



# PPMA, a probabilistic predictive multicast algorithm for ad hoc networks <sup>☆</sup>

Dario Pompili <sup>a,\*</sup>, Marco Vittucci <sup>b</sup>

<sup>a</sup> *Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA*

<sup>b</sup> *Dipartimento di Informatica e Sistemistica, University "La Sapienza" of Rome, 00184, Italy*

Received 5 April 2005; accepted 5 September 2005

## Abstract

Ad hoc networks are collections of mobile nodes communicating using wireless media without any fixed infrastructure. Existing multicast protocols fall short in a harsh ad hoc mobile environment, since node mobility causes conventional multicast trees to rapidly become outdated. The amount of bandwidth resource required for building up a multicast tree is less than that required for other delivery structures, since a tree avoids unnecessary duplication of data. However, a tree structure is more subject to disruption due to link/node failure and node mobility than more meshed structures. This paper explores these contrasting issues and proposes PPMA, a Probabilistic Predictive Multicast Algorithm for ad hoc networks, that leverages the tree delivery structure for multicasting, solving its drawbacks in terms of lack of robustness and reliability in highly mobile environments. PPMA overcomes the existing trade-off between the bandwidth efficiency to set up a multicast tree, and the tree robustness to node energy consumption and mobility, by decoupling tree efficiency from mobility robustness. By exploiting the non-deterministic nature of ad hoc networks, the proposed algorithm takes into account the estimated network state evolution in terms of node residual energy, link availability and node mobility forecast, in order to maximize the *multicast tree lifetime*, and consequently reduce the number of costly tree reconfigurations. The algorithm statistically tracks the relative movements among nodes to capture the dynamics in the ad hoc network. This way, PPMA estimates the node future relative positions in order to calculate a *long-lasting* multicast tree. To do so, it exploits the most *stable* links in the network, while minimizing the total network *energy consumption*. We propose PPMA in both its *centralized* and *distributed* version, providing performance evaluation through extensive simulation experiments. © 2005 Elsevier B.V. All rights reserved.

**Keywords:** MANET; Ad hoc multicast; Route stability; Tree robustness; Mobility prediction; Performance

## 1. Introduction

Ad hoc networks are collections of mobile nodes communicating using wireless media without any fixed infrastructure. Conventional multicast routing protocols are inadequate in a harsh mobile environment, as mobility can cause rapid and frequent

<sup>☆</sup> A preliminary version of this paper was accepted for publication in the Proceedings of Med-Hoc-Net 2004, Bodrum, Turkey [14].

\* Corresponding author. Tel.: +1 404 894 6616.

E-mail addresses: [dario@ece.gatech.edu](mailto:dario@ece.gatech.edu) (D. Pompili), [marco.vittucci@alice.it](mailto:marco.vittucci@alice.it) (M. Vittucci).

changes in the network topology. Frequent state changes require constant updates, reducing the already limited bandwidth available for data, and possibly never converging to accurately portray the current topology. Mobility represents the most challenging issue to be addressed by multicast routing protocols. In fact, if a multicast protocol shows robustness to mobility, it often incurs in other shortcomings such as protocol overhead or loops. Conversely, if a protocol is primarily designed to limit or to optimize the network with respect to signaling overhead and power consumption, commonly its performance quickly degrades as the mobility increases.

Multicast communications can be classified into *source-specific* and *group-shared*. In *source-specific* multicast communication, only one node in the multicast group sends data, while all the other member nodes receive data. In *group-shared* multicast communication, each node in the multicast group wants to send/receive data to/from member nodes. A tree that spans all member nodes is called *multicast tree*. Multicast trees can be classified into *source rooted* and *shared* trees, according to the communication strategy. A *source-rooted* tree has the source node as root and is optimized for source-specific multicast communications. A *shared tree*, on the other hand, is optimized for group-shared communications, and connects each group member to all the other group members.

*Tree-based multicast* is a very well-established concept in wired networks, both for source-specific and group-shared application support [16]. In the tree-based approach, multicast routing uses a source-based or group-shared tree among sources and receivers, depending on the application requirements. This approach is characterized by high bandwidth efficiency, since only one path exists between any pair of nodes. The amount of bandwidth resource required for building up a tree is less than that required for other multicast delivery structures, since a multicast tree avoids unnecessary duplication of data [19]. This way, the optimization routing problem in tree-based multicast is to find the *minimum-weight tree* that spans all the nodes in the multicast group [1,13,18]. However, a multicast tree is more subject to disruption due to link/node failure and node mobility than more meshed structures.

This paper explores these contrasting issues and proposes PPMA, a Probabilistic Predictive Multicast Algorithm for ad hoc networks, that leverages the tree delivery structure for multicasting, solving its

drawbacks in terms of lack of robustness and reliability in highly mobile environments. There is, in fact, a trade-off between the bandwidth resource used to set up a multicast tree and the tree robustness to energy node consumption and mobility. The primary objective of the proposed algorithm is to address this trade-off, by decoupling tree efficiency from mobility robustness. The intuition this paper is based on is that the deterministic nature that characterizes traditional multicast protocols tends to become their limiting factor when aiming at robustness and scalability, especially in highly dynamic ad hoc networks. By exploiting the non-deterministic nature of ad hoc networks, PPMA takes into account the estimated network state evolution in terms of node residual energy, link availability and node mobility forecast, in order to maximize the *multicast tree lifetime*. The algorithm statistically tracks the relative movements among nodes to capture the dynamics in the ad hoc network. PPMA estimates the node future relative positions in order to calculate a *long-lasting* multicast tree. To do so, it exploits the most *stable* links in the network, while minimizing the total network *energy consumption*. We propose PPMA in both its *centralized* and *distributed* version, providing performance evaluation through extensive simulation experiments.

The remainder of the paper is organized as follows. In Section 2 we review the leading ad hoc multicast routing protocols, which the present work is related to. In Section 3 we describe the motivations and goals of this paper, and we introduce our novel probabilistic cost function. In Section 4 we explore the terms our cost function is based on, and point out how they achieve the described goals. In Section 5 we present PPMA, a Probabilistic Predictive Multicast Algorithm for ad hoc networks, in its centralized version, while in Section 6 the distributed version is described. In Section 7 we show numerical results through extensive simulation experiments. Finally, in Section 8 we conclude the paper.

## 2. Related work

There have been several multicast routing protocols proposed for ad hoc networks in the literature. In the following we present those which the present work is related to, focusing on three tree-based algorithms that face the problem of determining a robust and reliable multicast tree in mobile ad hoc networks. We will point out which are their strengths and weaknesses. In particular, in Sections

2.1–2.3 we present pros and cons of PAST-DM (Progressively Adapted Sub-Tree in Dynamic Mesh) [6], ITAMAR (Independent-Tree Ad hoc Multicast Routing) [17] and AODV (Ad hoc On-demand Distance Vector Protocol) [15], respectively. In Section 3 we will then describe how our proposed probabilistic multicast algorithm effectively addresses most of their drawbacks.

### 2.1. PAST-DM: progressively adapted sub-tree in dynamic mesh

PAST-DM [6] utilizes a *virtual mesh topology*, which has the advantage of scaling very well, since the virtual topology can hide the real network topology regardless of the network dimension. A multicast session begins with the construction of a virtual mesh connecting all group members. PAST-DM gradually adapts the virtual mesh to the changes of the underlying network topology in a fully distributed manner [7]. From the computed virtual mesh, a multicast tree for packet delivery is extracted and then progressively adjusted according to the latest local topology information. At each node, the topology map is represented as a link-state table. The entry of this table is the link-state information of the virtual network, e.g., the hop distance, which is acquired from the virtual neighbors. Through the link-state table, each node has a local view of the whole virtual topology. In PAST-DM, each source constructs its own multicast data delivery tree based on its local link-state table. In order to minimize the total multicast tree cost, each source needs to construct a Steiner tree [1], which is extracted from the virtual mesh. To this end, PAST-DM uses a source-based Steiner tree heuristic that relies on the definitions of *hop distance*, *cost* and *adapted cost* for each virtual link. *Hop distance* is the minimum distance in terms of number of hops from one of the two nodes of the virtual link to the source node; *cost* is a generic metric that characterizes the physical link which the virtual link is associated with; and *adapted cost* of a virtual link is the link cost multiplied by its hop distance to the source. Thus, for any virtual link that has the source node as one of its two nodes, both its distance value and adapted cost are zero.

PAST-DM computes an approximation of the Steiner tree by exploiting an heuristic, namely the source-based Steiner tree algorithm, that takes the defined adapted costs as key metric to approximate the Steiner tree. By applying this heuristic, in

PAST-DM the source makes all its neighbors as its children in the multicast tree, and divides the remaining nodes into subgroups. Each subgroup forms a sub-tree rooted at one of the first-level children of the source. Then, each child recursively repeats the source-based Steiner tree algorithm. Consequently, the source node does not need to compute the whole multicast tree; in fact, each child is responsible of further delivering the data packets to all nodes in its subgroup.

#### 2.1.1. Main disadvantages

PAST-DM does not explicitly take into account node mobility prediction in the computation of the adapted cost, which is a key metric for the construction of multicast trees. Consequently, weighting the link cost by a distance that is rapidly changing may result useless, or even incorrect, in a dynamic environment.

### 2.2. ITAMAR: independent-tree ad hoc multicast routing

ITAMAR [17] finds a set of pre-calculated alternate trees to quickly react to link failures. This way, delay could be reduced whenever a viable backup tree is available at the time of failure of the current tree. Moreover, backup trees have the advantage of making the tree-based scheme more robust to node mobility. The basic idea of this multicast routing scheme is that backup multicast trees with minimum number of common links are used to reduce the number of service interruptions. This reduces the mean time between route discoveries, limiting the control overhead and the rate of data loss. At the same time, ITAMAR aims to keep the cost of transmission low. It is worth noting that this method is effective only if the tree failure times are independent [17]. Thus, under the constraint that nodes move independently, multicast trees must have no common nodes, and hence no common edges, to be independent. Since totally independent trees may not be found in many cases, ITAMAR objective is to minimize the dependence between the failure times, i.e., the correlation of the failure times of the main tree and the backup tree.

#### 2.2.1. Main disadvantages

The main disadvantage of ITAMAR [17] is that minimizing the correlation between the main tree and the backup tree is a high-computation operation, and may not be convenient in all situations.

Also, preventing a link failure by pre-calculating a tree as much independent from the previous tree as possible, may result in a greater control messaging overhead, which is required for route establishment, than that used for repairing only the failed links while leaving all other tree nodes unchanged. Another disadvantage of ITAMAR is that it does not scale well since it requires every node to have a global knowledge of the network topology.

### 2.3. AODV: ad hoc on-demand distance vector protocol

AODV [15] routing protocol is capable of unicast, broadcast and multicast communications. One of the main advantages of combining in the same protocol unicast and multicast communication abilities is that route information obtained when searching for a multicast route can also increase unicast routing knowledge, and vice versa. Unicast and multicast routes are discovered on-demand using a *broadcast route-discovery* mechanism. In order to reduce communication overheads, updates are propagated only along active routes, i.e., routes that have been used in the recent past. Broadcast data delivery is provided by using the Source IP Address and Identification field of the IP header as a unique identifier of the packet. As nodes join their multicast group, a multicast tree composed of group members and nodes connecting the group members is created. A multicast-group leader maintains the multicast-group sequence number.

#### 2.3.1. Main disadvantages

There are some disadvantages in AODV protocol concerning latency, utilization efficiency, mobility robustness and scalability under particular conditions. As far as latency is concerned, since routes may differ from the shortest paths, the average data delivery latency is expected to be higher than in a shortest path algorithm. In terms of resource utilization efficiency, source routing utilizes a lot of bandwidth due to the use of lists of addresses that increase the size of the data packet headers. Moreover, since AODV keeps *hard states* in its routing table, the protocol has to actively track and react to changes in the tree. As far as mobility is concerned, AODV suffers high-rate mobility due to the transmission of many routing packets, since each node maintains a routing-table entry for each multicast group for which the node is a member or a router. Another disadvantage of AODV is that

storm of replies and repetitive updates in host caches may occur since nodes make use of their routing caches to reply to route queries. This leads to poor scalability performance.

## 3. Problem setup

### 3.1. Motivations and goals

The main motivations for introducing PPMA, a probabilistic predictive algorithm for multicasting in ad hoc networks, are summarized hereafter:

- The deterministic nature that characterizes traditional multicast protocols tends to become their limiting factor when aiming at robustness and scalability, especially in highly dynamic ad hoc networks.
- The amount of bandwidth resource required for building up a tree is less than that required for other more meshed multicast delivery structures.
- The trade-off between the bandwidth resource required by a multicast tree, and the tree robustness to energy node consumption and mobility has never been evaluated through extensive simulation experiments.

By exploiting the non-deterministic nature of ad hoc networks, PPMA takes into account the estimated network state evolution in terms of node residual energy, link availability and node mobility forecast, in order to maximize the *multicast tree lifetime*, and consequently reduce the number of costly tree reconfigurations. The algorithm statistically tracks the relative movements among nodes to capture the dynamics in the ad hoc network. This way, PPMA estimates the node future relative positions, and computes a *long-lasting* multicast tree. To do so, it exploits the most *stable* links in the network, while minimizing the total network *energy consumption*.

To target these goals, we individuate a set of *general rules* that aims to achieve such objectives:

1. The higher the battery charge a node avails, the higher its availability to take part in the tree should be.
2. The higher the number of multicast trees a node belongs to, the lower the node availability should be.
3. If the available battery charge goes under a pre-determined threshold  $\mathcal{E}^{\min}$ , then a node should

no more be considered available to take part in multicast communications.

4. The larger the distance between two nodes is, the smaller their availability to establish a communication should be. Obviously, if the distance is larger than a limit range  $D^{\max}$ , no link should be considered.
5. The more prone a link is to fail or break, the smaller its probability of being included in a branch of a multicast tree should be. Such a property clearly depends on the positions, speeds and directions of the nodes.

### 3.2. Transmission energy model

An accurate model for energy consumption per bit at the physical layer is  $E = E_{\text{elec}}^{\text{trans}} + \beta d^\gamma + E_{\text{elec}}^{\text{rec}}$ , where  $E_{\text{elec}}^{\text{trans}}$  is a *distance-independent* term that takes into account overheads of transmitter electronics (PLLs, VCOs, bias currents, etc.) and digital processing [8,9],  $E_{\text{elec}}^{\text{rec}}$  is a *distance-independent* term that takes into account the overhead of receiver electronics, while  $\beta d^\gamma$  accounts for the radiated power necessary to transmit one bit over a distance  $d$  between source and destination. As in [12], we assume that  $E_{\text{elec}}^{\text{trans}} = E_{\text{elec}}^{\text{rec}} = E_{\text{elec}}$ . Thus, the overall expression simplifies to  $E = 2E_{\text{elec}} + \beta d^\gamma$ , where

- $\gamma$  is the exponent of the path loss ( $2 \leq \gamma \leq 5$ );
- $\beta$  is a constant [ $\text{J}/(\text{bit m}^\gamma)$ ];
- $E_{\text{elec}}$  is the energy needed by the transceiver circuitry to transmit or receive one bit [ $\text{J}/\text{bit}$ ].

The values considered in this paper are reported in Table 2.

### 3.3. Probabilistic link cost

In order to synthesize the properties presented in Section 3.1, we identify a *probabilistic link cost function*, composed of three multiplicative terms, an *Energy Term*, a *Distance Term* and a *Lifetime Term*, which will be extensively explained in Sections 4.1–4.3, respectively. For two generic nodes  $i$  and  $j$ , we consider as their link cost  $C_{ij}$  the following cost function:

$$C_{ij} = -\log(P_i^\mathcal{E} \cdot P_{ij}^\mathcal{D} \cdot P_{ij}^\mathcal{L}), \quad (1)$$

where

$$P_i^\mathcal{E} = P_i^\mathcal{E}(\mathcal{E}_i, \mathcal{E}^{\min}, \mathcal{E}^{\max}, \mathcal{W}_i^{\text{out}}, \mathcal{W}_i^{\max}) \quad (2)$$

is the *Energy Term* for node  $i$  and it weights how much residual energy node  $i$  avails for communications;

$$P_{ij}^\mathcal{D} = P_{ij}^\mathcal{D}(d_{ij}, D_i^{\max}[r^{\text{req}}, \epsilon_b^{\text{mod}}, \eta_B^{\text{mod}}], \gamma) \quad (3)$$

is the *Distance Term* between node  $i$  and node  $j$ , and it takes into account the transmission power needed by node  $i$  to communicate with node  $j$ ; and

$$P_{ij}^\mathcal{L} = P_{ij}^\mathcal{L}(d_{ij}, D_i^{\max}[r^{\text{req}}, \epsilon_b^{\text{mod}}, \eta_B^{\text{mod}}], \sigma_{d_{ij}}, \Delta t) \quad (4)$$

is the *Lifetime Term* between node  $i$  and node  $j$ , and it statistically evaluates the probability that the distance between these two nodes remains bounded for  $\Delta t$  seconds by the maximum transmission range of node  $i$ ,  $D_i^{\max}$ , given the current distance. The explanation of the parameters and variables each term depends on, as well as their units, are reported hereafter for the reader's convenience:

$\mathcal{E}_i$  [J] is the battery state of node  $i$  (residual charge);  $\mathcal{E}^{\min}$  [J] is the energy threshold under which every node is no more available, and  $\mathcal{E}^{\max}$  [J] is the maximum charge among nodes;

$\mathcal{W}_i^{\text{out}}$  [W] is the power spent for the ongoing multicast communications by node  $i$ , and  $\mathcal{W}_i^{\max}$  [W] is the maximum power node  $i$  can consume to communicate;

$d_{ij}$  [m] is the distance between node  $i$  and node  $j$  at time  $t$  [s], and  $D_i^{\max}[r^{\text{req}}, \epsilon_b^{\text{mod}}, \eta_B^{\text{mod}}]$  [m] is the maximum radio range node  $i$  can reach while guaranteeing a bitrate  $r^{\text{req}}$  [Kbps], given the energy-per-bit  $\epsilon_b^{\text{mod}}$  [J/bit] used in radio transmission, and the spectral efficiency  $\eta_B^{\text{mod}}$  [bit/s/Hz] of the adopted modulation scheme;

$\beta$  and  $\gamma$  are parameters that depend on the environment;  $\Delta t$  [s] is the time interval used for mobility prediction, and  $\sigma_{d_{ij}}$  [m] is the standard deviation of the gaussian process used in the prediction of the distance evolution between nodes  $i$  and  $j$ .

All terms are normalized, so that they range in  $[0, 1]$ , and can be viewed as *pseudo-probability* terms. Consequently, the link cost function  $C_{ij}$  in (1) can be viewed as a pseudo-probability as well. Actually, only the *Lifetime Term*  $P_{ij}^\mathcal{L}$  is a correctly defined probability, while the other terms are normalized so as to maintain the  $C_{ij}$  values in the same variation range. This way, because of the statistical meaning of the link cost  $C_{ij}$ , we can associate a probability metric to the multicast trees computed by the

proposed probabilistic predictive algorithm. The *tree probability* is defined as the sum of all the costs of the links included in the tree. This tree probability is obviously expected to decrease over time, if nodes are not stationary and if their future movements are unknown. The tree probability can be interpreted as the probability that all the links in a multicast tree will survive at least for a time period  $\Delta t$ , which is a critical parameter in (4). In Section 4 we describe in detail all the terms of the link cost function in (1), why each of them is necessary, and their synergic effect to meet the goals outlined in Section 3.1.

#### 4. Probabilistic link cost function terms

In this section we describe the properties of each term that is argument of the link cost function in (1). In particular, in Section 4.1 we detail the Energy Term in (2), in Section 4.2 we present the Distance Term in (3) and in Section 4.3 we derive the Lifetime Term in (4).

##### 4.1. Energy Term

The main purpose of the *Energy Term* is to keep all the nodes of the network alive as long as possible, while respecting the general rules (1)–(3), presented in Section 3.1. Also, we want to interpret the Energy Term associated with node  $i$  as the probability of choosing node  $i$  during the multicast tree construction. So, the Energy Term varies in the range  $[0, 1]$  and is a function of the battery state of the node, i.e., the residual charge. Specifically, it assumes greater values for those nodes that have a greater residual charge. The final expression for the Energy Term is reported in (8), while in the following we provide the formal derivation of it. Given  $\mathcal{V}$  the set of indexes that correspond to the nodes in the network, a first attempt for the Energy Term associated with node  $i$  could be

$$P_i(\mathcal{E}_i) = \left( \frac{\mathcal{E}_i - \min_{n \in \mathcal{V}} \{\mathcal{E}_n^{\min}\}}{\max_{n \in \mathcal{V}} \{\mathcal{E}_n^{\max}\} - \min_{n \in \mathcal{V}} \{\mathcal{E}_n^{\min}\}} \right) \cdot u(\Delta_{\mathcal{E}_i}^{\min}), \quad (5)$$

where  $\Delta_{\mathcal{E}_i}^{\min} = \mathcal{E}_i - \min_{n \in \mathcal{V}} \{\mathcal{E}_n^{\min}\}$  and the unit step  $u(\Delta_{\mathcal{E}_i}^{\min})$  is defined as

$$u(\Delta_{\mathcal{E}_i}^{\min}) = \begin{cases} 1 & \Delta_{\mathcal{E}_i}^{\min} \geq 0, \\ 0 & \Delta_{\mathcal{E}_i}^{\min} < 0. \end{cases} \quad (6)$$

Roughly,  $P_i(\mathcal{E}_i)$  can be viewed as the percentage charge of node  $i$ , with respect to the maximum possible charge of nodes, i.e.,  $\max_{n \in \mathcal{V}} \{\mathcal{E}_n^{\max}\}$ . The reason for the presence of the operators  $\min$  and  $\max$  in (5) is that we want to associate with the Energy Term of a node its physical residual charge. In fact, if we designed  $P_i(\mathcal{E}_i)$  as the percentage charge of the maximum charge of node  $i$ , we would miss the objective of maximizing the lifetime of all the nodes in the network.

This can be better understood through an example. Fig. 1 shows  $P_i(\mathcal{E}_i)$  and  $P_j(\mathcal{E}_j)$  associated with node  $i$  and node  $j$ , respectively, according to the following definition for the generic node  $k$ , where  $\min$  and  $\max$  operators are used;

$$P_k(\mathcal{E}_k) = \left( \frac{\mathcal{E}_k - \mathcal{E}_k^{\min}}{\mathcal{E}_k^{\max} - \mathcal{E}_k^{\min}} \right) \cdot u(\mathcal{E}_k - \mathcal{E}_k^{\min}), \quad (7)$$

where we chose  $\mathcal{E}_i^{\max} = 3$  [J],  $\mathcal{E}_j^{\max} = 5$  [J] and  $\mathcal{E}_i^{\min} = \mathcal{E}_j^{\min} = 0$  [J], for the sake of clarity. We note that (7) coincides with the percentage charge of the total charge of node  $k$ . Also, we note that node  $j$  has a larger total charge  $\mathcal{E}_j^{\max}$  than node  $i$ , and that, consequentially,  $P_j(\mathcal{E}_j)$  has a smaller slope than  $P_i(\mathcal{E}_i)$ . If node  $i$  and node  $j$  had equal charge  $\mathcal{E}^*$ , their probabilities in (7) would be different, since  $\mathcal{E}_i^{\max} \neq \mathcal{E}_j^{\max}$ . This way, the charges of the two nodes would not be equally weighted. In particular, node  $i$  would be more likely to join multicast trees than  $j$ , although equipped with a smaller energy supply. This simple example stresses the need for the operators  $\min$  and  $\max$  in (5).

Let us point out a drawback in (5): if a new node  $h$ , equipped with an energy supply  $\mathcal{E}_h^{\max}$  higher than

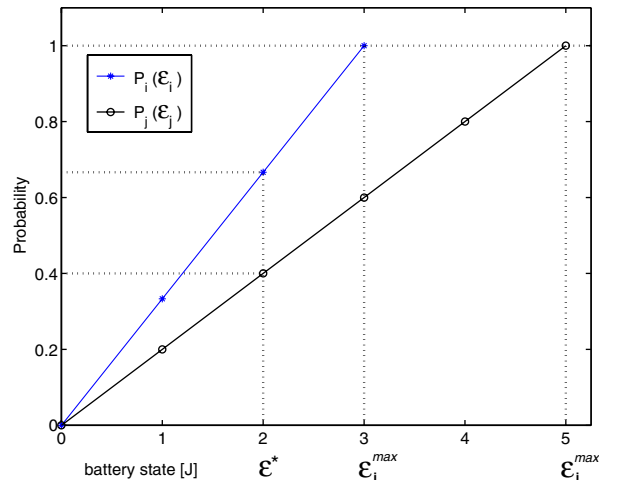


Fig. 1. Example of the *wrong* Energy Term expression in (7).

any other node in the network, joined the network, we should redefine (5) by replacing  $\max_{n \in \mathcal{V}} \{\mathcal{E}_n^{\max}\}$  with  $\mathcal{E}_h^{\max}$ , where the max operator is extended only to the already deployed nodes. Such a problem could be avoided either by replacing  $\max_{n \in \mathcal{V}} \{\mathcal{E}_n^{\max}\}$  with an a priori-known  $\mathcal{E}^{\max}$ , or with a value large enough to hold for all the nodes of the network. If the battery charge of node  $i$  becomes lower than its  $\mathcal{E}_i^{\min}$ , such a node cannot join multicast trees anymore; consequently, the term must be equal to zero. Otherwise, if node  $i$  is not currently involved in any multicast communication, its availability linearly depends on its charge  $\mathcal{E}_i$ . Thus, all other parameters being equal, we choose the most charged nodes among the possible ones. Conversely, if a node is involved in some multicast communications, its availability should be less than the previous case. Moreover, the higher the number of multicast paths a node shares, the lower its availability should be, according to rule (2) in Section 3.1. This behavior can be ensured by adding an appropriate exponent to (5). Therefore, the final *Energy Term* is

$$P_i^{\mathcal{E}} = \left[ \frac{\mathcal{E}_i - \min_{n \in \mathcal{V}} \{\mathcal{E}_n^{\min}\}}{\max_{n \in \mathcal{V}} \{\mathcal{E}_n^{\max}\} - \min_{n \in \mathcal{V}} \{\mathcal{E}_n^{\min}\}} \right]^{\frac{\mathcal{W}_i^{\max}}{\mathcal{W}_i^{\max} - \mathcal{W}_i^{\text{out}} + \epsilon}} u(\Delta_{\mathcal{E}_i}^{\min}), \quad (8)$$

where  $\epsilon$  is a positive constant close to 0 whose purpose is to avoid a possible division by zero, and  $\Delta_{\mathcal{E}_i}^{\min} = \mathcal{E}_i - \min_{n \in \mathcal{V}} \{\mathcal{E}_n^{\min}\}$ .

The exponent in (8) is an adimensional term, and it increases from 1 to  $\infty$  when  $\mathcal{W}_i^{\text{out}}$  increases from 0 to  $\mathcal{W}_i^{\max}$ . Thus, as  $\mathcal{W}_i^{\text{out}}$  increases to  $\mathcal{W}_i^{\max}$ , the Energy Term tends to a *two-slope* broken line. In particular, it has slope equal to zero for  $\mathcal{E}_i \in [\mathcal{E}^{\min}, \mathcal{E}^{\max}[$ , and slope equal to  $\infty$  in  $\mathcal{E}^{\max}$ . In this limit case the term is always equal to zero but in  $\mathcal{E}^{\max}$ .

Fig. 2 shows how the Energy Term changes as  $\mathcal{W}_i^{\text{out}}$  increases from 0 to  $\mathcal{W}_i^{\max}$ , where the constants used in (8) can be found in Table 2. Since the Energy Term tends to a two-slope broken line as  $\mathcal{W}_i^{\text{out}}$  tends to  $\mathcal{W}_i^{\max}$ , we can note that the more instantaneous power is spent in multicast communications by node  $i$ , the less node  $i$  is available to be part of other multicast communications.

#### 4.2. Distance Term

The Distance Term synthesizes the general rule (4) in Section 3.1. It takes into account how much

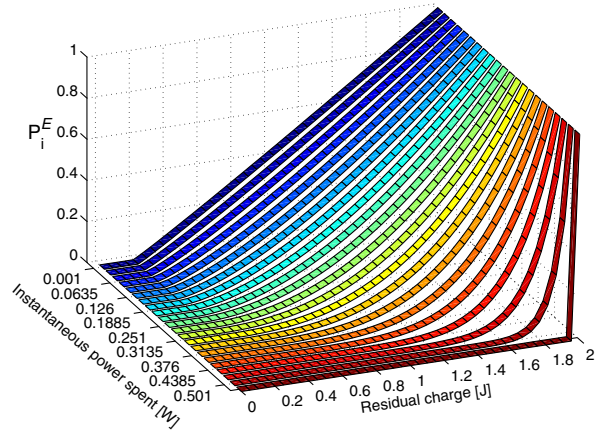


Fig. 2. Energy Term for different values of the instantaneous power spent in multicast communications.

power will be spent by node  $i$  to maintain link  $(i, j)$  in the multicast tree. An essential piece of information to be known is an estimate of the current distance  $d_{ij}(t)$  between node  $i$  and node  $j$ , hereon indicated as  $\overline{d_{ij}(t)}$ . The Distance Term can be expressed as

$$P_{ij}^{\mathcal{D}} = \left( \frac{D_i^{\max} - \overline{d_{ij}(t)}}{D_i^{\max}} \right)^{\gamma} \cdot u(D_i^{\max} - \overline{d_{ij}(t)}), \quad (9)$$

where  $D_i^{\max} = D_i^{\max}[r^{\text{req}}, \epsilon_b^{\text{mod}}, \eta_B^{\text{mod}}], \forall i \in \mathcal{V}$ , is the maximum radio range node  $i$  can reach, which depends on the requested bitrate ( $r^{\text{req}}$ ), and on the spectral efficiency ( $\epsilon_b^{\text{mod}}$ ) and energy-per-bit ( $\eta_B^{\text{mod}}$ ) characterizing the used modulation scheme. In particular, the higher the requested bitrate, the shorter  $D_i^{\max}$  [4,5].

Clearly, if  $\overline{d_{ij}(t)} > D_i^{\max}$ , the communication link  $(i, j)$  does not respect the QoS requirements in terms of requested bitrate  $r^{\text{req}}$ . Thus, in this case,  $P_{ij}^{\mathcal{D}}$  must be 0. As far as the unit step function  $u(\cdot)$  in (9) is concerned, it is needed to ensure  $P_{ij}^{\mathcal{D}}$  not be negative for a distance  $\overline{d_{ij}(t)}$  greater than  $D_i^{\max}$ . All these issues explain why  $P_{ij}^{\mathcal{D}}$  decreases from 1 to 0 as  $\overline{d_{ij}(t)}$  increases from 0 to  $D_i^{\max}$ . In particular, this decrease is characterized by at least a quadratic trend. In fact, the received power decreases over distance with an exponent equal to  $\gamma$ , according to the propagation model in Section 3.2. In the free space  $\gamma$  is equal to 2, while in a real indoor or outdoor environment it ranges in [3,5].

In Fig. 3 the Distance Term for different values of  $D_i^{\max}$  in the range [50,200] m is depicted. It is interesting noting that the larger the maximum radio

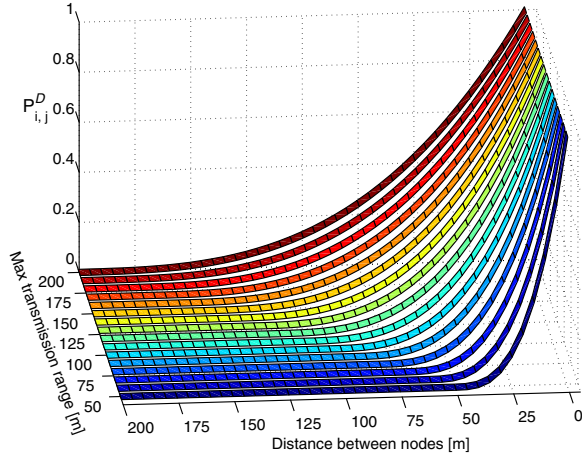


Fig. 3. Distance Term for different transmission ranges.

range  $D_i^{\max}$ , the smoother the decrease of the Distance Term when the distance between two nodes increases. Conversely, for low values of  $D_i^{\max}$ , e.g., 50 m, the Distance Term abruptly tends to zero as the distance between two nodes approaches the maximum radio range. As will be clearer later on, for each term modeling the link cost in (1), the more graceful the decrease, the slower the degradation of the link cost itself, and the better the performance achieved by our algorithm.

### 4.3. Lifetime Term

The Lifetime Term synthesizes the general rule (5) in Section 3.1. It has a predictive role and is defined as the probability that the maximum predicted distance  $d_{ij}^* = \text{MAX}_{\mathbf{V}_{ij}}(d_{ij}(t + \Delta t))$  between node  $i$  and node  $j$  be smaller or equal than  $D_i^{\max}$ , after  $\Delta t$  seconds elapsed from time instant  $t$ .  $\mathbf{V}_{ij}$  is the relative velocity between node  $i$  and node  $j$  and  $D_i^{\max}$  is the maximum distance transmitter  $i$  can reach, as introduced in Section 4.2. Differently from [10,11], we consider the following probability:

$$\text{Prob}\{d_{ij}^* \leq D_i^{\max} | d_{ij}^* = \text{MAX}_{\mathbf{V}_{ij}}(d_{ij}(t + \Delta t))\}. \quad (10)$$

We assume that  $d_{ij}(t + \Delta t)$  is a stochastic variable, obtained by summing two other stochastic variables, namely the distance,  $d_{ij}(t)$ , and the relative velocity between the nodes,  $\mathbf{V}_{ij}$ , i.e.,

$$d_{ij}(t + \Delta t) = d_{ij}(t) + \mathbf{V}_{ij}(t) \cdot \Delta t. \quad (11)$$

The maximum predicted distance of  $d_{ij}(t + \Delta t)$ ,  $d_{ij}^*$ , is computed in the worst case of node mobility, i.e., the relative velocity vector lies on the line con-

necting nodes  $i$  and  $j$ , and it is oriented such that the nodes move apart from each other. Hence, we can replace in (11) the relative velocity  $\mathbf{V}_{ij}$  with its norm  $|\mathbf{V}_{ij}|$ , which we refer to as *relative speed*,

$$d_{ij}^* = d_{ij}(t) + |\mathbf{V}_{ij}(t)| \cdot \Delta t. \quad (12)$$

Consequently, the Lifetime Term is

$$P_{ij}^{\mathcal{L}} = \text{Prob}\{(d_{ij}(t) + |\mathbf{V}_{ij}(t)| \cdot \Delta t) \leq D_i^{\max}\}. \quad (13)$$

The determination of the Lifetime Term is a challenging issue, because it generally depends on several factors. It could be well determined only if the current positions and the future destinations of nodes were exactly known, as well as their trajectories. Furthermore, a closed-form expression of  $P_{ij}^{\mathcal{L}}$  is hard to be found when all possible statistical parameters are taken into account, even under rough approximations of node mobility. Consequently, some assumptions about movements of nodes are necessary in order to obtain a useful expression of  $P_{ij}^{\mathcal{L}}$ . In the following, we describe the Probability Density Functions (PDFs) of  $d_{ij}(t)$  and  $|\mathbf{V}_{ij}|$ , which are used to compute the Lifetime Term as in (13).

To determine the PDF of  $d_{ij}(t)$ , it is worth pointing out that its measured value,  $\overline{d_{ij}(t)}$ , may be prone to errors. In most cases, this error can be accurately described by a gaussian PDF  $\mathcal{N}(\overline{d_{ij}(t)}, \sigma_{d_{ij}}^2)$ , centered at  $\overline{d_{ij}(t)}$  and with variance  $\sigma_{d_{ij}}^2$ . Because distances are not negative, we propose to weight the gaussian PDF by a unit step function  $u(z)$ , which is defined in (6), i.e.,

$$P_{d_{ij}(t)}(z) = \frac{1}{\sqrt{2\pi} \cdot \sigma_{d_{ij}}} \cdot e^{-\frac{1}{2} \left( \frac{z - \overline{d_{ij}(t)}}{\sigma_{d_{ij}}} \right)^2} \cdot u(z). \quad (14)$$

The variance  $\sigma_{d_{ij}}^2$  can be either set to a fixed value, which depends on the measurement system (as in the case of nodes equipped with a Global Position System (GPS) capabilities) or to a value proportional to  $\overline{d_{ij}(t)}$  (as in the case of nodes capable of measuring relative distances by message exchanging). Note that (14) is not a well-defined PDF, since its integral between  $-\infty$  and  $\infty$  is not equal to 1. Thus, we normalize (14) to 1, by means of a function  $\mathcal{A}(\overline{d_{ij}(t)}, \sigma_{d_{ij}})$  that represents the area of  $P_{d_{ij}(t)}(z)$  over the positive axis multiply by  $\sqrt{2\pi} \cdot \sigma_{d_{ij}}$ . Hence, (14) can be rewritten as

$$P_{d_{ij}(t)}(z) = \frac{1}{\mathcal{A}(\overline{d_{ij}(t)}, \sigma_{d_{ij}})} \cdot e^{-\frac{1}{2} \left( \frac{z - \overline{d_{ij}(t)}}{\sigma_{d_{ij}}} \right)^2} \cdot u(z), \quad (15)$$

where

$$\begin{aligned} \mathcal{A}(\overline{d_{ij}(t)}, \sigma_{d_{ij}}) &= \int_0^\infty e^{-\frac{1}{2} \left( \frac{w - \overline{d_{ij}(t)}}{\sigma_{d_{ij}}} \right)^2} dw \\ &= \sqrt{\frac{\pi}{2}} \sigma_{d_{ij}} \cdot \left[ 1 + \operatorname{erf} \left( \frac{\overline{d_{ij}(t)}}{\sqrt{2} \cdot \sigma_{d_{ij}}} \right) \right] \end{aligned} \quad (16)$$

and

$$\begin{aligned} \operatorname{erf}(\Gamma) &= \frac{2}{\sqrt{\pi}} \int_0^\Gamma e^{-t^2} dt \cong \frac{2}{\sqrt{\pi}} \Gamma - \frac{2}{3\sqrt{\pi}} \Gamma^3 \\ &\quad + \frac{1}{5\sqrt{\pi}} \Gamma^5 + o(\Gamma^7). \end{aligned} \quad (17)$$

Fig. 4 shows the PDF of  $d_{ij}(t)$  as in (15) for different values of the measured distance  $\overline{d_{ij}(t)}$  between nodes  $i$  and  $j$ .

To derive an expression of the relative speed,  $|\mathbf{V}_{ij}|$ , some assumptions about node mobility are required. We assume a zero mean gaussian PDF for each  $x$ - and  $y$ -component of the velocity vector  $\mathbf{V}_i = V_{x_i} \hat{x} + V_{y_i} \hat{y}$ , characterizing node  $i$ , and  $\mathbf{V}_j = V_{x_j} \hat{x} + V_{y_j} \hat{y}$  characterizing node  $j$ . This is equivalent to assuming a Brownian motion for each node around its current position in the time interval  $\Delta t$ . So, for the generic node  $i$  we have,

$$P_{V_{x_i} V_{y_i}}(v_{x_i}, v_{y_i}) = \frac{1}{2\pi \cdot \sigma_i^2} \cdot e^{-\frac{1}{2} \left( \frac{\sqrt{v_{x_i}^2 + v_{y_i}^2}}{\sigma_i} \right)^2}, \quad (18)$$

where  $\sigma_i^2$  is the variance of the conjuncted stochastic processes  $V_{x_i}$  and  $V_{y_i}$ . Consequently, the amplitude of the velocity vector  $|\mathbf{V}_i| = \sqrt{V_{x_i}^2 + V_{y_i}^2}$ , which we refer to as *speed* of node  $i$ , has a Rayleigh PDF,

$$P_{|\mathbf{V}_i|}(x) = \frac{x}{\sigma_i^2} \cdot e^{-\frac{x^2}{2\sigma_i^2}} \cdot u(x). \quad (19)$$

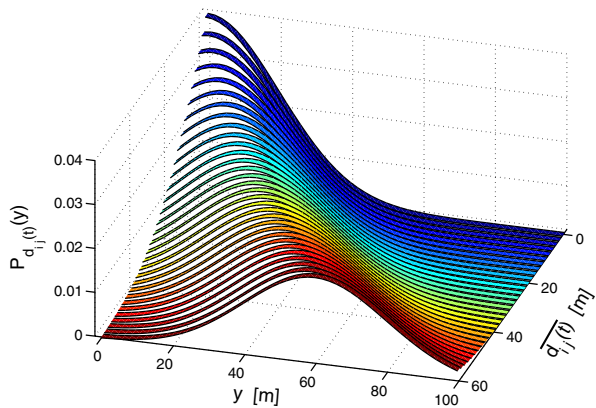


Fig. 4. PDF as in (15) for different distances between the nodes.

Let us now formally introduce the relative velocity  $\mathbf{V}_{ij} = V_{x_{ij}} \hat{x} + V_{y_{ij}} \hat{y}$  between nodes  $i$  and  $j$ , whose  $x$ - and  $y$ -components are

$$\begin{cases} V_{x_{ij}} = V_{x_i} - V_{x_j}, \\ V_{y_{ij}} = V_{y_i} - V_{y_j}. \end{cases} \quad (20)$$

Since  $V_{x_i}, V_{y_i}, V_{x_j}$  and  $V_{y_j}$  are gaussian stochastic variables,  $V_{x_{ij}}$  and  $V_{y_{ij}}$  are still gaussian stochastic variables. Furthermore, by assuming the  $x$ - and  $y$ -components of the node velocity vector statistically independent, it results that  $V_{x_{ij}}$  and  $V_{y_{ij}}$  are statistically independent too. Hence,  $V_{x_{ij}}$  has mean equal to the difference of the means of  $V_{x_i}$  and  $V_{x_j}$ , and variance equal to the sum of the variances of  $V_{x_i}$  and  $V_{x_j}$ . The same holds for  $V_{y_{ij}}$ . Thus, the conjuncted probability is

$$P_{V_{x_{ij}} V_{y_{ij}}}(x, y) = \frac{1}{2\pi \cdot (\sigma_i^2 + \sigma_j^2)} \cdot e^{-\frac{1}{2} \left( \frac{\sqrt{x^2 + y^2}}{\sqrt{\sigma_i^2 + \sigma_j^2}} \right)^2}. \quad (21)$$

Since  $V_{x_{ij}}$  and  $V_{y_{ij}}$  are zero mean gaussian variables,  $|\mathbf{V}_{ij}|$  has a Rayleigh PDF. So, taking  $\sigma_{|\mathbf{V}_{ij}|}^2 = \sigma_i^2 + \sigma_j^2$ , it yields,

$$P_{|\mathbf{V}_{ij}|}(x) = \frac{x}{\sigma_{|\mathbf{V}_{ij}|}^2} \cdot e^{-\frac{x^2}{2\sigma_{|\mathbf{V}_{ij}|}^2}} \cdot u(x). \quad (22)$$

Finally, we need to find an expression for  $\sigma_{|\mathbf{V}_{ij}|}$ . A reasonable assumption is that

$$E\{|\mathbf{V}_{ij}|\} = \overline{|\mathbf{V}_{ij}|}, \quad (23)$$

i.e., the mean expected value of (22),

$$E\{|\mathbf{V}_{ij}|\} = \frac{\sqrt{2\pi}}{2} \cdot \sigma_{|\mathbf{V}_{ij}|}, \quad (24)$$

coincides with the measured mean value of  $|\mathbf{V}_{ij}|$ ,

$$\overline{|\mathbf{V}_{ij}|} = \frac{\overline{d_{ij}(t)} - \overline{d_{ij}(t - \Delta\tau_{ij})}}{\Delta\tau_{ij}}, \quad (25)$$

where  $\Delta\tau_{ij}$  is the time interval between the current distance measure  $\overline{d_{ij}(t)}$  and the last distance measure  $\overline{d_{ij}(t - \Delta\tau_{ij})}$ . So,  $\sigma_{|\mathbf{V}_{ij}|}$  can be written as

$$\sigma_{|\mathbf{V}_{ij}|} = \sqrt{\frac{2}{\pi}} \cdot \left( \frac{\overline{d_{ij}(t)} - \overline{d_{ij}(t - \Delta\tau_{ij})}}{\Delta\tau_{ij}} \right). \quad (26)$$

The final expression of the Lifetime Term  $P_{ij}^{\mathcal{L}}$  can be found by integrating (15) and (22), accordingly to (13) (the rigorous derivation can be found in Appendix A). Variables  $\Delta t$  and  $\sigma_{|\mathbf{V}_{ij}|}$  appear always in a product form, so that they actually represent

a single variable, which will be simply indicated as  $\sigma_{\Delta_{ij}} = \Delta t \cdot \sigma_{|v_{ij}|}$ . Hence,

$$P_{ij}^{\mathcal{L}} = \frac{\operatorname{erf}\left(\frac{D_i^{\max} - \overline{d_{ij}(t)}}{\sqrt{2} \cdot \sigma_{d_{ij}}}\right) + \operatorname{erf}\left(\frac{\overline{d_{ij}(t)}}{\sqrt{2} \cdot \sigma_{d_{ij}}}\right)}{1 + \operatorname{erf}\left(\frac{\overline{d_{ij}(t)}}{\sqrt{2} \cdot \sigma_{d_{ij}}}\right)} \cdot \frac{\sigma_{\Delta_{ij}} e^{-\frac{1}{2} \left(\frac{D_i^{\max} - \overline{d_{ij}(t)}}{\sqrt{\sigma_{d_{ij}}^2 + \sigma_{\Delta_{ij}}^2}}\right)^2}}{\sqrt{\sigma_{d_{ij}}^2 + \sigma_{\Delta_{ij}}^2} \cdot \left[1 + \operatorname{erf}\left(\frac{\overline{d_{ij}(t)}}{\sqrt{2} \cdot \sigma_{d_{ij}}}\right)\right]} \cdot \left[ \operatorname{erf}\left(\frac{\sigma_{\Delta_{ij}}}{\sqrt{2} \sigma_{d_{ij}}} \frac{(D_i^{\max} - \overline{d_{ij}(t)})}{\sqrt{\sigma_{d_{ij}}^2 + \sigma_{\Delta_{ij}}^2}}\right) + \operatorname{erf}\left(\frac{D_i^{\max} \sigma_{d_{ij}}^2 - \overline{d_{ij}(t)} \sigma_{\Delta_{ij}}^2}{\sqrt{2} \sigma_{d_{ij}} \sigma_{\Delta_{ij}} \sqrt{\sigma_{d_{ij}}^2 + \sigma_{\Delta_{ij}}^2}}\right) \right]. \quad (27)$$

Although the expression in (27) may seem cumbersome, it can be easily computed if the erf function is approximated by expanding it with the Maclaurin series, as in (17).

Fig. 5(a) shows the Lifetime Term for different values of  $\overline{d_{ij}(t)}$  in [0, 90] m and  $\Delta t$  in [0, 40] s, with  $\sigma_{d_{ij}} = 20$  m,  $\sigma_{|v_{ij}|} = 1$  m/s and  $D_i^{\max} = 60$  m. Fig. 5(b) shows the Lifetime Term for different values of  $\sigma_{d_{ij}}$  in [0, 60] m and  $\Delta t$  in [0, 40] s, with  $\sigma_{|v_{ij}|} = 1$  m/s,  $D_i^{\max} = 60$  m and  $\overline{d_{ij}(t)} = 30$  m. In both figures, it can be seen how the Lifetime Term decreases as  $\Delta t$  increases. In fact, if the time interval elapsed from the current estimate  $\overline{d_{ij}(t)}$  is large, the mobility prediction based on such estimate may have poor significance. Moreover, it is worth noting that values of the Lifetime Term close to 1 correspond to low distances between the nodes, low values of  $\sigma_{d_{ij}}$  and low values of relative speed, i.e., when the nodes are likely to remain close.

## 5. Centralized PPMA

Algorithm 1 represents the pseudo-code for PPMA, the proposed probabilistic predictive multicast algorithm, in its centralized version. It extends the centralized Bellman–Ford algorithm [3], whose objective is to compute the shortest path from node  $x$  to source  $s$ , with respect to a certain metric, for every node  $x$  in the network. The centralized Bellman–Ford algorithm builds up a spanning tree that has source  $s$  as its root. It chooses the father  $f_x$  of a node  $x$  by minimizing the cost of the associated path toward  $s$ . The multicast tree connecting the source node with the receiver nodes is extracted from the computed spanning tree. This procedure leads in general to several unicast paths, which inefficiently connect the sender to the receivers with no bandwidth saving. On the other hand, the spanning tree has the good property that it does not distinguish the receiver nodes from the other nodes. This property can be fruitfully used if some nodes want to become a receiver, since a path from the new-member node to the sender  $s$  has already been computed.

Centralized PPMA, our centralized solution for the computation and setup of multicast trees, differs from the centralized Bellman–Ford in the crucial choice of the predecessor node. There are a number of ways to choose the father of a node in order to create a multicast tree. In the following, we propose two different criteria applied by centralized PPMA to efficiently address this issue, namely the

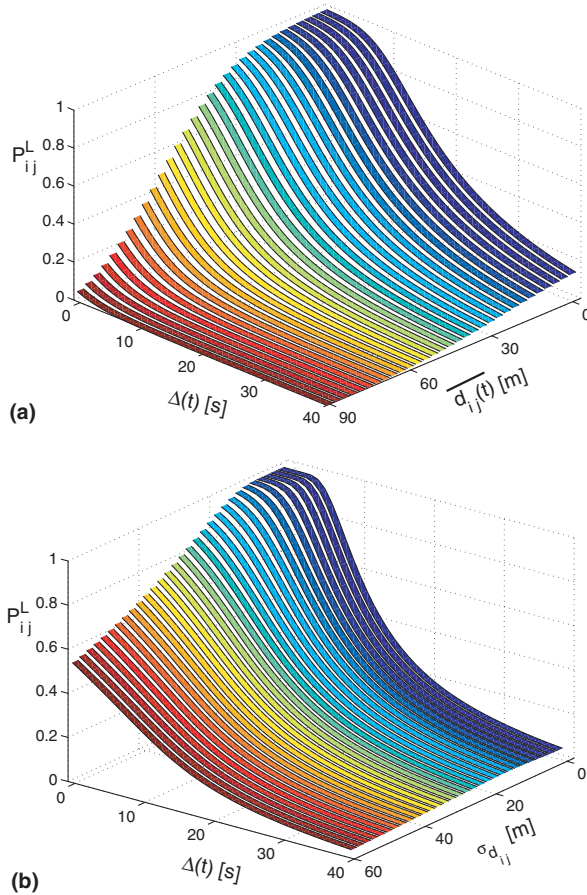


Fig. 5. Lifetime Term for different transmission ranges (a), and for different values of  $\sigma_{d_{ij}}$  (b).

**Algorithm 1.** Centralized PPMA**INIT :**

- 1: node set of the networks  $\rightarrow \mathcal{V}$
- 2: multicast-group set  $\rightarrow \mathcal{M}$
- 3:  $\forall m \in \mathcal{M}$  :
- 4: source  $\rightarrow s$
- 5: current node  $\rightarrow x$
- 6: father of  $x \rightarrow f_x$
- 7: cost of link  $(i, j) \rightarrow C_{ij}$
- 8: path from  $x$  to  $s \rightarrow \mathcal{P}_x$
- 9: cost of path from  $x$  to  $s \rightarrow C_{\mathcal{P}_x}$
- 10:  $\mathcal{H}(x) \equiv \{h \in \mathcal{V} | f_h = x\}$  {child set}
- 11:  $\mathcal{N}(x) \equiv \{n \in \mathcal{V} | C_{nx} < \infty\}$  {neighbor set}
- 12:  $\mathcal{A}(x) \equiv \{a \in \mathcal{V} | d_{sa} < d_{sx}\}$  {positive-advance set}
- 13:  $\mathcal{W} \equiv \{w \in \mathcal{V} | \exists \mathcal{P}_w, C_{\mathcal{P}_w} < \infty\}$  {set of nodes with a feasible path to  $s$ }
- 14:  $\mathcal{F} \equiv \{f \in \mathcal{V} | \mathcal{H}(f) \neq \emptyset\}$  {set of nodes with at least a child}
- 15:  $\mathcal{PP}(x) \equiv \mathcal{N}(x) \cap \mathcal{A}(x) \cap \mathcal{W}$  {potential-predecessor set}
- 16:  $\mathcal{PF}(x) \equiv \mathcal{PP}(x) \cap \mathcal{F}$  {predecessor-father set}

**CENTRALIZED PPMA :**

- 1: **for all**  $m \in \mathcal{M}$  **do**
- 2:     **for all** number of hops **do**
- 3:         **for all**  $x \in \mathcal{V}$  **do**
- 4:             **for all**  $i \in \mathcal{V} - \mathcal{H}(x)$  **do**
- 5:                 FIND\_SET( $\mathcal{PP}(x)$ )
- 6:                 FIND\_SET( $\mathcal{PF}(x)$ )
- 7:                  $\mathcal{PF}^{\text{in}}(x) \leftarrow \mathcal{H}(x) \cap \mathcal{PF}(x)$
- 8:                 **if**  $\mathcal{PF}^{\text{in}}(x) \equiv \emptyset$  **then**
- 9:                      $\mathcal{PF}^{\text{out}}(x) \leftarrow \mathcal{PF}(x) - \mathcal{PF}^{\text{in}}(x)$
- 10:                     **if**  $\mathcal{PF}^{\text{out}}(x) \equiv \emptyset$  **then**
- 11:                          $\mathcal{NF}(x) \leftarrow \mathcal{PP}(x) - \mathcal{PF}(x)$
- 12:                         **if**  $\mathcal{NF}(x) \equiv \emptyset$  **then**
- 13:                              $f_x \leftarrow \text{NULL}$
- 14:                         **else**  $\{\mathcal{NF}(x) \equiv \emptyset\}$
- 15:                              $f_x \leftarrow \text{ARGMIN}_j(C_{\mathcal{P}_j} + C_{jx}), \forall j \in \mathcal{NF}(x)$
- 16:                         **end if**
- 17:                         **else**  $\{\mathcal{PF}^{\text{out}}(x) \neq \emptyset\}$
- 18:                              $f_x \leftarrow \text{FIND\_FATHER\_OUT}(\mathcal{PF}^{\text{out}}(x))$
- 19:                         **end if**
- 20:                         **else**  $\{\mathcal{PF}^{\text{in}}(x) \neq \emptyset\}$
- 21:                              $f_x \leftarrow \text{FIND\_FATHER\_IN}(\mathcal{PF}^{\text{in}}(x))$
- 22:                         **end if**
- 23:             **end for**
- 24:         **end for**
- 25:     **end for**
- 26: **end for**

distance-based criterion and the link-cost-based criterion.

As can be seen in the initialization phase of Algorithm 1, namely ‘INIT’, we define some useful node

sets in order to explain how PPMA works. Let  $\mathcal{V}$  be the set of all the nodes of the network, and  $\mathcal{M}$  the set of all the multicast groups. For a given multicast group  $m \in \mathcal{M}$ , and a given maximum number of

**Algorithm 2.** Choice of the predecessor

**INIT :**

1: current radio-transmitting range of node  $k \rightarrow D_{TX}^k$

**DISTANCE -BASED CRITERION :**

1:  $\mathcal{H}(x) \equiv \{k \in \mathcal{V} | d_{kx} \leq D_{TX}^k\}$

2: FIND\_FATHER\_OUT  $\leftarrow \text{ARGMIN}_z(d_{zx} - D_{TX}^z), \forall z \in \mathcal{P}\mathcal{F}^{\text{out}}(x)$

3: FIND\_FATHER\_IN  $\leftarrow \text{ARGMIN}_w(d_{wx} - D_{TX}^w), \forall w \in \mathcal{P}\mathcal{F}^{\text{in}}(x)$

**LINK -COST -BASED CRITERION :**

1:  $\mathcal{H}(x) \equiv \{k \in \mathcal{V} | C_{kx} \leq C_{kh}^{\max}, h^{\max} = \text{ARGMAX}_h(C_{kh}), h \in \mathcal{H}(k)\}$

2: FIND\_FATHER\_OUT  $\leftarrow \text{ARGMIN}_z(C_{zx} - C_{zh}^{\max}), h^{\max} = \text{ARGMAX}_h(C_{zh}), h \in \mathcal{H}(z), \forall z \in \mathcal{P}\mathcal{F}^{\text{out}}(x)$

3: FIND\_FATHER\_IN  $\leftarrow \text{ARGMIN}_w(C_{wx} - C_{wh}^{\max}), h^{\max} = \text{ARGMAX}_h(C_{wh}), h \in \mathcal{H}(w), \forall w \in \mathcal{P}\mathcal{F}^{\text{in}}(x)$

hops, the current node  $x$  will choose a predecessor in its set of *potential predecessors*  $\mathcal{P}\mathcal{P}(x)$ . We consider a generic node  $v \in \mathcal{V}$  to be a potential predecessor of node  $x$ , if node  $v$  belongs to the intersection of the three sets defined below

$$\begin{aligned} \mathcal{N}(x) &= \{n \in \mathcal{V} | C_{nx} < \infty\} && \text{(neighbor set),} \\ \mathcal{A}(x) &= \{a \in \mathcal{V} | d_{sa} < d_{sx}\} && \text{(positive advance),} \\ \mathcal{W} &= \{w \in \mathcal{V} | \exists \mathcal{P}_w, C_{\mathcal{P}_w} < \infty\} && \text{(nodes with a} \\ &&& \text{path to } s), \end{aligned} \quad (28)$$

where  $C_{ij}$  is the cost of link  $(i, j)$ , according to (1);  $d_{xy}$  is the distance between node  $x$  and node  $y$ ;  $\mathcal{P}_x$  is a feasible path from node  $x$  to the sender  $s$ ;  $C_{\mathcal{P}_x}$  is the cost of the path  $\mathcal{P}_x$ , i.e., the sum of the cost of the links that belong to  $\mathcal{P}_x$ , formally defined as

$$C_{\mathcal{P}_x} = \sum_{(i,j) \in \mathcal{P}_x} C_{ij}. \quad (29)$$

Thus, the set of potential predecessor of node  $x$ ,  $\mathcal{P}\mathcal{P}(x)$ , is

$$\mathcal{P}\mathcal{P}(x) = \mathcal{N}(x) \cap \mathcal{A}(x) \cap \mathcal{W}. \quad (30)$$

- The set  $\mathcal{N}(x)$  contains all the neighbors of node  $x$ , and imposes that the predecessor of  $x$  has to be necessarily a neighbor of  $x$ , where a node  $n$  is defined as a neighbor of  $x$  if the cost of the link  $(n, x)$  is lower than  $\infty$ , i.e.,  $C_{nx} < \infty$ . Note that, according to (1) and (9), the link cost may be  $\infty$  if the distance between the two nodes  $n$  and  $x$  exceeds the maximum radio-transmitting range of node  $n$ .
- The set  $\mathcal{A}(x)$  is the positive-advance set, and contains all the nodes located inside the circle of radius  $d_{sx}$  around  $s$ . The set  $\mathcal{A}(x)$  is needed in

(30) in order to avoid loop formation during the construction of the tree, as proved in [12].

- Finally, the set  $\mathcal{W}$  contains all the nodes that know a feasible path towards  $s$ . A potential predecessor  $w$  of  $x$  must know a feasible path toward the sender  $s$ , otherwise node  $x$  cannot receive any packet from  $s$ .

The set  $\mathcal{P}\mathcal{P}(x)$  can be further partitioned in two subsets:  $\mathcal{P}\mathcal{F}(x)$ , i.e., the set of *predecessor-father* nodes, and  $\mathcal{N}\mathcal{F}(x)$ , i.e., the set of *predecessor-not-father* nodes. It holds that

$$\mathcal{P}\mathcal{P}(x) \equiv \mathcal{P}\mathcal{F}(x) \cup \mathcal{N}\mathcal{F}(x), \mathcal{P}\mathcal{F}(x) \cap \mathcal{N}\mathcal{F}(x) \equiv \emptyset. \quad (31)$$

1. The first subset,  $\mathcal{P}\mathcal{F}(x)$ , is composed of those potential predecessors that currently have other children, so we can call them potential-predecessor fathers, where the *child set* of node  $a$  is defined as the set of nodes whose father is  $a$ . Hence,

$$\mathcal{H}(a) \equiv \{h \in \mathcal{V} | f_h = a\}. \quad (32)$$

We can define a general *father set*  $\mathcal{F}$  as

$$\begin{aligned} \mathcal{F} &\equiv \{f \in \mathcal{V} | \mathcal{H}(f) \neq \emptyset\} \\ &\text{(nodes with at least one child),} \end{aligned} \quad (33)$$

which all the network nodes that have at least one child belong to. Thus, the predecessor-father set can be expressed as

$$\mathcal{P}\mathcal{F}(x) \equiv \mathcal{P}\mathcal{P}(x) \cap \mathcal{F}. \quad (34)$$

2. The second subset,  $\mathcal{N}\mathcal{F}(x)$ , is composed of the remaining potential predecessors and can simply be defined as

$$\mathcal{N}\mathcal{F}(x) \equiv \mathcal{P}\mathcal{P}(x) - \mathcal{P}\mathcal{F}(x). \quad (35)$$

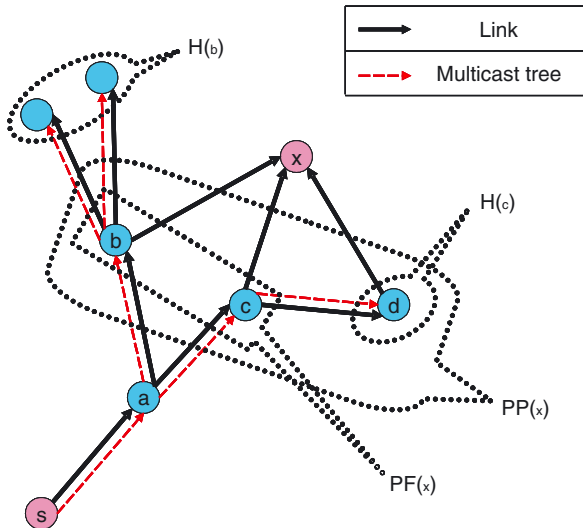


Fig. 6. Centralized PPMA: partition of the potential-predecessor set  $\mathcal{P}\mathcal{P}(x)$ .

In Fig. 6 a simple example of partition of  $\mathcal{P}\mathcal{P}(x)$  is depicted. If a link between two nodes is available, i.e., the link cost is lower than  $\infty$ , it is depicted with a solid arrow, while the multicast tree is depicted with dashed arrows. Node  $x$ , the current node, has to choose its predecessor. The potential-predecessor set of  $x$ ,  $\mathcal{P}\mathcal{P}(x)$ , is composed of nodes  $b$ ,  $c$  and  $d$ . It is easy to verify that all these nodes belong to the sets  $\mathcal{A}(x)$ ,  $\mathcal{N}(x)$  and  $\mathcal{W}$ . In fact, they belong to  $\mathcal{N}(x)$ , since they are neighbors of  $x$ . Moreover, they belong to  $\mathcal{A}(x)$ , since their distance toward  $s$  is less than the distance from  $s$  to  $x$ . Finally, they belong to  $\mathcal{W}$ , since they all have joined the multicast tree, and thus know a feasible path toward the sender  $s$ . However, as can be seen in Fig. 6, only nodes  $b$  and  $c$  have not a void child set. In fact, the child set of node  $b$ ,  $\mathcal{H}(b)$ , is composed of two nodes, nameless in the figure, and the child set of node  $c$ ,  $\mathcal{H}(c)$ , contains node  $d$ . Thus, the potential-predecessor father set of  $x$ ,  $\mathcal{P}\mathcal{F}(x)$ , is composed of nodes  $b$  and  $c$ , while the not-father set,  $\mathcal{N}\mathcal{F}(x)$ , coincides with node  $d$ .

In the choice of the predecessor  $f_x$  of  $x$ , Centralized PPMA gives higher priority to the elements in  $\mathcal{P}\mathcal{F}(x)$  than to the elements in  $\mathcal{N}\mathcal{F}(x)$ . PPMA tries to avoid building up a multicast tree composed of unicast paths, by giving higher priority to those nodes that match a predefined requisite, e.g., being a father. Although such a process may lead to a high tree cost, the total number of links of the spanning tree is kept low. The rationale of this strategy is to

exploit those links that are already used to reach some other receivers. In fact, this is the key feature of multicasting. Hence, PPMA prefers father nodes, and connects node  $x$  to  $s$ , when it is possible, by exploiting already-used paths.

The set  $\mathcal{P}\mathcal{F}(x)$  is divided in two other subsets,  $\mathcal{P}\mathcal{F}^{\text{in}}(x)$  and  $\mathcal{P}\mathcal{F}^{\text{out}}(x)$ , which will be assigned different priorities in the choice of the predecessor of node  $x$ , as will be clearer in the following. So we have

$$\begin{aligned} \mathcal{P}\mathcal{F}(x) &\equiv \mathcal{P}\mathcal{F}^{\text{in}}(x) \cup \mathcal{P}\mathcal{F}^{\text{out}}(x), \\ \mathcal{P}\mathcal{F}^{\text{in}}(x) \cap \mathcal{P}\mathcal{F}^{\text{out}}(x) &\equiv \emptyset. \end{aligned} \quad (36)$$

In particular, PPMA prefers to choose the nodes that belong to  $\mathcal{P}\mathcal{F}^{\text{in}}(x)$  as predecessor of node  $x$  than the nodes that belong to  $\mathcal{P}\mathcal{F}^{\text{out}}(x)$ . We propose two different ‘criteria’ to partition the set  $\mathcal{P}\mathcal{F}(x)$ , the ‘distance-based’ criterion and the ‘link-cost-based’ criterion, which are described in details in the following sections.

### 5.1. Distance-based criterion

The first criterion, namely the ‘distance-based’ criterion, distinguishes the elements of  $\mathcal{P}\mathcal{F}(x)$  that have a current radio-transmitting range  $D_{TX}$  large enough to reach node  $x$ , from those that have not. Thus, according to such criterion, the set  $\mathcal{P}\mathcal{F}^{\text{in}}(x)$  contains the nodes  $f^{\text{in}}$  that are transmitting to some other node  $a$  that is located at a distance  $d_{f^{\text{in}}a}$  from  $f^{\text{in}}$  larger than the distance  $d_{f^{\text{in}}x}$  between  $f^{\text{in}}$  and  $x$ . We can formally define  $\mathcal{P}\mathcal{F}^{\text{in}}(x)$  as

$$\mathcal{P}\mathcal{F}^{\text{in}}(x) \equiv \{f^{\text{in}} \in \mathcal{P}\mathcal{F}(x) | d_{f^{\text{in}}x} \leq D_{TX}^{\text{in}}\}, \quad (37)$$

where  $D_{TX}^{\text{in}}$  is the current radio-transmitting range of node  $f^{\text{in}}$ . Differently, a node that has to increase its current radio-transmitting range to allow node  $x$  to receive the packets it sends, is included in  $\mathcal{P}\mathcal{F}^{\text{out}}(x)$ . So

$$\mathcal{P}\mathcal{F}^{\text{out}}(x) \equiv \{f^{\text{out}} \in \mathcal{P}\mathcal{F}(x) | d_{f^{\text{out}}x} > D_{TX}^{\text{out}}\}. \quad (38)$$

Fig. 7(a) shows an example of network topology that explains this concept. A source  $s$  and three receivers,  $r_1$ ,  $r_2$  and  $r_3$ , must be connected by a multicast tree. The multicast tree is depicted with unidirectional dashed arrows, while unidirectional solid arrows are used to represent existing links (whose costs are shown nearby), and bidirectional dotted arrows indicate distances. Receivers  $r_1$  and  $r_3$  have joined the tree through nodes  $b$  and  $c$ , respectively, but receiver  $r_2$  has not joined the tree yet. Since both

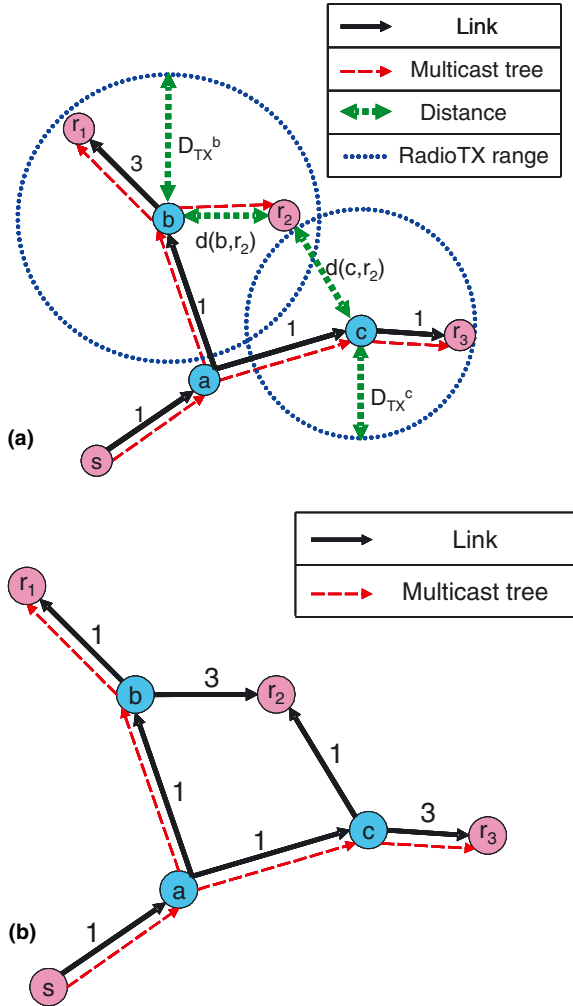


Fig. 7. Centralized PPMA: partition of  $\mathcal{PF}(x)$  in  $\mathcal{PF}^{\text{in}}(x)$  and  $\mathcal{PF}^{\text{out}}(x)$  based on the 'distance' criterion (a), and based on the 'link-cost' criterion (b).

nodes  $b$  and  $c$  have a child, the *potential-predecessor set* and the *predecessor-father set* of receiver  $r_2$  coincide, and are composed of nodes  $b$  and  $c$ , so that  $\mathcal{PP}(r_2) \equiv \mathcal{PF}(r_2) \equiv \{b, c\}$ . However, the current radio-transmitting range of node  $b$ ,  $D_{TX}^b$ , is larger than the distance  $d_{br_2}$  from  $b$  and  $r_2$ , while the radio-transmitting range of node  $c$ ,  $D_{TX}^c$ , is smaller than  $d_{cr_2}$ . Thus,

$$\mathcal{PF}^{\text{in}}(x) \equiv \{b\}, \quad \mathcal{PF}^{\text{out}}(x) \equiv \{c\}. \quad (39)$$

Consequently, receiver  $r_2$  is connected to the sender  $s$  through the predecessor  $b$ , which does not have to increase its current radio-transmitting range to reach  $r_2$ . Note that, according to this partition criterion, the costs of the links  $(c, r_2)$  and  $(b, r_2)$  have not influence in the partition of  $\mathcal{PF}(r_2)$ , i.e., in the

priority given to the nodes in the choice of the predecessor of a node.

## 5.2. Link-cost-based criterion

The second proposed criterion for partitioning the predecessor father set  $\mathcal{PF}(x)$ , namely the '*link-cost-based*' criterion, considers link costs instead of distances. In this case,  $\mathcal{PF}^{\text{in}}(x)$  contains the nodes  $f^{\text{in}} \in \mathcal{PF}(x)$  such that the maximum cost of the links towards their children,  $C_{f^{\text{in}}h^{\text{max}}}$ , is greater than  $C_{f^{\text{in}}x}$ , where  $h^{\text{max}}$  is the child of  $f^{\text{in}}$  that maximizes the cost of the link from  $f^{\text{in}}$ . That is,

$$\begin{aligned} \mathcal{PF}^{\text{in}}(x) &\equiv \{f^{\text{in}} \in \mathcal{PF}(x) | C_{f^{\text{in}}x} \leq C_{f^{\text{in}}h^{\text{max}}}\}, \\ h^{\text{max}} &= \text{ARGMAX}_h(C_{f^{\text{in}}h}), \quad h \in \mathcal{H}(f^{\text{in}}). \end{aligned} \quad (40)$$

Differently, if a node  $f^{\text{out}}$  does not connect a child  $h$ , whose associated link cost  $C_{f^{\text{out}}h}$  is greater than  $C_{f^{\text{out}}x}$ , then  $f^{\text{out}}$  is included in  $\mathcal{PF}^{\text{out}}(x)$ . That is,

$$\begin{aligned} \mathcal{PF}^{\text{out}}(x) &\equiv \{f^{\text{out}} \in \mathcal{PF}(x) | C_{f^{\text{out}}x} > C_{f^{\text{out}}h^{\text{max}}}\}, \\ h^{\text{max}} &= \text{ARGMAX}_h(C_{f^{\text{out}}h}), \quad h \in \mathcal{H}(f^{\text{out}}). \end{aligned} \quad (41)$$

It is worth pointing out that, if we choose the distance  $d_{ij}$  between node  $i$  and node  $j$  as the metric to compute the cost of the link  $(i, j)$ , the '*distance-based*' criterion and the '*link-cost-based*' criterion coincide. Anyway, this second criterion can be viewed in general as an extension of the first one, since the link cost could account for the node speed or the battery charge in addition to the distance.

To illustrate this second criterion, Fig. 7(b) shows the same network topology depicted in Fig. 7(a), but with different link costs. Let us assume that the metric of the link cost takes also into account the *relative speed* of the nodes, i.e., the higher the relative speed between two nodes, the higher the cost of the link connecting them. As in the previous example, in Fig. 7(b) source  $s$  and three receivers,  $r_1$ ,  $r_2$  and  $r_3$ , must be connected by a multicast tree, where  $r_1$  and  $r_3$  have joined the tree through nodes  $b$  and  $c$ , respectively, while  $r_2$  has not joined the tree yet. Since both nodes  $b$  and  $c$  have a child, the potential-predecessor set and the predecessor-father set of receiver  $r_2$  coincide, and are composed by nodes  $b$  and  $c$ ,

$$\mathcal{PP}(r_2) \equiv \mathcal{PF}(r_2) \equiv \{b, c\}. \quad (42)$$

However, the cost of the link  $(b, r_1)$  is smaller than the cost of the link  $(b, r_2)$ , while the cost of the

link  $(c, r_3)$  is larger than the cost of the link  $(c, r_2)$ . Thus,

$$\mathcal{PF}^{\text{in}}(x) \equiv \{c\}, \quad \mathcal{PF}^{\text{out}}(x) \equiv \{b\}. \quad (43)$$

Note that, according to this partition criterion, the distances  $d_{cr_2}$  and  $d_{br_2}$  have influence in the partition of  $\mathcal{PF}(r_2)$ , i.e., in the priority given to the nodes, since the distance between nodes is included in the computation of the link cost. Thus, the receiver  $r_2$  is connected to the sender  $s$  through the predecessor  $c$ , which has already connected a child through a link with a higher cost than  $C_{cr_2}$ .

In general, the partition of the set  $\mathcal{PF}(x)$  can be obtained by intersecting it with another set,  $\mathcal{H}(x)$ , that depends on the used kind of partition.

1. In the case of the ‘distance-based’ partition, the set  $\mathcal{H}(x)$  is defined as

$$\mathcal{H}(x) \equiv \{k \in \mathcal{V} \mid d_{kx} \leq D_{TX}^k\}; \quad (44)$$

2. whereas, in the case of the ‘link-cost-based’ partition, the set  $\mathcal{H}(x)$  is defined as

$$\mathcal{H}(x) \equiv \{k \in \mathcal{V} \mid C_{kx} \leq C_{kh^{\max}}\}, \quad (45)$$

$$h^{\max} = \text{ARGMAX}_h(C_{kh}), \quad h \in \mathcal{H}(k).$$

Once the set  $\mathcal{H}(x)$  is defined, the sets  $\mathcal{PF}^{\text{in}}(x)$  and  $\mathcal{PF}^{\text{out}}(x)$  are easily obtained as

$$\mathcal{PF}^{\text{in}}(x) \equiv \mathcal{PF}(x) \cap \mathcal{H}(x), \quad (46)$$

$$\mathcal{PF}^{\text{out}}(x) \equiv \mathcal{PF}(x) - \mathcal{PF}^{\text{in}}(x). \quad (47)$$

### 5.3. Criteria to choose the predecessor

So far, we defined three sets,  $\mathcal{PF}^{\text{in}}(x)$ ,  $\mathcal{PF}^{\text{out}}(x)$  and  $\mathcal{NF}(x)$ , which contain all the possible predecessors for the current node  $x$ . As explained in Algorithm 1, PPMA looks for the predecessor of  $x$ ,  $f_x$ , primarily in the  $\mathcal{PF}^{\text{in}}(x)$  set. Then, if that set is void, PPMA searches  $f_x$  in the  $\mathcal{PF}^{\text{out}}(x)$  set, and finally, if also  $\mathcal{PF}^{\text{out}}(x)$  is void, in the  $\mathcal{NF}(x)$  set. However, we did not explain how PPMA chooses the predecessor of  $x$ , among the elements of each set. To this end, we propose three criteria for choosing  $f_x$  in a given set  $\mathcal{P}(x)$ .

1. The first criterion chooses the node  $f_x$  such that the path from the receiver  $x$  to the sender  $s$  has the minimum cost, i.e.,

$$f_x \leftarrow \text{ARGMIN}_j(C_{\mathcal{P}_j} + C_{jx}), \quad \forall j \in \mathcal{P}(x). \quad (48)$$

2. The second criterion chooses the node  $f_x$  such that the link from it to the receiver  $x$  has the minimum link cost, i.e.,

$$f_x \leftarrow \text{ARGMIN}_j(C_{jx}), \quad \forall j \in \mathcal{P}(x). \quad (49)$$

3. The third criterion takes into account the differences between the distances or the link costs, whether the partition of the set  $\mathcal{P}(x)$  is ‘distance’-based or ‘link-cost’-based, respectively. In the former case, the difference is between the current radio-transmitting range of the node candidate as predecessor of  $x$  and the distance with  $x$ , i.e.,

$$f_x \leftarrow \text{ARGMIN}_z(d_{zx} - D_{TX}^z), \quad \forall z \in \mathcal{P}(x). \quad (50)$$

In the latter case, the difference is between the cost of the link between the candidate and  $x$ , and the maximum cost of the link between the candidate and its children, i.e.,

$$f_x \leftarrow \text{ARGMIN}_z(C_{zx} - C_{zh^{\max}}),$$

$$h^{\max} = \text{ARGMIN}_h(C_{zh}), \quad h \in \mathcal{H}(z), \quad \forall z \in \mathcal{P}(x). \quad (51)$$

Each of the above criteria can be used by two functions, ‘FIND\_FATHER\_IN’ and ‘FIND\_FATHER\_OUT’, to choose  $f_x$  in the sets  $\mathcal{PF}^{\text{in}}(x)$  and  $\mathcal{PF}^{\text{out}}(x)$ , respectively. The choice among the elements of  $\mathcal{NF}(x)$ , instead, is always based on the first criterion, as in (48).

To summarize, as described in Algorithm 1, if the set  $\mathcal{PF}^{\text{in}}(x)$  is not void, Centralized PPMA chooses the predecessor of  $x$ ,  $f_x$ , in that set, using the function ‘FIND\_FATHER\_IN’. Otherwise, if the set  $\mathcal{PF}^{\text{in}}(x)$  is void, PPMA finds the set  $\mathcal{PF}^{\text{out}}(x)$ , and, if the set is not void, the predecessor of  $x$  is chosen by the function ‘FIND\_FATHER\_OUT’. If also this set is void,  $f_x$  is chosen among the elements in  $\mathcal{NF}(x)$ , according to the criterion in (48). In Algorithm 2 we propose two algorithms, the *distance-based* and the *link-cost-based* algorithm, based on the criteria described in (50) and (51), respectively. Each algorithm defines the set  $\mathcal{H}(x)$  in order to partition  $\mathcal{PF}(x)$ , and also defines the two functions ‘FIND\_FATHER\_IN’ and ‘FIND\_FATHER\_OUT’ to choose  $f_x$  among the elements in  $\mathcal{PF}^{\text{in}}(x)$  and in  $\mathcal{PF}^{\text{out}}(x)$ , respectively.

## 6. Distributed PPMA

Algorithm 3 represents the pseudo-code for Distributed PPMA, the proposed probabilistic

**Algorithm 3.** Distributed PPMA**INIT :**

- 1: node set of the network  $\rightarrow \mathcal{V}$
- 2: multicast-group set  $\rightarrow \mathcal{M}$
- 3: set of receivers  $\rightarrow \mathcal{R}^m, \forall m \in \mathcal{M}$
- 4: multicast tree  $\rightarrow \mathcal{T}^m$
- 5: current node  $\rightarrow x$
- 6: path  $\rightarrow \mathcal{P} \in \mathcal{T}^m$  {path from  $x$  to  $s^m$ }
- 7: private cost of path  $\mathcal{P} \rightarrow C_{\text{priv}}^{\mathcal{P}}$
- 8: public cost of path  $\mathcal{P} \rightarrow C_{\text{pub}}^{\mathcal{P}}$
- 9: father of  $x$  via  $\mathcal{P} \rightarrow f_x^{\mathcal{P}}$
- 10:  $\mathcal{H}(x) \equiv \{h \in \mathcal{V} | \exists \mathcal{P} \in \mathcal{T}, f_h^{\mathcal{P}} = x\}$  {child set}

**DISTRIBUTED PPMA :**

- 1: **for all**  $m \in \mathcal{M}$  **do**
- 2:     **if** ( $x \in \mathcal{R}^m$ ) **then**
- 3:         **if** ( $x \in \mathcal{T}^m$ ) **then**
- 4:              $\mathcal{P}_{\text{new}} \leftarrow \text{ARGMIN}_{\mathcal{P}_{\text{min}}^{\text{priv}}, \mathcal{P}_{\text{current}}} (C_{\text{priv}})$
- 5:             **if**  $\mathcal{P}_{\text{new}} \neq \text{NULL}$  **then**
- 6:                 NEW & CURRENT\_COSTS( $\mathcal{P}_{\text{new}}, \mathcal{P}_{\text{current}}$ )
- 7:                 **if** ( $\text{new\_cost} < \text{current\_cost}$ ) **then**
- 8:                     SWITCH\_FATHER( $f_x^{\mathcal{P}_{\text{current}}}, f_x^{\mathcal{P}_{\text{new}}}$ )
- 9:                     UPDATE\_CURRENT\_COSTS( $\mathcal{P}_{\text{current}}$ )
- 10:                     UPDATE\_NEW\_COSTS( $\mathcal{P}_{\text{new}}$ )
- 11:             **end if**
- 12:         **end if**
- 13:         **else**  $\{x \notin \mathcal{T}^m\}$
- 14:              $\mathcal{P}^* \leftarrow \text{ARGMIN}_{\mathcal{P}} (C_{\text{pub}}^{\mathcal{P}})$
- 15:             JOIN( $f_x^{\mathcal{P}^*}$ )
- 16:             UPDATE\_NEW\_COSTS( $\mathcal{P}^*$ )
- 17:         **end if**
- 18:         **else**  $\{x \notin \mathcal{R}^m\}$
- 19:              $\mathcal{P}_{\text{min}}^{\text{priv}} \leftarrow \text{ARGMIN}_{\mathcal{P}} (C_{\text{priv}}^{\mathcal{P}})$
- 20:              $\mathcal{P}_{\text{min}}^{\text{pub}} \leftarrow \text{ARGMIN}_{\mathcal{P}} (C_{\text{pub}}^{\mathcal{P}})$
- 21:              $\mathcal{P}^+ \leftarrow \text{ARGMIN}_{\mathcal{P}_{\text{min}}^{\text{priv}}, \mathcal{P}_{\text{min}}^{\text{pub}}} (C_{\text{priv}}, C_{\text{pub}})$
- 22:             **if**  $\mathcal{P}^+ \neq \text{NULL}$  **then**
- 23:                 STORE( $\mathcal{P}^+, C_{\text{priv}}^{\mathcal{P}^+}, C_{\text{pub}}^{\mathcal{P}^+}$ )
- 24:             **end if**
- 25:         **end if**
- 26:     **end for** {Daemon running  $\forall x \in \mathcal{V}$ }

**NEW & CURRENT\_COSTS :**

- 1:  $f_x \leftarrow f_x^{\mathcal{P}_{\text{current}}}$
- 2:  $\text{max}_1 \leftarrow \text{MAX}_y (C_{xy}), \forall y \in \mathcal{H}(x) \cap \mathcal{R}^m$
- 3:  $\text{max}_2 \leftarrow \text{MAX}(C_{xf_x})$
- 4:  $\text{max}_3 \leftarrow \text{MAX}_z (C_{f_xz}), \forall z \in \mathcal{H}(f_x) \cap \mathcal{R}^m - \{x\}$
- 5:  $\text{max}_4 \leftarrow \text{MAX}_w (C_{f_xw}), \forall w \in \mathcal{H}(f_x) \cup \mathcal{R}^m$
- 6:  $\text{new\_cost} \leftarrow C_{\text{priv}}^{\mathcal{P}_{\text{new}}} + \text{max}_2 + \text{max}_3$
- 7:  $\text{current\_cost} \leftarrow C_{\text{priv}}^{\mathcal{P}_{\text{current}}} - C_{f_x} + \text{max}_1 + \text{max}_4$

predictive multicast algorithm in its distributed version. In order to build up a multicast tree, Distributed PPMA uses two different types of link costs: a *private link cost*, denoted with  $C_{\text{priv}}$ , and a *public link cost*, denoted with  $C_{\text{pub}}$ . The *private link cost* is used by all nodes, except the receivers not already connected to the tree, to find a minimum-cost path toward the source. The *public link cost* is used by the receivers not already connected to the tree to aggregate the paths, which have been computed using the private costs, to form multicast trees. This way, (i) the number of nodes that belong to the tree is reduced, and (ii) the overhead for control messages is kept low.

For each link we have one *private link cost*, and as many *public link costs* as the total number of multicast groups. The private link costs are computed as in (1), whereas the public link costs either coincide with the private link costs, if the link is not used in a multicast tree, or are set to 0, if the link is part of a multicast tree. The terms ‘*public*’ and ‘*private*’ account for the fact that the two types of costs are or are not ‘*visible*’ to the not already connected receivers, respectively.

Let  $\mathcal{M}$  be the set of the multicast groups.

For a given multicast group  $m \in \mathcal{M}$ , if a node  $x$  is a receiver ( $x \in \mathcal{R}^m$ ), it may or may not have just joined the tree. If the receiver has joined the tree ( $x \in \mathcal{T}^m$ ), it tries to find a new path,  $\mathcal{P}_{\text{new}}$ , which respects the *positive advance* and whose private cost,  $C_{\text{priv}}^{\mathcal{P}_{\text{new}}}$ , is lower than the private cost of the current path,  $C_{\text{priv}}^{\mathcal{P}_{\text{current}}}$ . If a lower cost path is found, the receiver changes path only if  $\text{new\_cost} < \text{current\_cost}$ . This condition ensures that the path change is convenient, i.e., the new path has a cost sufficiently lower than the current one so that the resources spent for changing path will be repaid by the saving induced by the new path. If this condition holds, the SWITCH\_FATHER function switches the old predecessor of node  $x$ ,  $f_x^{\mathcal{P}_{\text{current}}}$ , with the new one,  $f_x^{\mathcal{P}_{\text{new}}}$ . Then, the UPDATE\_CURRENT\_COSTS function changes the public link costs of  $\mathcal{P}_{\text{current}}$  links with their private costs, and the UPDATE\_NEW\_COSTS function sets the public link costs of  $\mathcal{P}_{\text{new}}$  links to 0.

If a receiver has not joined the tree ( $x \notin \mathcal{T}^m$ ), Distributed PPMA finds the path  $\mathcal{P}^*$  with minimum public cost, i.e.,  $\text{ARGMIN}_{\mathcal{P}}(C_{\text{pub}}^{\mathcal{P}})$ . The public cost of a path is computed by summing all the public costs of the links which are part of the path. If such a path is found, the receiver joins the tree and sends

to its new father,  $f_x^{\mathcal{P}^*}$ , a JOIN message to establish the link. Finally, it sets to 0 the public costs of the  $\mathcal{P}_{\text{current}}$  links.

If a node is not a receiver ( $x \notin \mathcal{R}^m$ ), it finds the most convenient path between the available *public paths* and *private paths*, and stores the related father. Whenever another node asks for the cost of the path to the source, the node will give these pieces of information.

Fig. 8 shows how the *new\_cost* and the *current\_cost* are computed by Distributed PPMA. For the sake of clarity, a simple network topology has been chosen. The values of the *private costs* are shown nearby the link arrows. A receiver  $x$  is currently connected to the source via node  $c$ , so that  $\mathcal{P}_{\text{current}} = \overline{xcas}$  is the current path from  $x$  towards the sender  $s$ . Let us assume that  $x$  finds another path towards  $s$ ,  $\mathcal{P}_{\text{new}} = \overline{xbas}$ , which has a lower private cost than  $\mathcal{P}_{\text{current}}$ , i.e.,  $C_{\text{priv}}^{\mathcal{P}_{\text{new}}} < C_{\text{priv}}^{\mathcal{P}_{\text{current}}}$ . In Fig. 8, we can see that this condition holds, since

$$\begin{aligned} C_{\text{priv}}^{\mathcal{P}_{\text{current}}} &= C_{sa} + C_{ac} + C_{cx} = 1 + 4 + 3 = 8 < 3 \\ &= 1 + 1 + 1 = C_{sa} + C_{ab} + C_{bx} = C_{\text{priv}}^{\mathcal{P}_{\text{new}}}. \end{aligned} \quad (52)$$

In such a situation, node  $x$  has to check if  $\text{new\_cost} < \text{current\_cost}$ . As can be seen in the ‘COMPUTE\_COSTS’ subroutine of Algorithm 3, *current\_cost* is composed of four terms,

$$\text{current\_cost} \leftarrow C_{\text{priv}}^{\mathcal{P}_{\text{current}}} - C_{fx} + \text{max}_1 + \text{max}_4. \quad (53)$$

The first term,  $C_{\text{priv}}^{\mathcal{P}_{\text{current}}}$ , is the cost of the current path from  $s$  to  $x$ . The second term,  $C_{fx}$ , is the cost of the

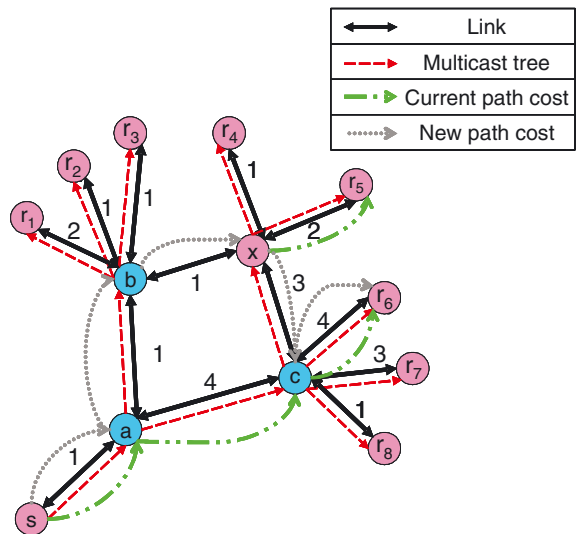


Fig. 8. Distributed PPMA: computation of *current\_cost* and *new\_cost* as in Algorithm 3.

link  $(f_x, x)$ . The third term,  $max_1$ , is the highest private cost of the links between  $x$  and its receiver children, i.e.,

$$\begin{aligned} max_1 &\leftarrow \text{MAX}_y(C_{xy}), \quad \forall y \in \mathcal{H}(x) \cap \mathcal{R}^m, \\ \mathcal{H}(x) &\equiv \{h \in \mathcal{V} \mid f_h = x\}, \quad f_h \text{ predecessor of node } h. \end{aligned} \quad (54)$$

The fourth term,  $max_4$ , is the highest cost of the links between  $f_x$  and its children, for all the children of  $f_x$ , as shown below,

$$max_4 \leftarrow \text{MAX}_w(C_{f_x w}), \quad \forall w \in \mathcal{H}(f_x) \cap \mathcal{R}^m, \quad (55)$$

The sum of these four terms is the *current\_cost* and represents the cost of the sub-tree that connects the source to node  $x$ , and node  $x$  to (i) its receiver child, which is associated to the highest private link cost, and (ii) the receiver child of  $f_x$ , which is associated to the highest private link cost, except  $x$ . Note that Distributed PPMA accounts for the communication broadcast nature in ad hoc networks, since it computes the cost of the sub-tree by summing, for each node belonging to such sub-tree, only the highest private link cost among all the outgoing links.

In Fig. 8, the cost of the link  $(f_x, x)$ , i.e.,  $(c, x)$ , is 3 and only the receivers  $r_4$  and  $r_5$  belong to  $\mathcal{H}(x)$ . Thus,  $max_1$  is  $C_{xr_5}$ , since the cost of the link  $(x, r_4)$  is lower than the cost of the link  $(x, r_5)$  ( $C_{xr_4} = 1 < 2 = C_{xr_5}$ ). Moreover,  $max_4$  is  $C_{cr_6} = 4$ , since receiver  $r_6$  is the child of node  $c$  associated to the highest link cost.

If node  $x$  changes its predecessor, it must be ensured that the new path has not a higher cost than the current one. However, we cannot know this, since we should know which links belong to both the new and the current paths. Thus, we need a *worst-case formula* that guarantees a convenience for the change of the predecessor. To this end, *new\_cost* is defined as

$$new\_cost \leftarrow C_{priv}^{\text{new}} + max_2 + max_3, \quad (56)$$

where the first term,  $C_{priv}^{\text{new}}$ , is the cost of the new path found by node  $x$ . The second term,  $max_2$ , is the maximum between  $max_1$  and the cost of the link connecting  $x$  to its current predecessor  $f_x$ , i.e.,

$$max_2 \leftarrow \text{MAX}(max_1, C_{xf_x}). \quad (57)$$

Finally, the third term,  $max_3$ , is the highest cost of the links between  $f_x$  and its receiver children, except  $x$ , i.e.,

$$max_3 \leftarrow \text{MAX}_z(C_{f_x z}), \quad \forall z \in \mathcal{H}(f_x) \cap \mathcal{R}^m - \{x\}. \quad (58)$$

Hence, *new\_cost* represents the cost of the possible sub-tree connecting the nodes we described in the interpretation of *current\_cost*. In fact, it may be necessary to reroute the children of the current predecessor of  $x$  onto the new path found by  $x$ .

In the case shown in Fig. 8,  $max_2$  is equal to 3 since the cost of the link between  $x$  and its predecessor  $f_x = c$ ,  $C_{xc}$ , is higher than  $max_1$ . In the figure we considered bidirectional links, where the link cost is the same in both directions, for the sake of simplicity. With this assumption,  $max_3$  coincides with  $max_4$ . The links involved in the computation of *new\_cost* and *current\_cost* are shown in figure with a dotted and dash-dotted lines, respectively. Finally, we can compute them as

$$\begin{aligned} current\_cost &= C_{sa} + C_{ac} + C_{cr_6} + C_{xr_5} \\ &= 1 + 4 + 4 + 2 = 11, \\ new\_cost &= C_{sa} + C_{ab} + C_{bx} + C_{xc} + C_{cr_6} \\ &= 1 + 1 + 1 + 3 + 4 = 10. \end{aligned}$$

According to this example, node  $x$  will change its predecessor since *new\_cost* is smaller than *current\_cost*.

## 7. Performance evaluation

This section deals with the performance evaluation of our developed algorithms. In particular, in Section 7.1 we review the most commonly used mobility models for ad hoc networks in the literature, and describe the mobility model that has been developed and used in this paper. Then, in Section 7.2 we describe the simulation scenario, and present the main achieved results.

### 7.1. Network mobility model

The network is represented as  $(\mathcal{V}, \mathcal{D})$ , where  $\mathcal{V} = \{v_1, \dots, v_N\}$  is a finite set of nodes in a finite-dimension terrain, with  $N = |\mathcal{V}|$ , and  $\mathcal{D}$  is a matrix whose element  $(i, j)$  contains the value of the distance between nodes  $v_i$  and  $v_j$ . As far as the mobility models in ad hoc networks are concerned, the most commonly used models will be briefly described hereafter. Then, we propose our mobility model, which is an extension to the Random Walk Model (RWM).

#### 7.1.1. Random waypoint model (RWPM)

In this model, nodes in a large *room* choose a destination, and move there at a random speed

uniformly chosen in  $]0, V_{\max}]$ . Once a node has reached its destination point, it pauses for a time uniformly chosen in  $[0, P_{\max}]$ . If the *terrain border effects* are modeled considering a *wrap around*<sup>1</sup> environment, in the steady state a uniform distribution of nodes in the whole region is generated, whereas if a *bounced back*<sup>2</sup> model is considered, a lower node density in proximity of the borders can be noticed.

### 7.1.2. Random walk model (RWM)

This model has been proposed by Einstein in 1926 to mimic Brownian movements of particles. A node starts moving by picking up a random direction uniformly in  $[0, 2\pi]$ . It continues the movement for a certain time, and then repeats the direction selection again. If during this period it hits the region border, different policies can be implemented to take into account the border effect. The node, in fact, can either be *bounced back*, or continue its movement as if it were in a *wrap around* environment, or be *deleted and replaced*<sup>3</sup> by another node that is placed in a random position inside the region. The first two border-effect models are equivalent in RWM, and give both a uniform distribution of nodes in the steady state, while the latter border-effect model concentrates nodes in the center of the region. The choice of speed is the same as in RWPM. Recent enhancements of this mobility model can be found in [2].

### 7.1.3. Random direction model (RDM)

This model is similar to the *Random Walk* model, differing from that only as far as the node behavior in the border of the region is concerned. According to this model, a node hitting the border will choose its next direction towards the inside half plane. The angle formed by the node direction and the tangent line of the border is chosen uniformly in  $[0, \pi]$ .

<sup>1</sup> In a *wrap around* environment, nodes are assumed to seamlessly move from a side of the terrain to its opposite side, as if the two sides coincided, i.e., all the boundaries of the terrain are assumed to be ‘melted’ together.

<sup>2</sup> In a *bounced back* environment, nodes are assumed to bounce when approaching one side of the terrain, as a ‘ball bounces against a wall’.

<sup>3</sup> In a *deleted and replaced* environment, nodes approaching a side of the terrain are virtually ‘deleted’ and ‘replaced’ by another node, which is placed inside the terrain according to a pre-established rule.

### 7.1.4. Extended-random walk model (E-RWM)

In this paper we extend the *Random Walk* model to include accelerations and decelerations, the main difference being that the speed of a node changes according to an instantaneous acceleration and deceleration, as shown in the following formula, which uses parameters in Table 1,

$$v(t + \Delta t_{\text{mob}}) = \max(V_{\min}, \min(v(t) + a \cdot \Delta t_{\text{mob}}, V_{\max})), \quad (59)$$

$$\begin{cases} a \in [A_{\min}, 0], & p_a > 0.5, \\ a \in [0, A_{\max}], & p_a \leq 0.5, \end{cases} \quad (60)$$

$$\begin{cases} x(t + \Delta t_{\text{mob}}) = x(t) + v(t + \Delta t_{\text{mob}}) \cdot \cos(\phi), \\ y(t + \Delta t_{\text{mob}}) = y(t) + v(t + \Delta t_{\text{mob}}) \cdot \sin(\phi), \end{cases} \quad (61)$$

where  $p_a$  and  $\phi$  are stochastic variables uniformly distributed in  $[0, 1]$  and  $[0, 2\pi]$ , respectively. In (59) the node velocity  $v(t)$  can change every  $\Delta t_{\text{mob}}$  [s], under the constraint that  $V_{\min} \leq v(t) \leq V_{\max}$ , and the additional constraint that its derivative  $a$  [m/s<sup>2</sup>], the node acceleration, is bounded, i.e.,  $A_{\min} \leq a \leq A_{\max}$ .

In the performed simulations, nodes belonging to the same multicast group are grouped into a cluster, which is modeled as a set of nodes deployed in a circular area with a given radius. The source of the relative multicast groups is in the center of this area. Inside a cluster, all nodes move with a low relative speed with respect to the source, i.e., nodes in a cluster are assumed to move ‘coherently’. As a final remark, not all nodes in a cluster are receivers or senders, i.e., some of them may be only relay nodes.

## 7.2. Simulation results

In this work a random ad hoc network has been generated and simulated with an ad hoc simulator. All the parameters used to run simulations are reported in Table 2. The trees built up by Centralized PPMA and Distributed PPMA are compared to the trees built up by the Steiner algorithm [1], which

Table 1  
Network node mobility

Mobility <sup>a</sup>	Velocity and acceleration constraints		
	$V_{\max}$ [m/s]	$A_{\min}$ [m/s <sup>2</sup> ]	$A_{\max}$ [m/s <sup>2</sup> ]
Low	5	−3	2
Medium	10	−5	3
High	15	−7	5

<sup>a</sup> Velocity is uniformly distributed with  $V_{\min} = 0$  for each scenario.

Table 2  
Simulation parameters

Parameter	Value
$E_{elec}$	5 [nJ/bit]
$\beta$	100 [pJ/(bit m <sup>2</sup> )]
$\gamma$	3
$[e^{min}, e^{max}]$	[0.2, 2] [J]
$\mathcal{P}^{max}$	0.55 [W]
Terrain	300 × 300 [m <sup>2</sup> ]
Nodes	65
$\Delta t_{mob}$	0.5 [s]
Packet size	128 [bit]
$r^{req}$	32 [Kbps]

builds multicast trees by minimizing their total cost. Each algorithm builds a tree with two different link-cost functions: the first includes the *Distance Term* only, whereas the second is the *Global Term*, i.e., it includes all the terms (the *Energy*, the *Distance* and the *Lifetime Term*, described in Sections 4.1–4.3, respectively). For each simulation several experiments have been run to ensure 95% relative confidence intervals smaller than 5%.

Starting from a completely unloaded randomly generated ad hoc network, two source-rooted multicast trees are built. Multicast groups are sequentially randomly generated. Multicast-group members (source and receivers) are randomly chosen among the ad hoc network nodes. In particular, as summarized in Table 3, we consider multicast requests from *Small Groups* (5 receivers as mean) and *Large Groups* (11 receivers as mean), in order to test the network under different load conditions.

In Fig. 9 the average *tree lifetime* for PPMA and the Steiner algorithms in a *medium* and *high* mobility environment for *small* multicast group size is shown, while in Fig. 10 the same metric for *large* multicast group size is shown. The curves related to the *Distance Term* show performance that gets worse with a larger multicast group size. Instead, curves related to the *Global Term* are less correlated to the multicast group size. Moreover, the increase of mobility affects the *Distance-Term*-based trees,

Table 3  
Multicast group size

Group size <sup>a</sup>	Mean and standard deviation	
	$N_{mean}$	$\sigma$ (standard deviation)
Small	5	2
Large	11	3

<sup>a</sup> Group size is uniformly distributed with  $N_{mean}$  and  $\sigma$  in the table.

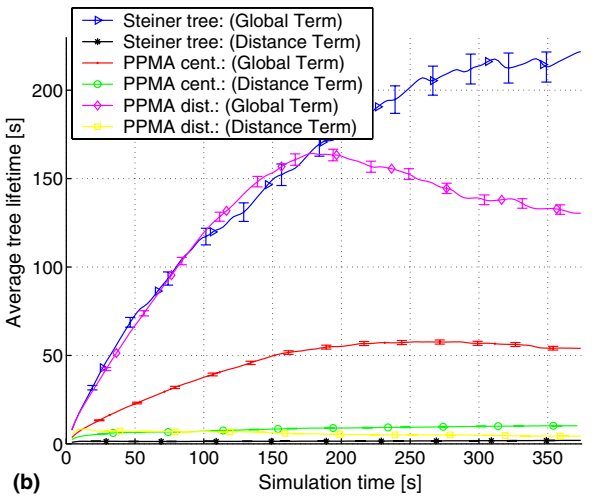
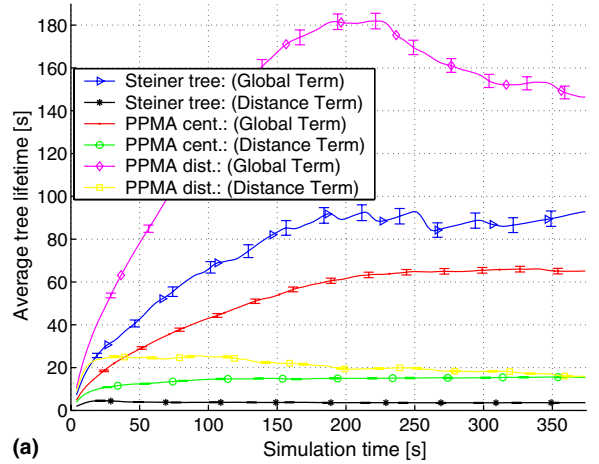
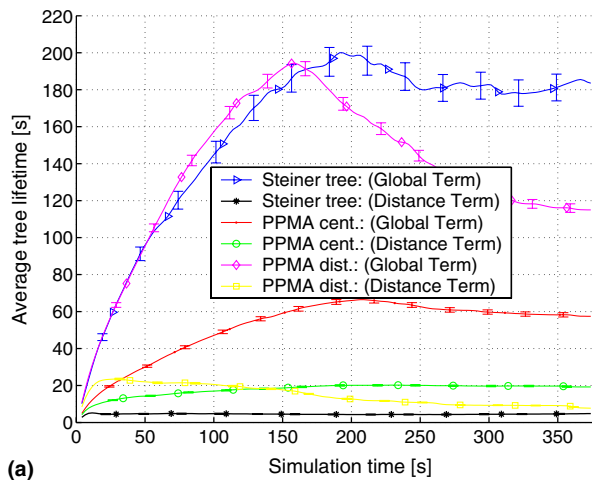
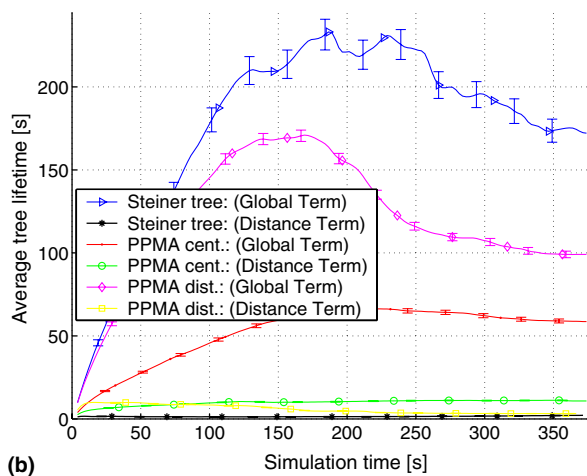


Fig. 9. Tree lifetime for PPMA and Steiner algorithm for small multicast groups in a *medium* mobility (a) and *high* mobility environment (b).

shortening their lifetime. Conversely, trees built by also considering the *Lifetime Term*, as network mobility grows improve their lifetime in the case of the Steiner algorithm or keep their lifetime constant in the case of PPMA. In particular, Distributed PPMA achieves the objective of maximizing the tree lifetime while optimizing the network power consume. The tree lifetime of Distributed PPMA using the *Global Term* metric is much longer than all other curves related to PPMA algorithms, including those obtained with Centralized PPMA. This is due to the lower number of receivers connected by the Distributed algorithm, which is also reflected by a lower number of tree switches needed. Moreover, Centralized PPMA rebuilds all



(a)

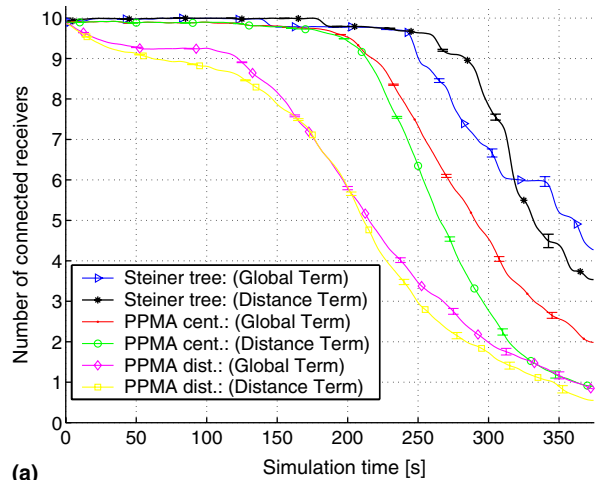


(b)

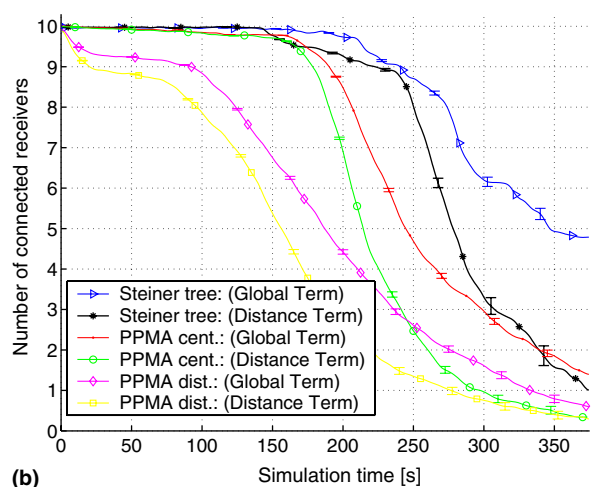
Fig. 10. Tree lifetime for PPMA and Steiner algorithm for *large* multicast groups in *medium* mobility (a) and *high* mobility environment (b).

the multicast trees every update time step, whereas Distributed PPMA tries to keep alive the links already exploited when it is possible. Finally, we can note that Distributed PPMA suffers a little bit more the increase of network mobility and multicast group size than Centralized PPMA. In fact, the latter builds up trees with an almost constant lifetime with respect to the grow of mobility and group size, while the former reduces its lifetime when the mobility or group size increase.

In Fig. 11 the number of *connected receivers* for PPMA and the Steiner algorithm in a *medium* and *high* mobility environment for *small* multicast group size is depicted, while in Fig. 12 the same metric for *large* multicast group size is shown. The number of



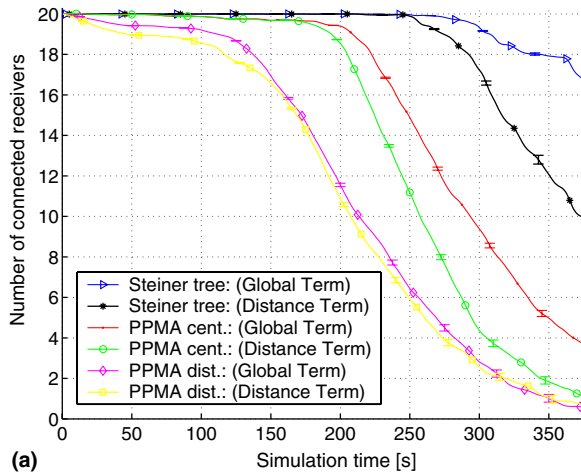
(a)



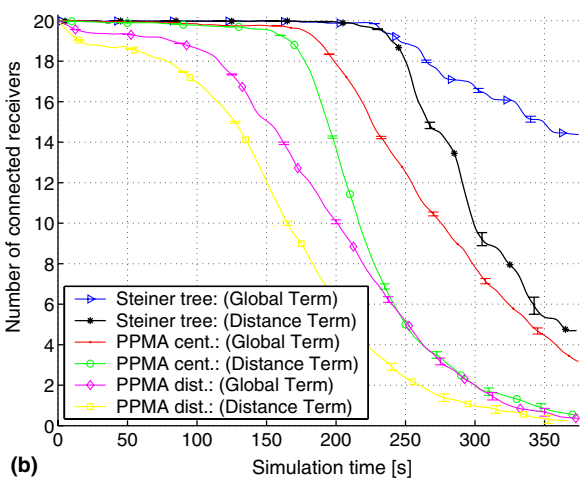
(b)

Fig. 11. Number of connected receivers for PPMA and Steiner algorithm for *small* multicast groups in a *medium* mobility (a) and *high* mobility environment (b).

connected receivers increases when both the group size and mobility increase. Analogously, differences among Distance-Term-based trees and Global-Term-based trees become more evident when both group size and mobility increase. In particular, we have to remark that the Steiner algorithm does not exploit the link cost function as efficiently as PPMA does. In fact, when the multicast group size is small, we can see that the Steiner algorithm performs initially better by using as link cost the Distance Term only, whereas at the end of the simulations it performs much better with the Global Term. Conversely, Centralized and Distributed PPMA have always the curves related to the Global Term higher than those related to the Distance



(a)

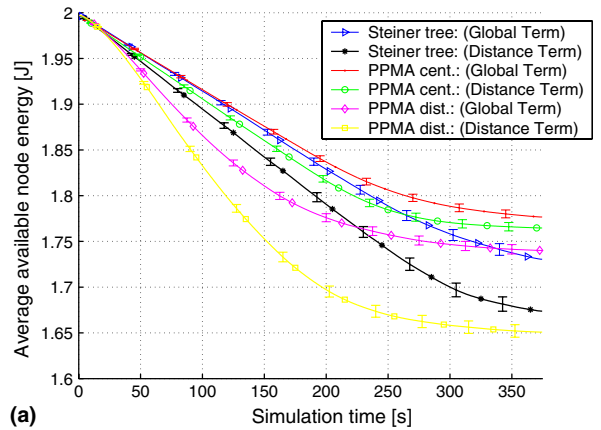


(b)

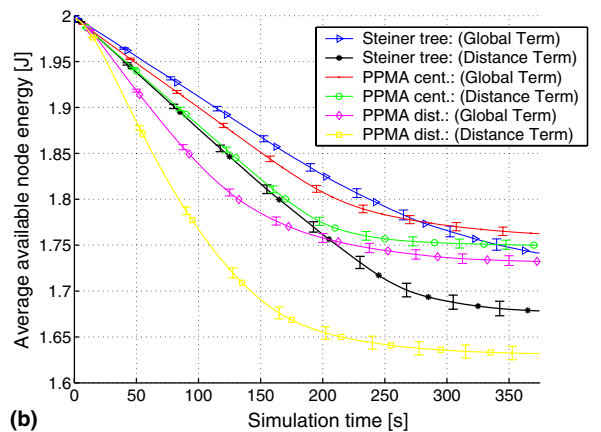
Fig. 12. Number of connected receivers for PPMA and Steiner algorithm for *large* multicast groups in a *medium* mobility (a) and *high* mobility environment (b).

Term. Finally, although Distributed PPMA is more sensible to network mobility than Centralized PPMA, both algorithms perform better in high mobile environment than the Steiner algorithm.

In Fig. 13, the average *battery charge* of the nodes for PPMA and the Steiner algorithm in a *medium* and *high* mobility environment for *small* multicast group size is shown, while in Fig. 14 the same metric for *small* multicast group size is shown. The Energy Term affects the slopes of the curves that include it in the link cost function. The algorithms using the Global Term outperform those using the Distance Term only, and that is particularly evident for the Steiner algorithm. However, Centralized PPMA has a slope comparable to the Steiner algorithm, the latter requiring a computational cost much larger than the former. The curves



(a)



(b)

Fig. 13. Average available node energy for PPMA and Steiner algorithm for *small* multicast groups in a *medium* mobility (a) and *high* mobility environment (b).

related to the PPMA algorithms change slope more rapidly than those related to the Steiner algorithm, since more rapidly the number of connected receivers reach 0. The gain of Distributed PPMA using the Global Term with respect to Distributed PPMA using the Distance Term is remarkable, as depicted in the figure. Moreover, Distributed PPMA using the Global-Term-metric also performs better than Centralized PPMA using the Distance-Term-metric only.

In all these bunches of simulations it is consistently shown that the predictive algorithms outperform the non-predictive algorithms, in both the *Small* and *Large* group simulation scenarios. Moreover, PPMA manages to better leverage the available network resources than the competing algorithm, by exploiting the mobility predictive features it is endowed with. PPMA shows also good robustness properties to mobility, as can be viewed in Figs. 9–12.

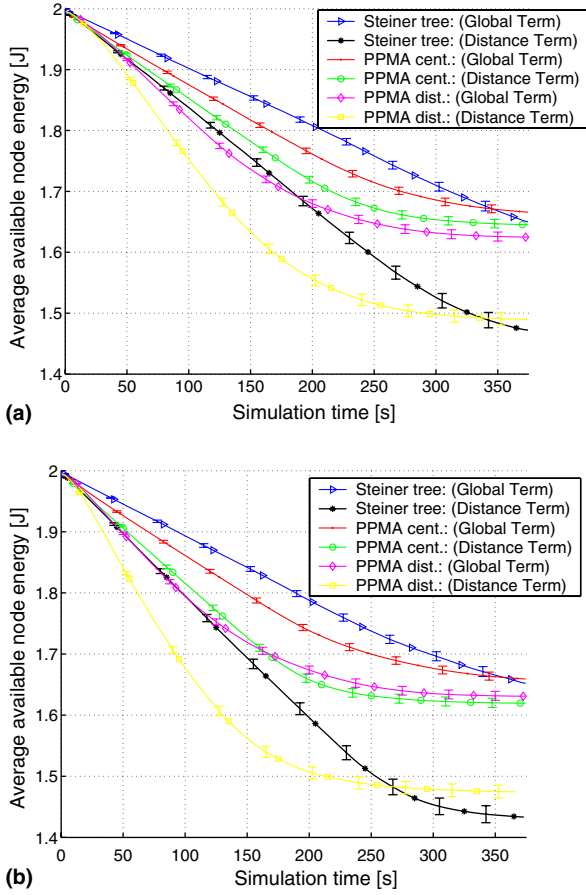


Fig. 14. Average available node energy for PPMA and Steiner algorithm for *large* multicast groups in a *medium* mobility (a) and *high* mobility environment (b).

## 8. Conclusions

This paper proposed PPMA, a new Probabilistic Predictive Multicast Algorithm for ad hoc networks that leverages the tree delivery structure for multicasting, overcoming its limitations in terms of lack of robustness and reliability in highly mobile environments. PPMA exploits the non-deterministic nature of ad hoc networks, by taking into account the estimated network state evolution in terms of node residual energy, link availability and node mobility forecast, to maximize the multicast *tree lifetime*. The algorithm statistically tracks the relative movements among nodes to capture the dynamics in the ad hoc network. This way it estimates the node future relative positions, in order to calculate a *long-lasting* multicast tree. To do so, it exploits the most *stable* links in the network, while minimizing the total network *energy consumption*. We proposed

PPMA in both its *centralized* and *distributed* version, providing performance evaluation through extensive simulation experiments.

## Acknowledgement

The authors are grateful to the many anonymous reviewers that with their unselfish comments greatly improved the content of this paper.

## Appendix A

In this appendix we provide the rigorous derivation of the Lifetime Term expression in (27). To this end, we assume for  $d_{ij}(t)$  and  $|\mathbf{V}_{ij}|$  the following PDFs:

$$P_{d_{ij}}(z) = \frac{1}{\mathcal{A}(\overline{d_{ij}}(t), \sigma_{d_{ij}})} \cdot e^{-\frac{1}{2} \left( \frac{z - \overline{d_{ij}}(t)}{\sigma_{d_{ij}}} \right)^2} \cdot u(z)$$

and

$$P_{|\mathbf{V}_{ij}|}(x) = \frac{x}{\sigma_{|\mathbf{V}_{ij}|}^2} \cdot e^{-\frac{x^2}{2\sigma_{|\mathbf{V}_{ij}|}^2}} \cdot u(x),$$

respectively, where  $\overline{d_{ij}}(t)$  is the measured value and  $\mathcal{A}(\overline{d_{ij}}(t), \sigma_{d_{ij}})$  is defined as

$$\begin{aligned} \mathcal{A}(\overline{d_{ij}}(t), \sigma_{d_{ij}}) &= \int_0^\infty e^{-\frac{1}{2} \left( \frac{w - \overline{d_{ij}}(t)}{\sigma_{d_{ij}}} \right)^2} dw \\ &= \sqrt{\frac{\pi}{2}} \sigma_{d_{ij}} \cdot \left[ 1 + \operatorname{erf} \left( \frac{\overline{d_{ij}}(t)}{\sqrt{2} \cdot \sigma_{d_{ij}}} \right) \right]. \end{aligned}$$

Our objective is computing the following probability:

$$\begin{aligned} &\operatorname{Prob}\{(d_{ij}(t) + |\mathbf{V}_{ij}(t)| \cdot \Delta t) \leq D_i^{\max}\} \\ &= \operatorname{Prob}\{|\mathbf{V}_{ij}(t)| \cdot \Delta t \leq D_i^{\max} - d_{ij}(t)\} \\ &= \int_{-\infty}^\infty P_{d_{ij}(t)}(z) \left( \int_{-\infty}^{D_i^{\max} - z} \frac{1}{\Delta t} \cdot P_{|\mathbf{V}_{ij}|} \left( \frac{x}{\Delta t} \right) dx \right) dz \\ &= \int_{-\infty}^\infty \frac{1}{\mathcal{A}(\overline{d_{ij}}(t), \sigma_{d_{ij}})} \cdot e^{-\frac{1}{2} \left( \frac{z - \overline{d_{ij}}(t)}{\sigma_{d_{ij}}} \right)^2} \cdot u(z) \\ &\quad \cdot \left( \int_{-\infty}^{D_i^{\max} - z} \frac{x}{\Delta t^2 \cdot \sigma_{|\mathbf{V}_{ij}|}^2} \cdot e^{-\frac{x^2}{2\Delta t^2 \cdot \sigma_{|\mathbf{V}_{ij}|}^2}} \cdot u \left( \frac{x}{\Delta t} \right) dx \right) dz \\ &= \frac{1}{\mathcal{A}(\overline{d_{ij}}(t), \sigma_{d_{ij}})} \cdot \frac{1}{\Delta t^2 \cdot \sigma_{|\mathbf{V}_{ij}|}^2} \cdot \int_0^{D_i^{\max}} e^{-\frac{1}{2} \left( \frac{z - \overline{d_{ij}}(t)}{\sigma_{d_{ij}}} \right)^2} \end{aligned}$$

$$\begin{aligned} & \cdot \left( \int_0^{D_i^{\max}-z} x \cdot e^{-\frac{x^2}{2\Delta t^2 \sigma_{|V_{ij}|}}} \cdot dx \right) dz = \frac{1}{\mathcal{A}(\bar{d}_{ij}(t), \sigma_{d_{ij}})} \\ & \cdot \int_0^{D_i^{\max}} e^{-\frac{1}{2} \left( \frac{z-\bar{d}_{ij}(t)}{\sigma_{d_{ij}}} \right)^2} \cdot \left( 1 - e^{-\frac{(D_i^{\max}-z)^2}{2\Delta t^2 \sigma_{|V_{ij}|}}} \right) dz. \quad (62) \end{aligned}$$

Since the variables  $\Delta t$  and  $\sigma_{|V_{ij}|}$  appear only in a product form, in the following we will simply indicate them as a single variable, namely  $\sigma_{\Delta_{ij}} = \Delta t \cdot \sigma_{|V_{ij}|}$ . Thus, (62) simplifies to

$$\begin{aligned} & \frac{1}{\mathcal{A}(\bar{d}_{ij}(t), \sigma_{d_{ij}})} \cdot \int_0^{D_i^{\max}} e^{-\frac{1}{2} \left( \frac{z-\bar{d}_{ij}(t)}{\sigma_{d_{ij}}} \right)^2} \cdot \left( 1 - e^{-\frac{(D_i^{\max}-z)^2}{2\sigma_{\Delta_{ij}}^2}} \right) dz \\ & = \frac{1}{\mathcal{A}(\bar{d}_{ij}(t), \sigma_{d_{ij}})} \cdot \int_0^{D_i^{\max}} e^{-\frac{1}{2} \left( \frac{z-\bar{d}_{ij}(t)}{\sigma_{d_{ij}}} \right)^2} dz \\ & - \frac{1}{\mathcal{A}(\bar{d}_{ij}(t), \sigma_{d_{ij}})} \cdot \int_0^{D_i^{\max}} e^{-\frac{1}{2} \left( \frac{z-\bar{d}_{ij}(t)}{\sigma_{d_{ij}}} \right)^2} \cdot \frac{(D_i^{\max}-z)^2}{2\sigma_{\Delta_{ij}}^2} dz \\ & = \frac{\sqrt{\frac{\pi}{2}} \sigma_{d_{ij}}}{\mathcal{A}(\bar{d}_{ij}(t), \sigma_{d_{ij}})} \\ & \cdot \left[ \operatorname{erf} \left( \frac{D_i^{\max} - \bar{d}_{ij}(t)}{\sqrt{2} \cdot \sigma_{d_{ij}}} \right) + \operatorname{erf} \left( \frac{\bar{d}_{ij}(t)}{\sqrt{2} \cdot \sigma_{d_{ij}}} \right) \right] \\ & - \frac{\sqrt{\frac{\pi}{2}} \sigma_{d_{ij}} \sigma_{\Delta_{ij}} \cdot e^{-\frac{1}{2} \left( \frac{D_i^{\max} - \bar{d}_{ij}(t)}{\sqrt{\sigma_{d_{ij}}^2 + \sigma_{\Delta_{ij}}^2}} \right)^2}}{\sqrt{\sigma_{d_{ij}}^2 + \sigma_{\Delta_{ij}}^2} \cdot \mathcal{A}(\bar{d}_{ij}(t), \sigma_{d_{ij}})} \\ & \cdot \left[ \operatorname{erf} \left( \frac{D_i^{\max} - \bar{d}_{ij}(t)}{\sqrt{2} \sigma_{d_{ij}} \cdot \sqrt{\sigma_{d_{ij}}^2 + \sigma_{\Delta_{ij}}^2}} \right) \right. \\ & \left. + \operatorname{erf} \left( \frac{D_i^{\max} \cdot \sigma_{d_{ij}}^2 - \bar{d}_{ij}(t) \cdot \sigma_{\Delta_{ij}}^2}{\sqrt{2} \sigma_{d_{ij}} \sigma_{\Delta_{ij}} \cdot \sqrt{\sigma_{d_{ij}}^2 + \sigma_{\Delta_{ij}}^2}} \right) \right] \\ & = \frac{\operatorname{erf} \left( \frac{D_i^{\max} - \bar{d}_{ij}(t)}{\sqrt{2} \cdot \sigma_{d_{ij}}} \right) + \operatorname{erf} \left( \frac{\bar{d}_{ij}(t)}{\sqrt{2} \cdot \sigma_{d_{ij}}} \right)}{1 + \operatorname{erf} \left( \frac{\bar{d}_{ij}(t)}{\sqrt{2} \cdot \sigma_{d_{ij}}} \right)} \\ & - \frac{\sigma_{\Delta_{ij}} \cdot e^{-\frac{1}{2} \left( \frac{D_i^{\max} - \bar{d}_{ij}(t)}{\sqrt{\sigma_{d_{ij}}^2 + \sigma_{\Delta_{ij}}^2}} \right)^2}}{\sqrt{\sigma_{d_{ij}}^2 + \sigma_{\Delta_{ij}}^2} \cdot \left[ 1 + \operatorname{erf} \left( \frac{\bar{d}_{ij}(t)}{\sqrt{2} \cdot \sigma_{d_{ij}}} \right) \right]} \end{aligned}$$

$$\begin{aligned} & \cdot \left[ \operatorname{erf} \left( \frac{(D_i^{\max} - \bar{d}_{ij}(t)) \cdot \sigma_{\Delta_{ij}}}{\sqrt{2} \sigma_{d_{ij}} \cdot \sqrt{\sigma_{d_{ij}}^2 + \sigma_{\Delta_{ij}}^2}} \right) \right. \\ & \left. + \operatorname{erf} \left( \frac{D_i^{\max} \cdot \sigma_{d_{ij}}^2 - \bar{d}_{ij}(t) \cdot \sigma_{\Delta_{ij}}^2}{\sqrt{2} \sigma_{d_{ij}} \sigma_{\Delta_{ij}} \cdot \sqrt{\sigma_{d_{ij}}^2 + \sigma_{\Delta_{ij}}^2}} \right) \right]. \quad (63) \end{aligned}$$

Eq. (63) is the final expression of the Lifetime Term, as it appears in (27).

## References

- [1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, 1993.
- [2] I.F. Akyildiz, Y.B. Lin, W.R. Lai, R.J. Chen, A new random walk model for PCS networks, IEEE Journal on Selected Areas in Communications 18 (7) (2000) 1254–1260.
- [3] C. Cheng, R. Riley, S.P.R. Kumar, J.J. Garcia-Luna-Aceves, A loop-free Bellman–Ford routing protocol without bouncing effect, in: Proceedings of ACM SIGMETRICS’89, Oakland, California, USA, September 1989, ACM Press, 1989, pp. 224–237.
- [4] A.J. Goldsmith, Variable-rate variable-power MQAM for fading channels, in: Proceedings of IEEE Vehicular Technology Conference, Atlanta, GA, USA, April 1996, pp. 815–819. URL [citeseer.ist.psu.edu/goldsmith97variable.html](http://citeseer.ist.psu.edu/goldsmith97variable.html).
- [5] A.J. Goldsmith, S.-G. Chua, Variable-rate variable-power MQAM for fading channels, IEEE Transactions on Communications 45 (10) (1997) 1218–1230.
- [6] C. Gui, P. Mohapatra, Efficient overlay multicast for mobile ad hoc networks, in: Proceedings of IEEE WCNC’03, New Orleans, Louisiana, USA, March 2003.
- [7] C. Gui, P. Mohapatra, Scalable multicasting in mobile ad hoc networks, in: Proceedings of IEEE INFOCOM’04, Hong Kong SAR, PRC, March 2004.
- [8] W. Heinzelman, Application-specific protocol architectures for wireless networks. Ph.d. thesis, Massachusetts Institute of Technology, 2000. URL [citeseer.ist.psu.edu/heinzelman00applicationspecific.html](http://citeseer.ist.psu.edu/heinzelman00applicationspecific.html).
- [9] W.B. Heinzelman, A. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless micro-sensor networks, IEEE Transactions on Wireless Communications 1 (4) (2002) 660–670.
- [10] A.B. McDonald, T.F. Znati, A path availability model for wireless ad-hoc networks, in: Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, Louisiana, USA, September 1999.
- [11] A.B. McDonald, T.F. Znati, Predicting node proximity in ad-hoc networks: a least overhead adaptive model for selecting stable routes, in: Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing, IEEE Press, Boston, MA, USA, 2000, pp. 29–33.
- [12] T. Melodia, D. Pompili, I.F. Akyildiz, Optimal local topology knowledge for energy efficient geographical routing in sensor networks, in: Proceedings of IEEE INFOCOM 2004, Hong Kong SAR, PRC, March 2004.

- [13] D. Pompili, L. Lopez, C. Scoglio, DIMRO, A DiffServ-integrated multicast algorithm for Internet resource optimization in source specific multicast applications, in: Proceedings of ICC 2004, Paris, France, June 2004.
- [14] D. Pompili, M. Vittucci, A probabilistic predictive multicast algorithm in ad hoc networks (PPMA), in: Proceedings of IEEE Med-Hoc-Net 2004, Bodrum, Turkey, June 2004.
- [15] E.M. Royer, C.E. Perkins, Multicast Operation of the ad-hoc on-demand distance vector routing protocol, in: Mobile Computing and Networking (MOBICOM), August 1999, pp. 207–218. URL <[citeseer.ist.psu.edu/royer99multicast.html](http://citeseer.ist.psu.edu/royer99multicast.html)>.
- [16] L.H. Sahasrabudde, B. Mukherjee, Multicast routing algorithms and protocols: a tutorial, *IEEE Network* 14 (1) (2000) 90–102.
- [17] S. Sajama, Z.J. Haas, Independent-tree ad hoc multicast routing (ITAMAR), *Mobile Networks & Applications* 8 (5) (2003) 551–566. <<http://dx.doi.org/10.1023/A:1025141912243>>.
- [18] H. Takahashi, A. Matsuyama, An approximate solution for the Steiner problem in graphs, *Mathematica Japonica* 6 (1980) 573–577.
- [19] R.T. Wong, A dual ascent approach for Steiner tree problems on a directed graph, *Mathematical Programming* 28 (1984) 271–287.



**Dario Pompili** graduated in Telecommunications Engineering “summa cum laude” from the University of Rome “La Sapienza” in 2001. In 2004 he earned from the University of Rome “La Sapienza” the Ph.D. degree in System Engineering. From June 2001 he has been working at the same university on the European Union IST Brahms and Satip6 projects. In 2003 he worked on Sensor Networks at the Broadband and Wire-

less Networking Laboratory, Georgia Institute of Technology, Atlanta, as a visiting researcher. Currently he is pursuing the Ph.D. degree in Electrical Engineering at the Georgia Institute of Technology, under the guidance of Dr. Akyildiz. His main research interests are in wireless ad hoc and sensor networks, underwater acoustic sensor networks and satellite networks.



**Marco Vittucci** received the “Laurea” degree in Electronic Engineering in 2004, “summa cum laude”, from the University of Rome “La Sapienza”, with a thesis titled “Project and Design of Centralized and Distributed Algorithms for Multicast Ad Hoc Communications”. Currently, he works as project engineer at the Space Engineering, Italy. His main research interests are in wireless ad hoc and sensor networks, digital

signal processing and multicast communications.