a node i and a destination d. We define the path between i and d comprising the link (i,j_1) as path1 and the path between i and d comprising the link (i,j_2) as path2. Now suppose that the shortest path between i and d is path1 and, after some seconds, the topology varies and the shortest path between i and d becomes path2. We consider two cases:

- *Case* 1: If j_1 is still in the transmission range of i, after this change, the ants will continue to select path1 since it is characterized by more pheromone. Nevertheless there will be some forward ants crossing the link (i,j_2). The correspondent backward ants will cross the link (j_2, i) before backward ants cross the link (j_1, i) (since path2 is shorter). As a consequence the quantity of pheromone on link (i,j_1) will start decreasing (see Eq. (1)). This process will change the pheromone on the two links until, depending on the values of τ and L_p, the forward ants will start selecting more frequently path2. Clearly, greater the difference in length between path1 and path2, sooner the forward ants will start selecting with more probability path2. This effect is accelerated if evaporation is implemented. Anyway if evaporation does not take place, this is not a problem since, as already pointed out, path1 is still a suitable path to d and the optimality of the solution found is not the first aim in case of networks with critical connectivity.
- *Case* 2: Now we consider the case in which j_1 is not in the transmission range of i (i.e. the link (i,j_1) does not exist anymore). We note that this case is much more likely than Case 1 for ad hoc networks with critical connectivity. With respect to Case 1, it is less likely that ants will be influenced by past dropped pheromone: After a short transient time, they will find the new right way to the destination by means of their foraging behavior. Thus evaporation is not needed in this case.

5.1. Mobility model

The performance results of an ad hoc network protocol significantly depends on the mobility model adopted for the simulation. A survey of mobility models that are used in the simulations of ad hoc networks can be found in [49,50]. As stated in [51], the use of a mobility model where the new choice for speed and direction is not correlated to previous values, may cause unrealistic movement behaviors with sudden speed changes and sharp turnings. For this reason, we adopted for our simulations the Gauss–Markov mobility model [52,53], which includes both speed and direction dependence from the values at the previous step. The following "border rule" is adopted: when a mobile nodes is subject to leave the simulation area, it is bounced back to this area with a direction which is perpendicular to the side where the bounce occurred. An example of a node movement following the Gauss–Markov model is given in Fig. 15. The values for the parameters used are listed in Table 3.

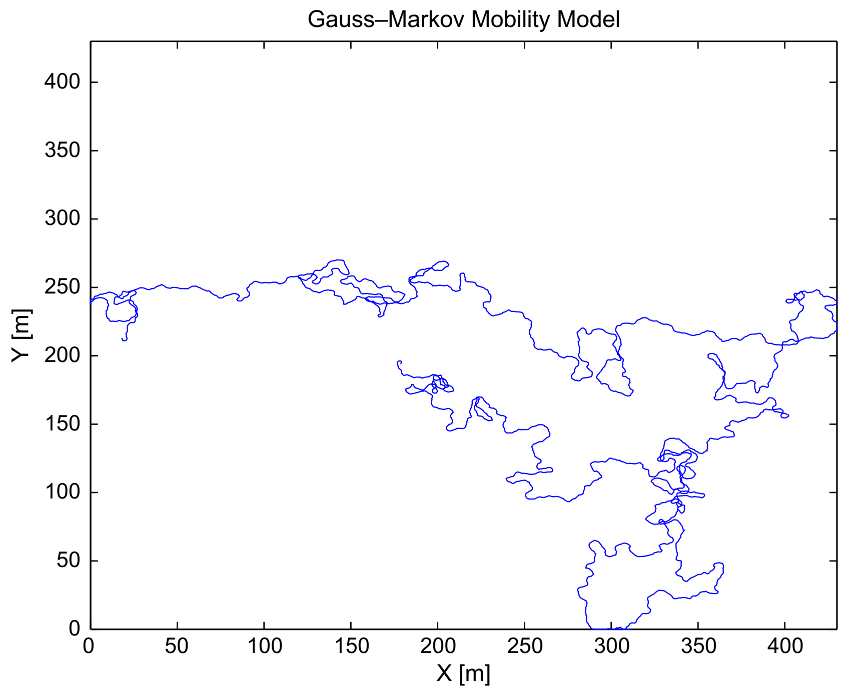

Fig. 15. Example of traveling pattern for a mobile node (duration 400 s, $v_{ave} = 0.1$ m/s).

Table 3
Parameter values of the Gauss–Markov mobility model

Position update interval	0.1 s
α	0.95
d_{ave}	Random selected in $[0, 2\pi]$
v_{ave}	0.1 m/s

5.2. Results of the analysis

Many different approaches to handle routing in ad hoc networks were proposed in recent years [54–56]. In Sections 2 and 4 DAR was presented and its performance assessed in comparison AODV because AODV is the most well-known MANET protocol which shares the main DAR features, e.g., the on-demand and hop-by-hop behaviors. Table 4 shows the characteristics of DAR together with the most well-known protocols in the area of MANET routing algorithms. In this way a range of design choices is covered, including periodic advertisements versus on-demand route discovery and hop-by-hop routing versus source routing.

The DSDV protocol has been included in Table 4 as an instance of proactive routing protocol (see also Section 1). Other proactive MANET routing protocols are Clusterhead Gateway Switch Routing protocol (CGSR) [57] and Wireless Routing Protocol (WRP) [58]. CGSR modifies DSDV by using a hierarchical cluster-head-to-gate-way routing approach. Each node is associated with a cluster member table where it stores the destination cluster head for each mobile node in the network. In WRP a shortest-path spanning tree is reported by each neighbor. For this reason reactions to failures may be far-reaching (i.e., every node which includes the failed link in its shortest-path spanning tree is involved in the failure reaction).

Other MANET routing protocols proposed in the literature have not been considered in this paper because they show characteristics quite different from DAR. For instance, Location-Aided Routing (LAR) [59] protocol belongs to the class of geographic routing algorithms, which limit the search for a route to the so-called request zone, determined based on the expected location of the destination node at the time of route discovery. This information is not always available in networks with critical connectivity.

An other example of MANET routing algorithms is given by hybrid protocols, which group the node into zones and use proactive scheme inside these zones and reactive between zones. In general they show high computational complexity and require additional traffic for creation and maintaining of their topology information. Hybrid protocols are: Core Extraction Distributed Ad Hoc Routing Protocol (CEDAR) [60], Zone-based Hierarchical Link State Routing Protocol (ZHLS) [61], Preferred Link-based Routing Protocol (PLBR) [62], Optimized Link State Routing Protocol (OLSR) [63]. A well-known hybrid protocol is ZRP [64]. We did not consider ZRP for comparison purposes because ZRP is a routing framework rather than an independent protocol. ZRP combines two completely different routing methods into one protocol. Within the routing zone, the proactive Component IntrAzone Routing Protocol (IARP) [65] maintains up-to-date routing tables. Routes outside the routing zone are discovered with the reactive component IntErzone Routing Protocol (IERP) [66] using route requests and replies.

A link in a MANET with critical connectivity could become unavailable for a relatively long period of time and no other alternative routes could be available. Potentially rapidly changing topology makes it important to find routes quickly, even if the route may be suboptimal. Normally the optimal route can be found only if the source node (and not an intermediate node) is the initiator of the route request, but, depending on the changes in topology and on the nodes movement, this route may not always remain the shortest. For this reason traditional MANET routing protocols (see for instance AODV) use error notification messages to discard a route even if only a portion of it becomes unavailable because of topology changing: most likely the available portion will not be a part of the new optimal path, thus it is worth to recalculate the whole route again. On the other hand in MANETs with critical connectivity the focus is not really on path optimality, but rather on a fast reaction to topology changes, since one of the major goals is to deliver as much data as possible to the destination node. In this case it is convenient to store the packets and forward them to the destination once a connection is resumed. If the current node cannot forward a datagram due to a link which becomes available, AODV drops the datagram and a RERR is sent to all the nodes using the link to forward packets, so that relevant routing tables can be changed. In DAR, the datagram is buffered and the node starts searching for a new route by sending forward ants. We define q_{lim} as the maximum number of packets which can be buffered at each node. Fig. 16 shows

Table 4
Some MANET routing protocols

Features	DSDV	DSR	TORA	AODV	DAR
Proactive versus reactive	– Proactive	– Reactive	– Proactive (and reactive)	– Reactive	– Reactive
Routing algorithm	– Distributed Bellman–Ford	– Link state	– Link-reversal relaxation	– Distributed Bellman–Ford	– Ant routing
Forwarding algorithm	– Hop-by-hop	– Source routing	– Hop-by-hop	– Hop-by-hop	– Hop-by-hop
Main information stored in the routing table	– Deterministic routes are maintained in a distributed fashion. Routing table entries are tuples in the form: destination, hops_to_destination, sequence_number	– Each node has a deterministic route cache, where complete routes to desired destinations are stored	– Each node stores the height metric associated with each neighbor and the assigned status of the link to such neighbor	– Deterministic routes are maintained in a distributed fashion. Routing table entries are tuples in the form: destination, next_hop, distance	– Stochastic routing table: the next hop is selected according to weighted probabilities
Route discovery	– Periodic advertisement	– A RREQ is broadcast. Once the RREQ reaches the destination, it replies with a RREP that copies the route from the RREQ and traverses it backwards	– The nodes use a "heigh" metric, which establishes a DAG rooted at the destination. Links are assigned a direction based on the relative height metric of neighboring nodes	– A RREQ is broadcast. Once the RREQ reaches the destination, it replies with a RREP that copies the route from the RREQ and traverses it backwards	– Forward ants are sent. Once they reach the destination, they generate backward ants which copy the route from the forward ants and traverses it backwards
Mechanisms used to guarantee the freshness of the routes	– Each node maintains a monotonically increasing even sequence number, which is disseminated in the network via update messages	– Intermediates nodes do not need to maintain up-to-date routing table	– For each interface a router maintains a sequence number that is incremented upon changes to the interface mode of operation (reactive/proactive)	– The source sequence number is used to maintain freshness information about the reverse route to the source and the destination sequence number specifies how fresh a route to the destination must be before it can be accepted by the source	– Ant routing process, pheromone evaporation
Route maintenance (behavior in case of link failure)	– A RERR is broadcast in order that any route through that next hop is assigned an infinite metric and an update sequence number	– A RERR is broadcast to the source in order to erase all routes in the route caches of all intermediate nodes on its path, if the route contained the failed link	– Done on a proactive basis through link-reversal route repair, whenever topological changes cause a node to loose its last downstream link. In case of a network partition, the protocol erases all invalid routes	– A RERR is sent backwards to the active neighbors, which forward them to their active neighbors and so on. All routing table entries are erased for which the failed link is on the active path	– Node starts searching for a new route by sending forward ants
Route deletion (when route is not necessary)	– Routes are always maintained	– Expiration timer	– Flooding CLR (clear packet)	– Expiration timer	– The labels of the relevant routing table entries are set to "DOWN"

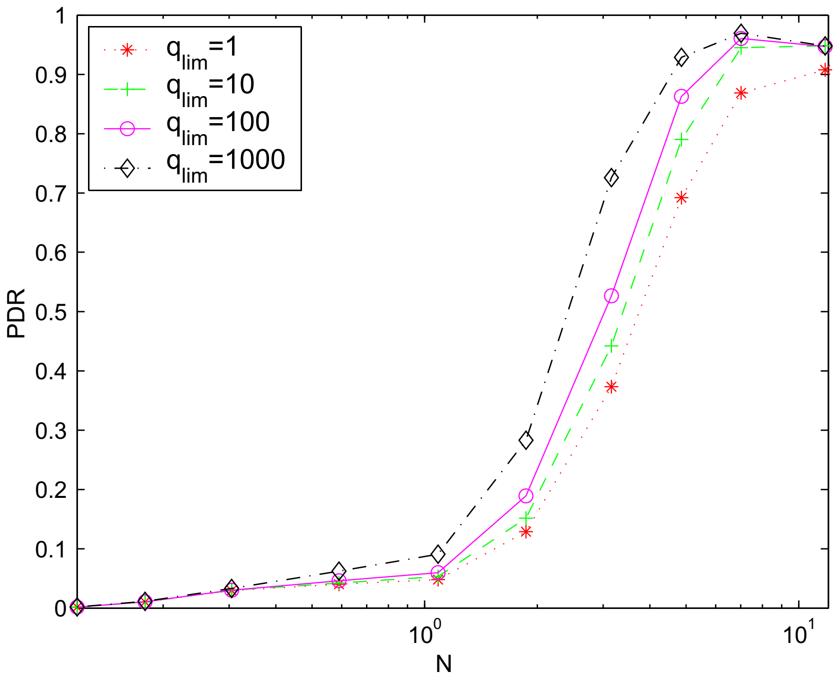

Fig. 16. Packet Delivery Ratio as a function of N for different values of the buffer size using the DAR algorithm.

that, for the DAR algorithm, PDR increases if the maximum number of packets buffered per node also increases. This implies a higher average end-to-end delay experienced by the datagrams to reach the destination, but a higher average fraction of packet delivery. We aimed at analyzing how PDR varies for different values of the average value N of neighbors that the mobile nodes experience over time. By using the same traffic patterns as described in Section 4 and the simulation parameters summarized in Table 2, we have run some simulation for 10 different values of N. N is varied by adjusting the simulation area and the initial positions of the mobile node, chosen in order to have N set to 10 values equally distributed in logarithmic scale in the range [0.1; 10] the number of mobile nodes. The figures below are plotted as functions of the average number N during the whole simulation. During the simulation the mobile nodes move with an average speed of 0.1 m/s.

In Fig. 17 we plotted PDR with varying N for different MANET routing protocols, i.e., AODV, TORA, DSR, DAR ($q_{\text{lim}} = 1000$).

The simulated model scenario is based on the comparison of AODV, DSR and TORA, the three prominent on-demand routing protocols for ad hoc networks. A performance comparison of DSR, TORA and AODV is presented, e.g., in [67–69].

The different basic working mechanisms of AODV, DSR and TORA leads to the differences in performance. The presence of mobility implies frequent link failures and each routing protocol reacts differently during link failures.

Data packets may be dropped for two reasons: the next hop link is broken when the data packet is ready to be transmitted, or there are no available routing table entries for the intended destination. In particular a number of packets are dropped during the route discovery phase.

As we can see from both Figs. 16 and 17, for low connectivity, few data packets are delivered due to lack of a route. Many of the sessions abort because routes to the destination are unavailable. The few sessions that are able to be completed are those with a small path length. As the connectivity increases, however, the number of delivered packets rapidly increases.

The performance of MANET routing protocols depends on a lot of factors, e.g., the mobility model, the number of mobile nodes, the traffic pattern, anyway, we can notice that DAR has a very good performance in comparison with the other algorithms.

The simulations of Fig. 17 show that AODV performs better from the point of view of the PDR compared to DSR and TORA. The same result was obtained in other papers, e.g.,in [69] with and

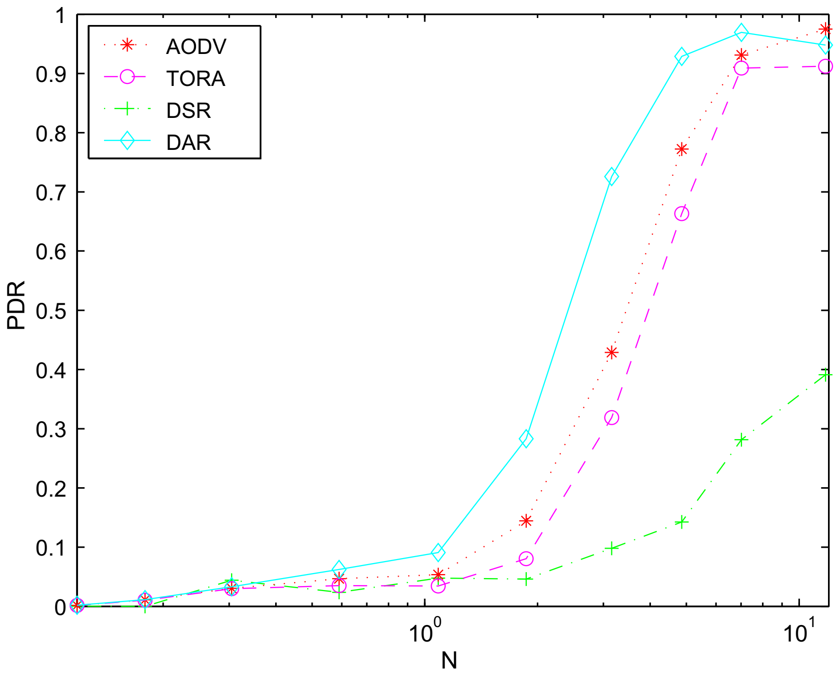

Fig. 17. Packet Delivery Ratio as a function of N using different MANETs algorithms.

without mobility in networks composed of a larger number of nodes (precisely greater than 20).

In AODV each link failure triggers new route discoveries because the routing table has at most one route per destination. AODV also uses route expiry, dropping some packets when a route expires and a new route must be found.

With respect to AODV, TORA causes more packet drops because the asynchrony in the distributed implementation can cause short-lived inconsistencies about the sense of the direction of a link as perceived by the nodes at the end-points of this link. Hence packets drop because of short-lived routing loops. This is a consequence of its link-reversal process. Moreover the initial route discovery takes longer in TORA with respect to AODV. In TORA there is a potential for oscillations to occur, especially when multiple sets of coordinating nodes are concurrently detecting partitions, erasing routes, and searching new paths based on each other [56].

DSR shows the worst performance from the point of view of the PDR. The reason for that be due to the absence of an explicit mechanism to expire stale routes (DSR does not depend on any periodic or timer based activity) and to its aggressive use of source routing and route caching [70]. In DSR, in case of link failure, a new path discovery is delayed until all cached multiple routes for the destination are not available. With high mobility, the cache might become stale. For this reason DSR is intended for networks in which the mobile nodes move at moderate speed with respect to packet transmission latency [3]. Assumptions the algorithm makes for operation are that the network diameter is relatively small.

6. Conclusions

In this paper we show the results obtained by ant-inspired heuristic and distributed algorithms to route packets in a MANET. A new ant routing algorithm, named DAR, has been shown and analyzed by means of theoretical analysis and a simulation campaign. The definition of this approach was made on the basis of a possible categorization of the most well-known ant routing algorithms to be found in the literature, in order to design an approach which requires the minimum computational complexity. The performance comparison of DAR with a well-known reference algorithm in ad hoc networks, the AODV, has revealed that with an appropriate tuning of the parameters, DAR gives better results from the point of view of the signalling load and the convergence time, in the same experimental conditions considered in this paper and that are representative of a MANET scenario. For this reason DAR is suitable in scenarios with critical connectivity, where the QoS requirements consist of a fast reaction to topology changes and a minimum signalling overhead regardless path optimality. An important instance is represented by ad hoc networks with critical connectivity, where traditional MANETs protocols could be ineffective, due to their intrinsic design goal to look for an optimal route. In this challenging scenario the comparison has been extended to other known routing protocol used in MANETs. In addition, the simplicity, flexibility and robustness of DAR are always appealing features which make the approach a good solution in different kinds of topology scenarios.

Acknowledgement

This work has been partially funded by the European Community under the 6th Framework Programme IST Networks of Excellence "SatNEx" (contract No. 507052) and "SatNEx II" (contract No. 027393).

Appendix A. Symbols

See Table 5.

Table 5
List of notations used for the definition and study of DAR

$\Delta p_i(n)$	increment in the probability $p_i(n)$ provided by the l-backward ant, with $l \leqslant n$
Δt_{ev}	evaporation interval [s]
λ	mean flow generation rate [s^{-1}]
μ	mean flow release rate [s^{-1}]
v_{max}	maximum node speed [m/s]
τ	quantity of pheromone deposited on each crossed link by a backward ant
τ_0	initial value of pheromone on each link
$\tau_i(n)$	amount of pheromone on the link (j,i) after n backward ants coming back to the current node j
$\tau_{tot}(n)$	sum of the quantities of pheromone on the links departing from the current node after n backward ants coming back to it
C_l	wireless shared link capacity [Mbps]
k_{ev}	evaporation constant
l_p	average path length
L_p	probability limit
N	average number of neighbors
N_l	number of unidirectional links in the network
n_b	number of backward ants
n_{Bs}	total load sent in the network [bytes]
$n_{b,min}$	minimum number of backwards ants that have to pass a link before it becomes available
n_{dp}	number of ant routing discovery processes
n_p	number of paths
$p_i(n)$	probability for a forward ant at the current node to choose the node i as the next hop after n backward ants coming back to the current node
q_{lim}	buffer size
r_{ae}	ant emission rate [s^{-1}]
r_F	transmission bit rate [bit/s]
S_a	size of DAR routing packets [bytes]
t_{conv}	convergence time [s]
$t_{conv\,min}$	minimum convergence time [s]
$y_i(n)$	binary variable which is 1 if the nth backward ant crosses the link (i,j) (with j current node), 0 otherwise

Appendix B. DAR pseudocode

Fig. 18 shows a flow-chart description of the algorithm. All the described actions take place in a completely distributed and concurrent way over the network nodes.

Tables 6–9 describe in more details the main algorithm steps.

Specifically, when a node receives a datagram, the procedure *Receive_Datagram* (see Table 6) is implemented.

If the routing entry of the node relevant to the destination (*dest_node*) of the datagram is labeled with a flag set to "UP", then the datagram is forwarded according to the next hop stored in the routing table, otherwise it is buffered at the node. If the label is set to "IN_REPAR", it means that ants looking for the best path towards that destination have already been sent. If the flag is "DOWN" then it is set to "IN_REPAIR" and ants are created and sent (procedure *Send_Request*, see Table 7). The procedure *Send_Request* describes the behavior of the ants. Note that the node where an ant is generated (*source_ant_node*) can be different from the source of the flow (*source_node*). Note also that the source of a forward ant, i.e., *source_ant_node*, and its destination are the destination and the source of the corresponding backward ant, respectively.

Forward ants choose the next hop according to the probabilities associated with the links departing from the current node (function *Select_Link*) and store the crossed nodes in the array *List_Crossed_nodes*. This array is used by the forward ants to avoid loops, i.e., the next hop is chosen only among the neighbors of the current node which are not been visited by the ant yet. Moreover the array is used by the correspondent backward ants to find its way back to (*source_ant_node*).

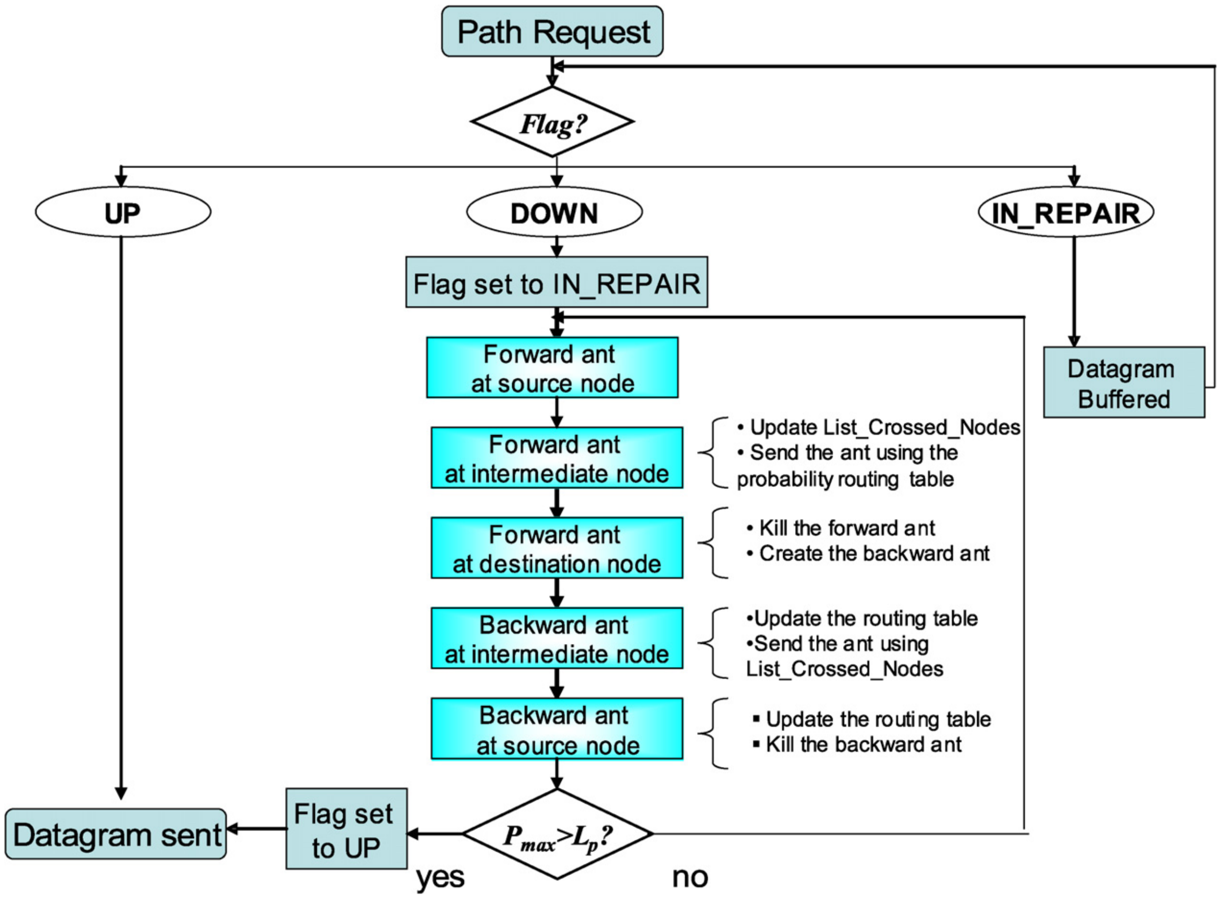

Fig. 18. Top level flow chart describing DAR algorithm.

Table 6
High-level description of *Receive_Datagram* procedure in pseudo-code

```
Procedure Receive_Datagram (p, current_node);
  if (TTL = 0),
    Drop(p);
    return;
  end if;
  dest_node := p ← dest_node;
  rt ← routing_table(current_node, dest_node);
  case (rt ← rt_flag):
  UP
    current_node := rt ← rt_next_hop;;
  IN_REPAIR
    enqueue(datagram, current_node);
  DOWN
    rt_flag := IN_REPAIR;
    enqueue(datagram, current_node;
    SEND_REQUEST(current_node, dest_node;
  end case;
end Procedure Receive_Datagram;
```

When the backward ant gets to its destination, the procedure *Recv_Reply* (see Table 8) is implemented.

According to this procedure, if the condition of the threshold probability is satisfied or *source_ant_node* has only one neighbor, then the relevant *rt_flag* is set to "UP" and the datagrams buffered at *source_ant_node* are forwarded according to the routing table. Each packet (both ants and datagrams) is associated with a Time To live (TTL), which is increased of 1 for each hop done by the packet. When a node receives a packet, the value of its TTL is checked. If the TTL value is 0, then the packet is dropped.

Beside these procedures, also neighbor management functions are implemented (see Table 9). Each node maintains a list of neighbors. If a node receives an "HELLO" packet from a node which is in its neighbor list, then the relevant expiration timer is

Table 7
High-level description of *Send_Request* procedure in pseudo-code

```
Procedure Send_Request(source_ant_node, dest_node);
  i := 0;
  while (current_node ≠ dest_node)
    if (TTL > 0) and (neighbor(j))
      next_hop_node := SELECT_LINK
  (current_node, dest_node, List_Crossed_nodes);
      list_crossed_nodes(i) := current_node;
      i++;
      current_node := next_hop_node;
    else
      DROP;
    end if;
  end while;
  CREATE_BACKWARD_ANT(List_Crossed_nodes);
  KILL_FORWARD_ANT;
  while (current_node neq source_ant_node),
    UPDATE_LOCAL_ROUTING_TABLE(current_node,dest_node);
    next_hop_node := list_crossed_nodes(i);
    i := i − 1;
    current_node := next_hop_node;
  end while;
  RECV_REPLY(source_ant_node, dest_node);
  KILL_BACKWARD_ANT;
end Procedure Send_Request;
```

Table 8
High-level description of Recv_Reply procedure in pseudo-code

```
Procedure Recv_Reply(source_node, dest_node);
  if (p_max(source_node, dest_node) > L_p) or (only_one neighbor
  (source_node)),
    rt ← rt_flag:=RTF_UP;
    dequeue(current_node);
    STOP_SENDING_ANTS;
  else
    rt ← rt_flag:=RTF_DOWN;
  end if;
end Procedure Recv_Reply;
```

Table 9
High-level description of Recv_Hello procedure in pseudo-code

```
ProcedureRecv_Hello(current_node, neighbor_node);
  if Nb_Lookup(neighbor_node),
    Update_Expiration_Timer(neighbor_node);
  else;
    Add_Neighbor (neighbor_node);
  end if;
  Kill_Hello_Packet;
end Procedure Recv_Hello;
```

updated (procedure *Update_Expiration_Timer*). If a node does not receive within a pre-defined interval an "HELLO" packet from one of its current neighbors, then this timed-out neighbor is deleted from the list. If a node receives an "HELLO" packet from a node which is not currently in its neighbor list, then this neighbor is added in the list (procedure *Add_Neighbor*). In both cases, the values of the pheromones and the probabilities associated with the links departing from the current node are updated accordingly. Then the "HELLO" packet is dropped.

References

[1] C.E. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, Computer Communications Review (10) (1994) 234–244.

[2] C.E. Perkins, E.M. Royer, S.R. Das, Ad hoc on demand distance vector (AODV) routing, in: IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 90–100.

[3] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad-hoc wireless networks, Mobile Computing (1996) 153–181.

[4] V.D. Park, M.S. Corson, A higly adaptive distributed routing algorithm for mobile wireless networks, in: Proceedings IEEE Conference on Computer Communications (INFOCOM), Kobe, Japan, April 1997.

[5] S.A. Curtis, M.L. Rilee, P.E. Clark, G.C. Marr, Use of swarm intelligence in spacecraft constellations for the resource exploration of the asteroid belt, in: Proceedings 3rd International Workshop on Satellite Constellations and Formation Flying, Pisa, Italy, February 2003.

[6] M. Kifle, M. Andro, Q.K. Tran, G. Fujikawa, P.P. Chu, Toward a dynamically reconfigurable computing and communications system for small spacecraft, in: Proceedings of the 21st International Communication Satellite Systems Conference & Exhibit (ICSSC 2003), Yokohama, Japan, 2003.

[7] Y. Chen, M. Hadjitheodosiou, C. Cheung, Optimizing communications for constellation space missions, in: Proceedings 21st International Communication Satellite Systems Conference & Exhibit (ICSSC 2003), Yokohama, Japan, 2003.

[8] A. Doria, M. Uden, D.P. Pandey, Providing connectivity to the saami nomadic community, in: Proceedings 2nd International Conference on Open Collaborative Design for Sustainable Innovation, Bangalore, India, December 2002.

[9] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, H. Weiss, Delay-tolerant networking: An approach to interplanetary internet, IEEE Communications Magazine 41 (6) (2003) 128–137.

[10] A. Vahdat, D. Becker, Epidemic routing for partially connected ad hoc networks, Duke University, Tech. Rep. CS-200006, April 2000.

[11] M. Heusse, D. Snyers, Y. Kermarrec, Adaptive agent driven routing in communication networks: comparison with a classical approach, Advances in Complex Systems 2 (2000).

[12] M. Heusse, S. Guerin, D. Snyers, P. Kuntz, Adaptive agent-driven routing and load balancing in communication networks, Advances in Complex Systems 1 (1998) 234–257.

[13] X. Wang, F. Li, S. Ishihara, T. Mizuno, A multicast routing algorithm based on mobile multicast agents in ad-hoc networks, Special Issue on Internet Technology, IEICE Transactions on Communications E84-B (8) (2001).

[14] Y.J. Suh, H.S. Shin, D.H. Kwon, An efficient multicast routing protocol in wireless mobile networks, Wireless Networks 7 (2001) 443–453.

[15] R.R. Choudhury, S. Bandyopadhyay, K. Paul, A distributed mechanism for topology discovery in ad hoc wireless networks using mobile agents, in: Proceedings First Annual Workshop on Mobile Ad Hoc Networking Computing, MobiHOC, Boston, MA, USA, August 2000.

[16] R.R. Choudhury, K. Paul, S. Bandyopadhyay, MARP: a multi-agent routing protocol for mobile wireless ad hoc networks, Autonomous Agents and Multi-Agent Systems 8 (2004) 47–68.

[17] M. Dorigo, M. Birattari, T. Stützle, Ant colony optimization: artificial ants as a computational intelligence technique, IEEE Computational Intelligence Magazine 1 (4) (2006) 28–39.

[18] C. Blum, Ant colony optimization: introduction and recent trends, Physics of Life Reviews 2 (4) (2005) 353–373.

[19] M. Dorigo, 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.

[20] L.M. Gambardella, M. Dorigo, Solving symmetric and asymmetric tsps by ant colonies, in: Proceedings IEEE International Conference on Evolutionary Computation, Nagoya, Japan, May 1996, pp. 622–627.

[21] B. Bullnheimer, R.F. Hartl, C. Strauss, An improved ant system algorithm for the vehicle routing problem, Annals of Operations Research 89 (1999) 319–328.

[22] V. Maniezzo, A. Colorni, The ant system applied to the quadratic assignment problem, IEEE Transactions on Knowledge and Data Engineering 11 (5) (1999) 769–778.

[23] M. Bessedik, R. Laib, A. Boulmerka, H. Drias, Ant colony optimization and swarm intelligence, in: Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce Vol-1 (CIMCA-IAWTIC), 2005.

[24] G. Di Caro, M. Dorigo, Antnet: distributed stigmergetic control for communications networks, Journal of Artificial Intelligence Research 9 (1998) 317–365.

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

[26] S. Goss, S. Aron, J.L. Deneubourg, Self-organized shortcuts in the Argentine ant, Naturwissenschaften 76 (1989) 579–581.

[27] R. Schoonderwoerd, O. Holland, J. Bruten, Ant-like agents for load balancing in telecommunications networks, in: Proceedings 1st International Conference on Autonomous Agents, 1997, pp. 209–216.

[28] X. Zheng, W. Guo, R. Liu, An ant-based distributed routing algorithm for ad-hoc networks, in: International Conference on Communications, Circuits and Systems (ICCCAS), Chengdu, China, 27–29 June 2004.

[29] W. Krzysztof, Ant algorithm for flow assignment in connection-oriented networks, International Journal of Applied Mathematics and Computer Science 15 (2) (2005) 205–220.

[30] O. Hussein, T. Saadawi, Ant routing algorithm for mobile ad hoc networks (ARAMA), in: International Performance Computing and Communications Conference, IPCCC 2004, Phoenix, Arizona, 15–17 April 2004.

[31] T. White, B. Pagurek, F. Oppacher, ASGA: improving the ant system by integration with genetic algorithms, in: Genetic Programming 1998: Proceedings Third Annual Conference, University of Wisconsin, Madison, Wisconsin, USA, 22–25 July 1998, pp. 610–617.

[32] X. Jing, C. Liu, X. Sun, Artificial cognitive BP-CT ant routing algorithm, in: Proceedings International Joint Conference on Neural Networks, Montreal, Canada, July 31–August 4, 2005.

[33] M. Heusse, Y. Kermarrec, Adaptive routing and load balancing of ephemeral connections, in: Proceedings IEEE ECUMN, Colmar, France, 2000.

[34] X. Cheng, Y. Hou, A study of genetic ant routing algorithm, in: Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an, November 2003.

[35] M. Heissenbüttel, T. Braun, Ants-based routing in large scale mobile ad hoc networks, in: Proceedings Kommunikation in Verteilten Systemen (KiVS03), March 2003.

[36] J.S. Baras, H. Mehta, A probabilistic emergent routing algorithm for mobile ad hoc networks, in: Proceedings WiOpt'O3: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, March 2003.

[37] T. White, B. Pagurek, F. Oppacher, Connection management by ants: an application of mobile agents in network management, in: Proceedings International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA), Las Vegas, Nevada, USA, 1998.

[38] D. Subramanian, P. Druschel, J. Chen, Ants and reinforcement learning: a case study in routing in dynamic networks, in: Proceedings IJCAI-97, International Joint Conference on Artificial Intelligence, Nagoya, Japan, 23–29 August 1997.

[39] S. Marvaha, C. Tham, D. Srinivasan, Mobile agents based routing protocol for mobile ad hoc networks, in: Proceedings IEEE GLOBECOM, Taipei, Taiwan, 17–21 November 2002.

[40] S. Marvaha, C. Tham, D. Srinivasan, A novel routing protocol using mobile agents and reactive route discovery for ad-hoc wireless networks, in: Proceedings IEEE International Conference on Networks (ICON), Singapore, 27–30 August 2002.

[41] D. Cmara, A.A.F. Loureiro, A novel routing algorithm for hoc networks, Baltzer Journal of Telecommunications Systems 18 (1–3) (2001) 85–100.

[42] M. Berioli, L. Rosati, M. Werner, G. Reali, Ant routing concepts for dynamic meshed satellite constellations, in: Proceedings 6th European Workshop on Mobile/Personal Satcoms (EMPS) and 2nd Conference on Advanced Satellite Mobile Systems (ASMS), Noordwijk, The Netherlands, September 2004.

[43] L. Rosati, M. Berioli, G. Reali, On ant routing algorithms in ad-hoc networks, in: Proceedings IEEE Global Telecommunications Conference (GLOBECOM), San Francisco, CA, USA, 27 November–1 December 2006.

[44] M. Berioli, L. Rosati, G. Reali, Routing techniques for ad-hoc space networks, in: Proceedings 23rd International Communication Satellite Systems Conference (ICSSC) and 11th Ka and Broadband Communications Conference, Rome, Italy, September 2005.

[45] M. Berioli, L. Rosati, G. Reali, Ant routing algorithms for irregular satellite constellations, in: Proceedings Symposium on Performance Evaluation of Computer and

Telecommunication Systems 2005 (SPECTS05), Philadelphia, PA, USA, 24–28 July 2005.
[46] G. Di Caro, M. Dorigo, Antnet: a mobile agents approach to adaptive routing, Universit Libre de Bruxelles, Belgium, Tech. Rep. 12, 1997.
[47] Collaboration between researchers at UC Berkeley LBL USC/ISI and Xerox PARC, The ns Manual, The VINT Project, March 2003.
[48] S. Deering, R. Hinden, Internet Protocol, Version 6 (IPv6) Specification, RFC 2460, December 1998.
[49] T. Camp, J. Boleng, V. Davies, A survey of mobility models for ad hoc network research, Wireless Communication and Mobile Computing WCMC 2 (5) (2002) 483–502.
[50] C. Bettstetter, Mobility modeling in wireless networks: categorization, smooth movement, and border effects, SIGMOBILE Mobile Computing Communication Review 5 (3) (2001) 55–66.
[51] X. Hong, M. Gerla, G. Pei, C.C. Chiang, A group mobility model for ad hoc wireless networks, in: Proceedings Intern. Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, Seattle, USA, 1999.
[52] B. Liang, Z. Haas, Predictive distance-based mobility management for PCS networks, in: Proceedings IEEE Conference on Computer Communications (INFOCOM), New York, USA, March 1999.
[53] V. Tolety, Load reduction in ad hoc networks using mobile servers, Master's thesis, Colorado School of Mines, Colorado, USA, 1999.
[54] C.E. Perkins, Ad Hoc Networking, Addison-Wesley, 2001, March.
[55] C. Toh, Ad Hoc Mobile Wireless Networks: Protocols and Systems, Prentice-Hall, 2002, March.
[56] E. Royer, C. Toh, A review of current routing protocols for ad hoc mobile wireless networks, IEEE Personal Communications 6 (2) (1999) 46–55.
[57] C.-C. Chiang, H. Wu, W. Liu, M. Gerla, Routing in clustered multihop, mobile wireless networks with fading channel, in: IEEE Singapore International Conference on Networks (SICON), Singapore, 14–17 April 1997.
[58] S. Murthy, J.J. Garcia-Luna-Aceves, An efficient routing protocol for wireless networks, ACM Mobile Networks and Applications Journal, Special issue on Routing in Mobile Communication Networks (1996).
[59] Y. Ko, N.H. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, Wireless Networks 6 (4) (2000) 307–321.
[60] R. Sivakumar, P. Sinha, V. Bharghavan, CEDAR: a core-extraction distributed AdHoc routing algorithm, IEEE Journal on Selected Areas in Communications 17 (8) (1999).
[61] M. Joa-Ng, I. Lu, A GPS-based peer-to-peer hierarchical link state routing for mobile ad hoc networks, in: Vehicular Technology Conference (VTC), Tokyo, 2000.
[62] R.S. Sisodia, B.S. Manoj, C.R. Murthy, A preferred link based routing protocol for wireless ad hoc networks, Journal of Communication and Networks 4 (1) (2002) 14–21.
[63] P. Jacquet, P. Muhlethaler, A. Qayyum, Optimized link state routing (OLSR) protocol, Internet Draft `draft-ietf-manet-olsr-06.txt`, July 2002.
[64] N. Beijar, Zone Routing Protocol (ZRP), Networking Laboratory, Helsinki University of Technology, Licentiate course on Telecommunications Technology, Tech. Rep., 2002. <http://www.netlab.hut.fi/opetus/s38030/k02/Papers/08-Nicklas.pdf>.
[65] Z.J. Haas, M.R. Pearlman, P. Samar, The Intrazone Routing Protocol (IARP) for ad hoc networks, Internet Draft, July 2002, work in progress <http://www.ietf.org/proceedings/02nov/I-D/draft-ietf-manet-zone-iarp-02.txt>.
[66] Z.J. Haas, M.R. Pearlman, P. Samar, The Interzone Routing Protocol (IERP) for ad hoc networks, Internet Draft, July 2002, work in progress. <http://www.ietf.org/proceedings/02nov/I-D/draft-ietf-manet-zone-ierp-02.txt>.
[67] J. Broch, D.A. Maltz, D.B. Johnson, Y. Hu, J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: Proceedings MobiCom98, Dallas, Texas, USA, 25–30 October 1998.
[68] S.R. Das, R. Castaneda, J. Yan, Simulation based performance evaluation of mobile, ad hoc network routing protocols, ACM/Baltzer Mobile Networks and Applications (MONET) (7) (2000) 179–189.
[69] S.R. Chaudhry, A. Al-Khwildia, Y. Casey, H. Aldelou, H.S. Al-Raweshidy, A performance comparison of multi on-demand routing in wireless ad hoc networks, in: Proceedings IEEE International Conference on Wireless And Mobile Computing, Networking And Communications (WiMob), 22–24 August 2005.
[70] S.R. Das, C.E. Perkins, E.M. Royer, Performance comparison of two on-demand routing protocols for ad hoc networks, in: Proceedings IEEE Conference on Computer Communications (INFOCOM), Tel Aviv, Israel, 26–30 March 2000.

Laura Rosati received the Laurea degree in electronic engineering (*magna cum laude*), from the University of Perugia, Perugia, Italy, in 2003, where she is currently working toward the Ph.D. degree in information and electronic engineering.

Since 2003, she has been with the German Aerospace Center (DLR), Oberpfaffenhofen, Germany, in the Digital Networks Group of the Institute of Communications and Navigation. Her main research activities include routing on MANETs and resource allocation in hybrid terrestrial/satellite networks. She is currently involved in the IST Network of excellence SatNEx II.

Matteo Berioli received a Laurea degree in electronic engineering, and the Ph.D. degree in information engineering from the University of Perugia (Italy), in 2001 and 2005 respectively, both with honours; the title of the Ph.D. thesis was "MPLS and IP tunnels in Dynamic Satellite Networks". Since 2002, he is with the German Aerospace Center (DLR). His main research activities include QoS and protocol analysis in IP-based dynamic networks, with a focus on satellite systems and their integration with terrestrial networks; key research issues are cross-layer techniques, multicast, dynamic routing, packet-layer coding. In the last years he was involved in several European and ESA projects, and he has also worked as expert for the European Telecommunications Standards Institute (ETSI) in the area of broadband satellite multimedia; he is currently chairing the satellite working group of the PSCE Forum (Public Safety Communications Europe Forum). He is author/co-author of more than 30 papers, which appeared in international journals and conference proceedings. He has been a reviewer for technical international journals and for many international conferences.

Gianluca Reali is an associate professor of the Department of Information and Electronic Engineering (DIEI) of the University of Perugia since January 2005. He received the "Laurea" degree in Electronic Engineering from the University of Perugia in 1991, with honours. He received the Ph.D. degree in Telecommunications from the University of Perugia in 1997. From April 1997 to December 2004 we was researcher at DIEI. From August 1999 to January 2000 he was visiting researcher at the Computer Science Department of the University of California at Los Angeles (UCLA) USA. Currently G. REALI coordinates the research activities and related projects in the area of Telecommunication Networks done at DIEI. His past and current research activity (published on about 80 papers in peer-refereed international journals and conferences) spans several areas, including spread spectrum techniques, equalization of propagation effects in wireless mobile channels, resource allocation over circuit-switched satellite networks, design and performance evaluation of broadband and wireless networks, IP QoS techniques, routing over terrestrial and satellite networks, pricing strategies for guaranteed network services, delivery of multimedia services over packet networks. His research and professional activities includes collaborations with many italian and international universities, companies such as Alcatel-Alenia Space and Magneti Marelli, research centres such as CNR, Centro Ricerche Progetto San Marco, Telecom Italia Labs, Fondazione Ugo Bordoni. He has been involved in management activities for several national and international projects: partner coordinator for the European projects IST SUITED (2000–2002) and IST WHYLESS.COM (2001–2003), unit/Task/WP coordinator in the projects FIRB "PRIMO" (2003–2006) and PRIN "TWELVE" (2004–2006), research coordinator for DIEI in research projects involving local and national companies (e.g. Sogei, Telephonica (current TeleUnit s.p.a.), Space Software Italia). He has collaborated in many other national and international projects and networks of excellence, such as ACTS CABSINET, ACTS ASSET, IST Simplicity, IST SatNEx and SatNExII (still active NoE), FIRB Vicom, PRIN Ramon. He has served as Technical Program Committee member and as referee for several international IEEE/ACM journals and conferences. He also coordinates the research and implementation activities of the Telecommunication Networks Research laboratory of DIEI. In 2005 he has been a consultant of the Regione Umbria, acting as the responsible of networking support for the realization of a regional network of high-precision GPS/GNSS stations.