A Probabilistic Predictive Multicast Algorithm in Ad Hoc Networks (PPMA)

Dario Pompili

Broadband and Wireless Networking Laboratory
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332
dario@ece.gatech.edu

Marco Vittucci

Dipartimento di Informatica e Sistemistica
University "La Sapienza" of Rome, Italy, 00184
iccuttivocram57@tiscali.it

***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short in a harsh ad hoc mobile environment because node mobility causes conventional multicast trees to rapidly become outdated. The amount of bandwidth resources required for building up a multicast tree is commonly 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 new probabilistic predictive multicast algorithm in ad hoc networks, which leverages the tree delivery structure for multicasting, solving its drawbacks in terms of lack of robustness and reliability in highly mobile environment. PPMA overcomes the existing tradeoff 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, displacement of set up multicast trees, 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. This way, PPMA estimates the nodes' future relative positions, in order to calculate a *long lasting* multicast tree. To do so, it exploits those more *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 campaign run on an ad hoc C++ based simulator.**

I. Introduction

Ad hoc networks are collections of mobile nodes communicating using wireless media, without any fixed infrastructure. Conventional multicast routings are inadequate in these scenarios, as mobility can cause rapid and frequent changes in the network topology. Existing multicast protocols fall short because node mobility causes conventional multicast trees to rapidly become outdated. 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 for multicast routing protocols to address. In fact, if an algorithm shows robustness to mobility, often incurs in other shortcomings, as protocol overhead or loop formation. 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 degrades with the increase of mobility.

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, based on 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 with 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 [1]. 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 resources required for building up a tree is commonly less than that required for other multicast delivery structures, since a multicast tree avoids unnecessary duplication of data [2]. 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 [3][4][5]. However, a multicast tree is more subject to disruption due to link/node failure and node mobility than more branched structures.

This paper explores these contrasting issues, and proposes PPMA, a new Probabilistic Predictive Multicast Algorithm in ad hoc networks, which leverages the tree delivery structure for multicasting, solving its drawbacks in terms of lack of robustness and reliability in highly mobile environment. There is, in fact, a tradeoff between the bandwidth efficiency 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 tradeoff, by decoupling tree efficiency from mobility robustness. The intuition this paper is based on is that the deterministic nature which characterizes traditional multicast protocols tends to become their limiting factor when aiming at robustness and scalability, particularly 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, displacement of set up multicast trees, 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 nodes' future relative positions, in order to calculate a *long lasting* multicast tree. To do so, it exploits those more *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 campaign run on an ad hoc C++ based simulator.

The remainder of the paper is organized as follows. In Section II we review the main wireless ad hoc network multicast routing protocols which the present work is related to. In Section III we describe the motivations and goals of this paper, introducing our novel probabilistic cost function. In Section IV we explore the terms our cost function is based on, and point out their need to reach the described goals. In Section V we present PPMA, a Probabilistic Predictive Multicast Algorithm in ad hoc networks, in both its centralized and distributed version. In Section VI we show numerical results through extensive simulations run on an ad-hoc C++ based simulator. Finally, in Section VII we conclude the paper.

II. Related Work

There have been several multicast routing protocols proposed for wireless ad hoc networks in literature. In the following we present those which the present work is related to, focusing on some tree based algorithms which 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 II-A, II-B, and II-C we present pros and cons of PAST-DM (Progressively Adapted Sub-Tree in Dynamic Mesh) [6], ITAMAR (Independent-tree ad hoc multicast routing) [7], and AODV (Ad Hoc On-Demand Distance Vector Protocol) [8], respectively. In Section III we describe how our proposed probabilistic multicast algorithm effectively addresses most of their drawbacks.

A. PAST-DM: Progressively Adapted Sub-Tree in Dynamic Mesh

PAST-DM [6] tries to optimize multicast trees in terms of their total link cost and data delivery delay. It utilizes a *virtual mesh topology* that gradually adapts to the changes in underlying network topology in a fully distributed manner, with minimum control cost. The multicast tree for packet delivery is progressively adjusted according to the latest local topology information. A multicast session begins with the construction of a virtual mesh connecting all group members. Each member node starts a *neighbor discovery* process. PAST-DM [6] makes each member node maintain the topology map of the virtual mesh, represented as a link state table. The link state of a node will eventually be carried to the faraway nodes after several exchanges. Through the link state tables, each node has a local view of the whole virtual topology. Thus, each source constructs its own data delivery tree based on its local link state table, with no extra overhead of control message.

This is the key difference between PAST-DM method and other source-based tree protocols. The topology information is more up-to-date and accurate close to the source and progressively less accurate as hop distance increases. Thus, between two virtual links with the same cost during the tree construction, the one that is closer to the source node is favored. To address this property, PAST-DM exploits a Source-Based Steiner tree algorithm [3], that builds up a spanning tree by using an *adapted cost*, i.e. a link cost weighted by the distance to the source. Using adapted cost is a good way to optimize some parameter such as power consume or delivery delay. A link adjacent to the source has distance and hence adapted cost equal to 0. The source makes all its neighbors as its children in the multicast tree and divides the remaining nodes into *subgroups*. Each subgroup forms a subtree rooted at one of the first-level children. The source node does not need to compute the whole multicast tree. It puts each subgroup into a packet header, combines the header with a copy of the data packet, and unicasts the packet to the corresponding child. Each child is then responsible of further delivering the data packet to all nodes in its subgroup and it does so by repeating the Source-Based Steiner tree algorithm [3][6].

Main disadvantages: Virtual mesh as well as any hierarchical structure takes the advantage of scaling very well, since virtual topology can hide the real one regardless of network dimension. However, PAST-DM does not take into account any prediction on nodes mobility in the adapted cost, and thus it could be useless weighting the link cost by a distance that has rapidly changing.

B. ITAMAR: Independent-Tree Ad hoc MulticAst Routing

ITAMAR [7] finds a set of pre-calculated alternate trees to promptly react to link breaks. This way delay could be reduced whenever a viable backup tree is available at the time of failure of the current tree. Backup trees have the advantage of making the tree based scheme more robust to node mobility. Specifically, velocity value, which leads to inconvenient route discovery and maintenance overhead, is much higher, when compared to other tree based schemes. Cost of the multicast tree is optimized along with minimizing the mutual correlation of failure times of each pair of trees under the constraints of partial knowledge of the network. The basic idea is that backup multicast trees with minimal overlap could be used, one after another, to reduce the number of service interruptions. This would also improve the mean time between route discovery cycles for a given interruption rate and hence reduce the control overhead and the rate of data loss. At the same time, ITAMAR aims to keep the cost of transmission low. This method is effective only if the failure times of the trees are independent of one another [7]. Thus, under constraint of nodes moving independently of one another, trees must have no common nodes and hence no common edges. Since totally independent trees could not be found in many cases, ITAMAR concentrates on minimizing the dependence between the failure times, i.e. the correlation of the failure times of the two trees.

Main disadvantages: The main disadvantage if ITAMAR [7] is that it is not convenient in all situations to choose independent trees. In fact, if only some links fail, a tree with a high cost can be chosen. Also, preventing a link-failure by pre-calculating a tree as much independent of the previous tree as possible, could result in a greater control messaging overhead (required for route establishment) than a method capable of repairing only failed links, leaving identical all other tree nodes. Another disadvantage of ITAMAR implementation is that Dijkstra algorithm does not allow to compute the trees in a distributed manner. Consequently, the protocol does not scale well since it requires every nodes to have a global knowledge of the network topology.

C. AODV: Ad hoc On Demand Distance Vector protocol

AODV [8] routing protocol is capable of unicast, broadcast, and multicast communication. One of the main advantage of combining unicast and multicast communication ability in the same protocol 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 as in DSR [9], along with hop-based routing as in DSDV [10], using a *broadcast route discovery* mechanism. In order to reduce communication overheads, updates are propagated only along active routes, i.e. routes that have monitored some track in the recent past. Broadcast data delivery is provided by AODV [8] by using the Source IP Address and Identification fields of the IP header as a unique identifier of the packet. As nodes join the 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. Multicast group members must also agree to be routers in the multicast tree.

Main disadvantages: There are some disadvantages in AODV protocol concerning latency, utilization efficiency, low mobility robustness, and poor scalability property under particular conditions. As far as concerns latency, since in AODV routes are not always the shortest ones, data delivery latency is expected to be worse than 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 which increases the size of the header of data packets. Moreover, since AODV keeps *hard-state* in its routing table, the protocol has to actively track and react to changes in the tree. AODV uses periodic beaconing to keep routing tables updated, thus adding a significant overhead to the protocol. Moreover, since each node maintains a routing table entry for each multicast group for which the node is a member or a router, it suffers high-rate mobility due to transmission of many routing packets. Another disadvantage of AODV is that since nodes make use of their routing caches to reply to route queries, a storm of replies and repetitive updates in hosts' caches may occur, leading to poor scalability performance.

III. Problem Setup

A. Motivations and Goals

The main motivations which convinced us to propose PPMA, a new probabilistic predictive algorithm for multicasting in ad hoc networks, are summarized hereafter.

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

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, displacement of set up multicast trees, 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, PPMA estimates the nodes' future relative positions, and computes a *long lasting* multicast tree. To do so, it exploits those more *stable* links in the network, while minimizing the total network *energy consumption*.

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

1) The more battery charge a node avails, the more its availability to take part in the tree should be;
2) The higher the number of multicast trees a node belongs to, the less the node availability should be;
3) If the available battery charge goes under a predetermined threshold $\mathcal{E}^{min}$, then a node should no more be considered available to take part to multicast communications;
4) The larger is the distance between two nodes, the smaller their availability to establish a communication should be. Obviously, if the distance is larger than a limit range D^{max}, no link between the nodes 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 positions, speeds, and directions of nodes.

B. Transmission Energy Model

An accurate model for node energy consumption per bit at the physical layer is in [11]:

$$E = E_{elec}^{trans} + \beta d^{\gamma} + E_{elec}^{rec} \tag{1}$$

where:

E_{elec}^{trans} is the energy utilized by transmitter electronics (PLLs, VCOs, bias currents, etc) and digital processing. This energy is independent of distance. E_{elec}^{rec} is the energy utilized by receiver electronics, and βd^{γ} accounts for the radiated power necessary to transmit over a distance d between source and destination (β and γ are parameters which depend on the environment, and their values are reported in Table I). As in [12], we assume

$$E_{elec}^{trans} = E_{elec}^{rec} = E_{elec} \tag{2}$$

Thus, the overall expression for E in eq. 1 simplifies to

$$E = 2 \cdot E_{elec} + \beta d^{\gamma} \tag{3}$$

C. Probabilistic Link Cost

In order to synthesize the properties presented in Subsection III-A, we identify a *probabilistic link cost function*, composed of four multiplicative terms, two *Energy Terms*, a *Distance Term*, and a *Lifetime Term*, which will be extensively explained in Subsections IV-A, IV-B, and IV-C, respectively. For two generic nodes i and

j, we consider as their link cost C_{ij} the following:

$$C_{ij} = P_i^{\mathcal{E}} \cdot P_{ij}^{\mathcal{D}} \cdot P_{ij}^{\mathcal{L}} \quad (4)$$

where:

$$P_i^{\mathcal{E}} = P_i^{\mathcal{E}}(\mathcal{E}_i, \mathcal{E}_i^{min}, \mathcal{E}_i^{max}, \mathcal{W}_i^{out}, \mathcal{W}_i^{max}) \quad (5)$$

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

$$P_{ij}^{\mathcal{D}} = P_{ij}^{\mathcal{D}}(d_{ij}, D_i^{max}[r^{req}, \epsilon_b^{mod}, \eta_B^{mod}], \gamma) \quad (6)$$

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

$$P_{ij}^{\mathcal{L}} = P_{ij}^{\mathcal{L}}(d_{ij}, D_i^{max}[r^{req}, \epsilon_b^{mod}, \eta_B^{mod}], \sigma, \Delta t) \quad (7)$$

is the *Lifetime Term* between node i and node j, and it helps statistically evaluate the probability that the distance between these two nodes remains bounded 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 are reported hereafter.

$\mathcal{E}_i$ $[J]$ is the battery state of node i (residual charge); $\mathcal{E}_i^{min}$ $[J]$ is a threshold under which the node is no more available, and $\mathcal{E}_i^{max}$ $[J]$ is the node maximum charge; min and max operators are extended to all nodes in the network;

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

d_{ij} $[m]$ is the distance between the two nodes i and j at time t $[s]$, and D_i^{max} $[m]$ is the maximum radio range node i can reach at its requested bitrate r^{req} $[bit/s]$, given the energy-per-bit ϵ_b^{mod} $[J/bit]$ used in radio transmission, and the spectral efficiency η_B^{mod} $[bit/s/Hz]$ of the adopted modulation system;

β and γ are parameters which depend on the environment;

Δt $[s]$ is the time interval used for mobility prediction, and σ $[m]$ is the standard deviation of the gaussian process used for inter-node distance prediction.

All terms are normalized and range in $[0,1]$, so that they can be viewed as *pseudo-probability* terms. Thus, C_{ij} in eq. 4 can be still viewed as a pseudo-probability. Actually, only the *Lifetime Term* $P_{ij}^{\mathcal{L}}$ is a correctly defined probability, while the other terms are normalized so as to maintain values of C_{ij} in the same variation range. This way, because of the statistical meaning of the link cost C_{ij}, we can associate a probability meaning to the multicast trees computed by the proposed probabilistic predictive algorithm. The *tree probability* will be the product of all the costs of those 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 meaning of 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 Δt, critical parameter in eq. 7. In Section IV we describe in detail all the terms of the link cost function, why each of them is necessary, and their synergic effect to meet the goals described in Subsection III-A.

IV. Probabilistic Link Cost Function Terms

A. Energy Term

The main purpose of the *Energy Term* is to keep all the nodes of the network alive as long as possible, respecting the general rules 1), 2), 3), presented in Section III-A. Also, we want to interpret the energy term associated to 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 (residual charge). Specifically, it assumes greater values for those nodes that have a greater residual charge. Hence, a first attemp for the energy term associated to node i could be:

$$P_i(\mathcal{E}_i) = \left(\frac{\mathcal{E}_i - min_i\{\mathcal{E}_i^{min}\}}{max_i\{\mathcal{E}_i^{max}\} - min_i\{\mathcal{E}_i^{min}\}} \right) \cdot \cdot u_{\text{-}1}(\mathcal{E}_i - min_i\{\mathcal{E}_i^{min}\}) \quad (8)$$

where the unit step $u_{\text{-}1}$ is defined as:

$$u_{\text{-}1}(\mathcal{E}_i - min_i\{\mathcal{E}_i^{min}\}) = \begin{cases} 1 & \mathcal{E}_i > min_i\{\mathcal{E}_i^{min}\} \\ 0 & elsewhere \end{cases} \quad (9)$$

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 ($max_i\{\mathcal{E}_i^{max}\}$). The reason for the presence of min and max operators in eq. 8 is that we want to associate to 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 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 to nodes i and j, respectively, according to the following definition, where no min and max operators are used:

$$P_k(\mathcal{E}_k) = \left(\frac{\mathcal{E}_k - \mathcal{E}_k^{min}}{\mathcal{E}_k^{max} - \mathcal{E}_k^{min}} \right) \cdot u_{\text{-}1}(\mathcal{E}_k - \mathcal{E}_k^{min}) \quad (10)$$

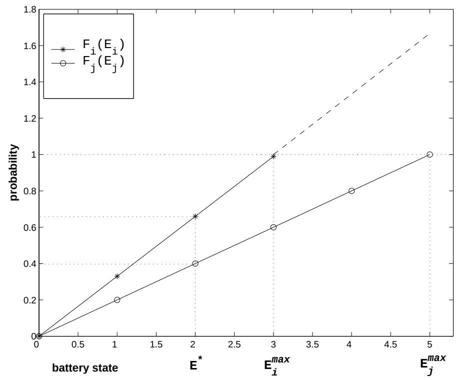


Fig. 1

EXAMPLE OF THE *wrong* ENERGY TERM EXPRESSION IN EQ. 10

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 sake of clearness. We note that eq. 10 coincides with the percentage charge of the total charge of the node k. Also, we note that node j has a larger total charge $\mathcal{E}_j^{max}$ and that, consequentially, $P_j(\mathcal{E}_j)$ has a smaller slope. Now, we see that probabilities of choosing node i and j that have equal charge $\mathcal{E}^*$ are different, so that the charges of the two nodes are not equally weighted. In particular, node i will be more likely to join multicast trees, although equipped with a smaller energy supply. This simple example prove the real need for the min and max operators in eq. 8.

Let us point out a drawback in eq. 8: if a new node h, equipped with an energy supply $\mathcal{E}_h^{max}$ larger than any other node in the network, joined the network, we should redefine eq. 8 by replacing $max_i\{\mathcal{E}_i^{max}\}$ with $\mathcal{E}_h^{max}$. Such an obstacle can be avoided either by knowing a priori $max_i\{\mathcal{E}_i^{max}\}$, or by choosing a $max_i\{\mathcal{E}_i^{max}\}$ large enough to hold for most of the nodes in the network.

If the node battery charge becomes lower than its $\mathcal{E}^{min}$, such a node cannot join anymore multicast trees; consequently, the term must be equal to zero. Otherwise, if a node is not currently involved in any multicast communication, its availability linearly depends on 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 more multicast paths a node shares, the less its availability should be. This behavior can be ensured by an exponent added to eq. 8. Therefor, the final *Energy Term* has the following formula:

$$P_i^{\mathcal{E}} = \left(\frac{\mathcal{E}_i - min_i\{\mathcal{E}_i^{min}\}}{max_i\{\mathcal{E}_i^{max}\} - min_i\{\mathcal{E}_i^{min}\}}\right)^{\left(\frac{\mathcal{W}_i^{max}}{\mathcal{W}_i^{out}+\delta}\right)} \cdot u_{-1}\left(\mathcal{E}_i - min_i\{\mathcal{E}_i^{min}\}\right) \quad (11)$$

where:

δ is a positive constant near to 0 whose objective is to avoid a possible division by zero.

The exponent in eq. 11 is an a-dimensional term, and it varies from 1 to ∞, while $\mathcal{W}_i^{out}$ increases from 0 to $\mathcal{W}_i^{max}$. Thus, as $\mathcal{W}_i^{out}$ grows to $\mathcal{W}_i^{max}$, the energy term should ideally tend to a two slope broken line. Precisely, it should have slope equal to zero in $\mathcal{E}_i \in [\mathcal{E}_i^{min}, \mathcal{E}_i^{max})$, and slope equal to ∞ in $\mathcal{E}_i^{max}$. In this limit case the term is always equal to zero but in $\mathcal{E}_i^{max}$.

B. Distance Term

This term synthesizes the general rule 4) in Section III-A. It takes into account how much power will be spent by nodes i and j to maintain the multicast tree. It can be expressed as follows:

$$P_{ij}^{\mathcal{D}} = \left(\frac{D_i^{max} - d_{ij}}{D_i^{max}}\right)^{\gamma} \cdot u_{-1}(D_i^{max} - d_{ij}) \quad (12)$$

We hide in D_i^{max} the dependence of the cost function on the modulation type:

$$D_i^{max} = D_i^{max}(r^{req}, \epsilon_b^{mod}, \eta_B^{mod}), \forall\, i \in \mathcal{V} \quad (13)$$

Clearly, if $d_{ij} > D^{max}$, the communication link between node i and j does not respect the QoS requirements in terms of minimum available rate r^{req}, and so $P_{ij}^{\mathcal{D}}$ must be 0. $P_{ij}^{\mathcal{D}}$ should decrease from 1 to 0 as d_{ij} increases from 0 to D_i^{max}, at least with a quadratic trend. In fact, received power decreases over distance with an exponent equal to $-\gamma$, according to our propagation model in eq. 3. In free space γ is equal to 2, while in a real indoor or outdoor environment, it can range in the interval [3,4.5]. The unit step function u_{-1} in eq. 12 is needed to ensure $P_{ij}^{\mathcal{D}}$ not to be negative for distance d_{ij} greater than D_i^{max}.

C. Lifetime Term

This term, which synthesizes the general rule 5) in Section III-A, is a predictive one and it is defined as a correct probability function.

$$P_{ij}^{\mathcal{L}} = Prob\{d_{ij}(t+\Delta t) \leq D_i^{max}(r^{req}, \epsilon_b^{mod}, \eta_B^{mod})\} \quad (14)$$

It represents the probability that the distance $d_{ij}(t+\Delta t)$ between nodes i and j, after that Δt seconds elapsed, is less or equal than D_i^{max}. Differently from [13][14], we consider the following probability:

$$Prob\{d_{ij}(t+\Delta t) \leq D_i^{max} \mid d_{ij}(t) \leq D_i^{max}\} \quad (15)$$

where D_i^{max} is the maximum distance a transmitter can reach. It means that we assume that a time t node j is in transmission range of node i ($d_{ij}(t) \leq D_i^{max}$).

The determination of this 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 are exactly known. Furthermore, a closed form expression of $P_{ij}^{\mathcal{L}}$ is hard to find when considering all possible statistical parameters, even under rough approximations of node mobility. Consequently, some assumptions about future movements of nodes are necessary in order to obtain a useful expression of $P_{ij}^{\mathcal{L}}$. An essential information to be known is an esteem of the current distance $d_{ij}(t)$. It is worth pointing out that such a piece of data has to come from some kind of measurements, and so it is prone to errors. In most cases, that error is accurately described by a Gaussian Probability Density Function (PDF) $\mathcal{N}(\overline{d_{ij}(t)}, \sigma_d)$, centered at the measured value $\overline{d_{ij}(t)}$ and with variance σ_d, weighted by a unit step function. So, it can be expressed as follows:

$$P_{d_{ij}(t)}(x) == \frac{1}{\sqrt{2\pi}\cdot\sigma_d}\cdot e^{-\frac{1}{2}\cdot\left(\frac{x-\overline{d_{ij}(t)}}{\sigma_d}\right)^2}\cdot u_{\text{-}1}(x) \quad (16)$$

where the unit step function $u_{\text{-}1}(x)$ is needed because distances are positive defined. Variance σ_d can be set either to a fixed value depending on the measurement system, as in the case of nodes equipped with GPS for location tracking, or to a value proportional to $d_{ij}(t)$, as in the case of nodes capable of measuring relative distances by message exchange.

If the only available piece of data is $\overline{d_{ij}(t)}$, we propose to consider $d_{ij}(t+\Delta t)$ as a stochastic variable, distributed with a gaussian probability density function. We suppose its mean value to be $\overline{d_{ij}(t)}$, i.e. the measured value of $d_{ij}(t)$, and its variance to be the sum of two distinct terms. Before exploring these terms, let us note that such measurement is available to be processed after a time interval Δt_d, that is not necessarily well-known. If we rely on a esteem of Δt_d or we consider it reasonably constant, then we can incorporate this factor into Δt. If not, we have to take a larger σ_d. Thus, actually, we should deal with a gaussian stochastic variable $d_{ij}(t+\Delta t)$, whose mean value is another gaussian stochastic variable. This causes the variance of $d_{ij}(t+\Delta t)$ to be the sum of the variance of $d_{ij}(t)$ (i.e. σ_d^2) and of a term depending on the node relative speeds and the time period Δt (σ_v^2). So, we have:

$$P_{d_{ij}(t+\Delta t)}(x) = \frac{1}{\sqrt{2\pi}\cdot\sigma}\cdot e^{-\frac{1}{2}\cdot\left(\frac{x-\overline{d_{ij}(t)}}{\sigma}\right)^2}\cdot u_{\text{-}1}(x) \quad (17)$$

where $\sigma^2 = \sigma_d^2 + \sigma_v^2$ is derived in appendix. Intuitively we can say that σ_v^2 depends on the relative speed v_{ij} between node i and node j. In fact, the greater v_{ij}, the higher the possible values of $d_{ij}(t+\Delta t)$.

To derive an expression for σ_v^2, we can start from the PDF of the node speed. We assume a zero mean gaussian PDF for each x- and y-component of the velocity vector. This is equivalent to assume a Brownian motion for each node around its current position. So, for the generic node i we have:

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

A profitable way to estimate σ_v^2 is considering the PDF of the amplitude of the velocity vector (which we will refer to as speed), i.e.

$$P_{|V_i|}(x) = \frac{x}{\sigma_i^2}\cdot e^{-\frac{x^2}{2\cdot\sigma_i^2}}\cdot u_{\text{-}1}(x) \quad (19)$$

where $|V_i| = \sqrt{V_{X_i}^2 + V_{Y_i}^2}$. Eq. 19 is a Rayleigh PDF. Under these assumptions, it is possible to tune σ_i^2 in order to properly fit the model. The x- and y-component of the relative speed between two nodes are:

$$\begin{cases} V_{X_{ij}} = \Delta V_X = V_{X_i} - V_{X_j} \\ V_{Y_{ij}} = \Delta V_Y = V_{Y_i} - V_{Y_j} \end{cases} \quad (20)$$

Because ΔV_X and ΔV_Y are difference of gaussian stochastic variables, they still are gaussian stochastic variables. Furthermore, by supposing statistically independent the x- and y-components of the node speed, it results that ΔV_X and ΔV_Y are statistically independent too. ΔV_X 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 ΔV_Y. So,

$$P_{\Delta V_X \Delta V_Y}(x,y) = \frac{1}{2\pi\cdot(\sigma_i^2+\sigma_j^2)}\cdot e^{-\frac{1}{2}\cdot\left(\frac{x^2+y^2}{\sigma_i^2+\sigma_j^2}\right)} \quad (21)$$

Now, we need to relate the V_{ij} amplitude to σ_v^2. A reasonable assumption is that σ_v^2 is proportional to the product of the expected amplitude value V_{ij} and to the time interval Δt. Thus,

$$\sigma_v^2 \propto \Delta t \cdot E\{|V_{ij}|\} = k_v \cdot \Delta t \cdot E\{|V_{ij}|\} \quad (22)$$

where $E\{|V_{ij}|\}$ is the expected value of $|V_{ij}|$, and k_v is a constant. Because ΔV_X and ΔV_Y are zero mean gaussian variables, it results that $|V_{ij}|$ has a Rayleigh PDF too. So, taking $\sigma^2_{V_{ij}} = \sigma^2_i + \sigma^2_j$, it yields:

$$P_{|V_{ij}|}(x) = \frac{x}{\sigma^2_{V_{ij}}} \cdot e^{-\frac{x^2}{2\cdot\sigma^2_{V_{ij}}}} \cdot u_{\text{-}1}(x) \tag{23}$$

$$E\{|V_{ij}|\} = \frac{\sqrt{2\pi}}{2} \cdot \sigma_{V_{ij}} = \sqrt{\frac{\pi}{2} \cdot (\sigma^2_i + \sigma^2_j)} \tag{24}$$

$$\sigma^2_v = k_v \cdot \Delta t \cdot \sqrt{\frac{\pi}{2} \cdot (\sigma^2_i + \sigma^2_j)} = k^I_v \cdot \Delta t \cdot \sqrt{(\sigma^2_i + \sigma^2_j)} \tag{25}$$

On the other hand, if an esteem of $|V_{ij}|$ is available, then $\sigma^2_v \propto \Delta t \cdot |V_{ij}| = k^{II}_v \cdot \Delta t \cdot |V_{ij}|$

V. Centralized and Distributed PPMA

Algorithm 1 represents the pseudo-code for PPMA, the proposed probabilistic predictive multicast algorithm, in its centralized version. It works like the centralized Bellman-Ford algorithm, with the exception of the choice of the father of a node in the computation of the multicast tree. Centralized Bellman-Ford finds the shortest path from node x to the source s, with respect to a certain metric, for every possible number of hops and for every node x. In Algorithm 1 we do not show the *for* loop associated with the number of hops, for sake of simplicity. For a given number of hops, a node x will have a set of potential fathers $\mathcal{F}(x)$. The Centralized Belman-Ford Algorithm chooses the father f_x by minimizing the cost of the associated path. Centralized PPMA, instead, gives higher priority to those potential fathers that have other children, in order to exploit the peculiarity of multicasting. Among these fathers, higher priority is given to those that have a current radio transmitting range which allows node x to receive packets from them. If any so defined fathers exist, node x will choose the closest one. If all potential fathers have to increase their radio range to reach node x, then x will choose that one that has to increase less that range, for power efficient reasons. If all potential fathers have not any children, node x will choose that father f_x that minimizes the LINK_COST(x, f_x).

Algorithm 2 represents the pseudo-code for PPMA, the proposed probabilistic predictive multicast algorithm, in its distributed version. Distributed PPMA uses two different costs: a $private\ cost$ ($\mathcal{C}_{priv}$) and a $public\ cost$ ($\mathcal{C}_{pub}$). The first is needed to find a minimum cost path toward the source, without optimizing the multicast tree. The second is used to enable a node to join an existent multicast tree, trying to reduce the number of nodes that belong to the tree. If a node is a receiver for a given multicast group $m \in \mathcal{M}$, it may have just joined the tree or not. If the receiver has joined the tree, it goes on finding better paths ($\mathcal{P}_{new}$) than the current one ($\mathcal{P}_{current}$). If a smaller cost path is found, the receiver will change path only if $new\_cost < current\_cost$. This condition ensures that path change is convenient, meaning that the new path should have a smaller cost than the current one, so that resources spent for changing path will be repaid by the resource saving induced by the use of the new path. If the receiver has not joined the tree, it finds the best $public\ path$. If it is found, the receiver joins the tree. It must send to its new father a JOIN_ACK for establishing the link, it must publicize to its neighbor the $public\ cost$ of the new path ($\mathcal{C}^{\mathcal{P}_{new}}_{pub}$) and, finally, it must set the $public\ cost$ of the current path to the $private$ one ($\mathcal{C}^{\mathcal{P}_{current}}_{pub} \leftarrow \mathcal{C}^{\mathcal{P}_{current}}_{priv}$). If a node is not a receiver, it finds the most convenient path between $public\ path$ and $private\ path$, and stores the related fathers. Whenever some other node asks for the cost of path to the source, the node will give the lowest-cost path and the related father. Finally, if a node is not a receiver and does not receive any query for links, it must replace the $public\ cost$ with the $private$ one.

VI. Performance Evaluation

A. Network Mobility Model

The network of nodes is represented as $(\mathcal{V}, \mathcal{D})$, where $\mathcal{V} = \{v_1, .., v_N\}$ is a finite set of nodes in a finite-dimension terrain, with $N = |\mathcal{V}|$, and $\mathcal{D}$ is the matrix whose element (i, j) contains the value of the distance between nodes v_i and v_j. As far as concerns the mobility models in ad hoc networks, the most commonly used models will be briefly described hereafter.

1) Random WayPoint Model (RWPM): in this model, nodes in a large *room* choose some 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 P uniformly chosen in $[0, P_{max}]$. If the *border effects* are modeled considering a *wrap around* environment, in the steady state a uniform distribution of nodes in the whole region is generated, whereas if a *bounced back* is considered, a lower node density in proximity of the borders can be noticed.

2) Random Walk Model (RWM): this model has been proposed by Einstein in 1926 to mimic particle's Brownian movement. A node starts moving by picking up a random direction uniformly in $[0, 2\pi]$. It continues

TABLE I
SIMULATION PARAMETERS

Parameter	Value
E_{elec}	$5\ [nJ/bit]$
β	$100\ [pJ/bit \cdot m^{\gamma}]$
γ	3
$[\mathcal{E}^{min}, \mathcal{E}^{max}]$	$[0.2, 2]\ [J]$
$\mathcal{W}^{max}$	$0.55\ [mW]$
$Terrain$	$300 \cdot 300\ [m^2]$
$Nodes$	65
Δt_{mob}	$0.5\ [s]$
$Packet\ size$	$128\ [bit]$
r^{req}	$32\ [Kbps]$

TABLE II
NETWORK NODE MOBILITY

Mobility*	**Velocity and Acceleration constraints**		
	$V_{max}[m/s]$	$A_{min}[m/s^2]$	$A_{max}[m/s^2]$
Low	5	-3	2
Medium	10	-5	3
High	15	-7	5

*Velocity is uniformly distributed with $V_{min} = 0$ for each scenario

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* in a random position inside the region. The first two solutions are in this model equivalent, giving both in the steady state a uniform distribution of nodes, while the latter solution concentrates nodes in the center of the region. The choice of speeds is the same as that in the Random WayPoint Model.

3) Random Direction Model (RDM): this model is similar to the *Random Walk* Model, differing from that only as far as concerns the node behavior in the border of the region. In this model a node hitting the border will choose its next direction towards the inside half plane. The angle formed by node movement direction and the tangent line of the border is chosen uniformly in $[0, \pi]$.

4) Extended-Random Walk Model (E-RWM): in this paper we extended the *Random Walk* Model to include accelerations and decelerations, the main difference being that the speed of a node can change according to an instantaneous acceleration and deceleration, as shown in the following formula which uses parameters in Table II.

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

$$\begin{cases} a \in [A_{min}, 0],\ p_a > 0.5 \\ a \in [0, A_{max}],\ p_a \leq 0.5; \end{cases} \tag{26}$$

$$\begin{cases} x(t + \Delta t_{mob}) = x(t) + v(t + \Delta t_{mob}) \cdot cos(\phi) \\ y(t + \Delta t_{mob}) = y(t) + v(t + \Delta t_{mob}) \cdot sin(\phi) \end{cases} \tag{27}$$

where p_a and ϕ are stochastic variables uniformly distributed in $(0, 1]$ and $[0, 2\pi]$, respectively. Moreover, nodes belonging to the same multicast group are grouped into a cluster. Clusters are modeled as a set of nodes deployed in a circular area with a given radius. The source is in the center of this area. Inside a cluster, all nodes move with a speed coherent to the speed of the source. However, not all nodes in a cluster are receivers or senders.

B. Simulation Results

In this work, a random ad hoc network has been generated, and parameters reported in Table I have been used to run simulations. The trees built up by PPMA are compared to the trees built up by the Steiner algorithm [3], which 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*, whereas the second includes all terms (the *Distance*, the *Energy*, and the *Lifetime*) terms. For each simulation several experiments have been run to ensure 95% relative confidence intervals smaller that 5%. Starting from a completely unloaded randomly generated ad hoc network, 2 source rooted multicast trees are built according to the two competing algorithms. Multicast groups are sequentially randomly generated. Multicast group members (source and receivers) are randomly chosen among ad hoc network nodes. In particular, we considered multicast requests from *Small Groups* (5 receivers) and *Large Groups* (10 receivers), in order to test the network under different load conditions.

In Fig. 2 the average *Tree Lifetime* for PPMA and Steiner algorithm in a *medium* and *high* mobility environment for *small* multicast group size is shown, while in Fig. 3 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 all terms are less correlated to the multicast group size. Moreover, the increase of mobility affects the distance term based trees, shortening their lifetime, while trees built by also considering the lifetime term improve their lifetime as network mobility grows.

In Fig. 4 the number of *Connected Receivers* for PPMA and Steiner algorithm in a *medium* and *high* mobility environment for *small* multicast group size is depicted, while in Fig. 5 the same metric for *large* multicast group size is shown. The number of connected receivers increases at the increase of both group size and mobility. Analogously, differences among distance term based trees and all term based trees get more evident at the growing of both group size and mobility.

In Fig. 6 the average *Battery Charge* of the nodes for PPMA and Steiner algorithm in a *medium* and *high* mobility environment for *small* multicast group size is shown, while in Fig. 7 the same metric for *small* multicast group size is shown. The energy term affects the slopes of the curves starting at about 250 seconds.

In Fig. 8 the *Tree Lifetime* for PPMA and Steiner algorithm for *large* multicast groups in a *high* mobility for three different values of the signaling energy cost for a tree switching is shown. $C_{switch} = 100\%$ corresponds to the energy required to the network to send one 128 bit packet along the multicast tree. Although the tree lifetime decreases at the increase of the considered switching cost, PPMA shows a less sensitive behavior, resulting in a greater robustness to the varying of C_{switch}.

In all these bunches of simulations it is consistently shown that our predictive PPMA algorithm outperforms the Steiner algorithm, both in the *Small* and *Large* Group simulations. Moreover, PPMA manages to leverage better 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 pointed out from Fig. 2-5.

VII. Conclusions

This paper proposed PPMA, a new probabilistic predictive multicast algorithm in ad hoc networks, which leverages the tree delivery structure for multicasting, overcoming its limitations in terms of lack of robustness and reliability in highly mobile environment. 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, displacement of set up multicast trees, 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 nodes' future relative positions, in order to calculate a *long lasting* multicast tree. To do so, it exploits those more *stable*

Algorithm 1
Centralized PPMA

$\boxed{INIT:}$

$nodes\ set\ of\ the\ network \rightarrow \mathcal{V}$
$multicast\ groups\ set \rightarrow \mathcal{M}$
$\forall\, m \in \mathcal{M}:$
$source \rightarrow s;\ current\ node \rightarrow x;\ father\ of\ x \rightarrow f_x$
$\mathcal{F}(x) = \{f \in \mathcal{V} \mid f_x \equiv f\}\{potential\ father\ set\}$
$\mathcal{H}(x) = \{h \in \mathcal{V} \mid f_h \equiv x\}\{children\ set\}$
$\mathcal{N}(x) = \{n \in \mathcal{V} \mid \text{LINK\_COST}(n,x) < \infty\}\{neighbor\ set\}$
$\mathcal{K}^{current}(x) = \mathcal{K}^{curr}(x) = \{k \in \mathcal{V} \mid d(x,k) < D_x^{current}\}$
$\mathcal{K}^{max}(x) = \{k \in \mathcal{V} \mid d(x,k) < D_x^{max}\}$

$\boxed{\text{CENTRALIZED PPMA:}}$

for all $m \in \mathcal{M}$ **do**
  for all $x \in \mathcal{V}$ **do**
    for all $i \in \mathcal{V} - \mathcal{H}(x)$ **do**
      $\mathcal{F}^{D_x^{current}} \leftarrow \text{FIND\_SET}(\mathcal{N}(x) \cap \mathcal{K}^{curr}(i) \cap \mathcal{F}(x))$
      if $\mathcal{F}^{D_x^{current}} \subseteqq \emptyset$ **then**
        $\mathcal{F}^{D_x^{max}} \leftarrow \text{FIND\_SET}(\mathcal{N}(x) \cap \mathcal{K}^{max}(i) \cap \mathcal{F}(x))$
        if $\mathcal{F}^{D_x^{max}} \subseteqq \emptyset$ **then**
          $\mathcal{F}^{new} \leftarrow \text{FIND\_SET}(\mathcal{N}(x) \cap \mathcal{K}^{max}(i))$
          if $\mathcal{F}^{new} \subseteqq \emptyset$ **then**
            $f_x \leftarrow \text{NULL}$
          else $\{\mathcal{F}^{new} \nsubseteqq \emptyset\}$
            $f_x \leftarrow \text{MIN}_j(\text{LINK\_COST}(x,j)), \forall j \in \mathcal{F}^{new}$
          end if
        else $\{\mathcal{F}^{D_x^{max}} \nsubseteqq \emptyset\}$
          $f_x \leftarrow \text{MIN}_z(d(x,z) - D_z^{current}), \forall z \in \mathcal{F}^{D_x^{max}}$
        end if
      else $\{\mathcal{F}^{D_x^{current}} \nsubseteqq \emptyset\}$
        $f_x \leftarrow \text{MIN}_w(d(x,w)), \forall w \in \mathcal{F}^{D_x^{current}}$
      end if
    end for
  end for
end for

Algorithm 2

DISTRIBUTED PPMA

INIT :

$nodes\ set\ of\ the\ network \rightarrow \mathcal{V}$

$multicast\ groups\ set \rightarrow \mathcal{M}$

$set\ of\ receivers \rightarrow \mathcal{R}^m,\ \forall\, m \in \mathcal{M}$

$multicast\ tree \rightarrow \mathcal{T}^m$

$source \rightarrow s^m;\ current\ node \rightarrow x$

$path \rightarrow \mathcal{P} \in \mathcal{T}^m$;{path from x to s^m}

$public\ cost\ of\ path\ \mathcal{P} \rightarrow \mathcal{C}^{\mathcal{P}}_{pub}$;

$private\ cost\ of\ path\ \mathcal{P} \rightarrow \mathcal{C}^{\mathcal{P}}_{priv}$

$father\ of\ x\ via\ \mathcal{P} \rightarrow f^{\mathcal{P}}_x$

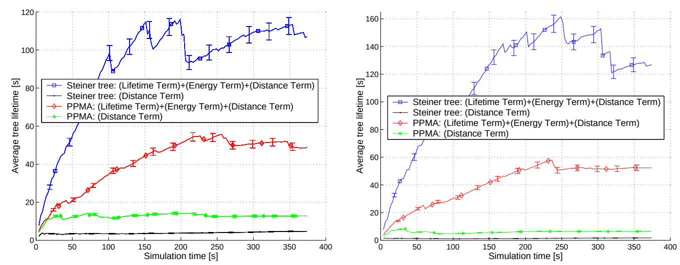
$\mathcal{H}(x) = \{h \in \mathcal{V} \mid \exists\, \mathcal{P} \in \mathcal{T}, f^{\mathcal{P}}_h \equiv x\}\{children\ set\}$

DISTRIBUTED PPMA:

for all $m \in \mathcal{M}$ **do**
  if $(x \neq s^m)$ **then**
    if $(x \in \mathcal{R}^m)$ **then**
      if $(x \in \mathcal{T}^m)$ **then**
        $\mathcal{P}^{priv}_{min} \leftarrow \text{MIN}_{\mathcal{P}}(\mathcal{C}^{\mathcal{P}}_{priv})$
        $\mathcal{P}_{new} \leftarrow \text{FIND\_PATH}(\text{MIN}(\mathcal{P}^{priv}_{min}, \mathcal{C}^{\mathcal{P}_{current}}_{priv}))$
        if $\mathcal{P}_{new} \neq \text{NULL}$ **then**
          $\text{COMPUTE\_COSTS}(\mathcal{P}_{new}, \mathcal{P}_{current})$
          if $(new\_cost < current\_cost)$ **then**
            $\text{SWITCH\_FATHER}(f^{\mathcal{P}_{current}}_x, f^{\mathcal{P}_{new}}_x)$
            $\text{UPDATE\_COST}(\mathcal{C}^{\mathcal{P}_{current}}_{pub}, \mathcal{P}_{current})$
            $\text{UPDATE\_COST}(\mathcal{C}^{\mathcal{P}_{new}}_{pub}, \mathcal{P}_{new})$
          end if
        end if
      else $\{x \notin \mathcal{T}^m\}$
        $\mathcal{P}^* \leftarrow \text{FIND\_PATH}(\text{MIN}_{\mathcal{P}}(\mathcal{C}^{\mathcal{P}}_{pub}))$
        $\text{JOIN}(f^{\mathcal{P}^*}_x)$
        $\text{UPDATE\_COST}(\mathcal{C}^{\mathcal{P}^*}_{pub}, \mathcal{P}^*)$
      end if
    else $\{x \notin \mathcal{R}^m\}$
      $\mathcal{P}^{priv}_{min} \leftarrow \text{MIN}_{\mathcal{P}}(\mathcal{C}^{\mathcal{P}}_{priv})$
      $\mathcal{P}^+ \leftarrow \text{FIND\_PATH}(\text{MIN}(\mathcal{P}^{priv}_{min}, \text{MIN}_{\mathcal{P}}(\mathcal{C}^{\mathcal{P}}_{pub})))$
      if $\mathcal{P}^+ \neq \text{NULL}$ **then**
        $\text{STORE}(\mathcal{P}^+, \mathcal{C}^{\mathcal{P}^+}_{priv}, \mathcal{C}^{\mathcal{P}^+}_{pub})$
      end if
    end if
  end if
end for $\{Daemon\ running\ \forall x \in \mathcal{V}\}$

COMPUTE_COSTS:

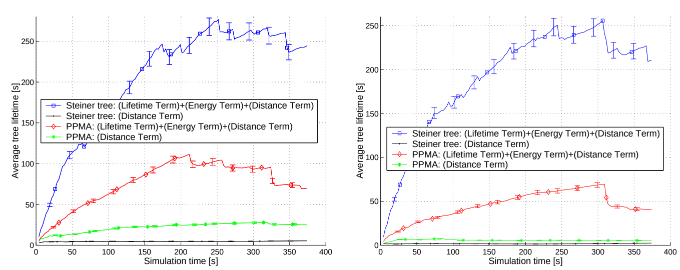
$f \leftarrow f^{\mathcal{P}_{current}}_x$
$max_1 \leftarrow \text{MAX}_y(\text{LINK\_COST}(x, y)), \forall\, y \in \mathcal{H}(x)$
$max_2 \leftarrow \text{MAX}_z(\text{LINK\_COST}(f, z)), \forall\, z \in \mathcal{H}(f) - \{x\}$
$min_1 \leftarrow \text{MIN}(max_1, max_2 + \mathcal{C}^{\mathcal{P}_{current}}_{priv})$
$max_3 \leftarrow \text{MAX}_w(\text{LINK\_COST}(f, w)),\ \forall\, w \in \mathcal{H}(f)$
$new\_cost \leftarrow min_1 + \mathcal{C}^{\mathcal{P}_{new}}_{priv}$
$current\_cost \leftarrow \mathcal{C}^{\mathcal{P}_{current}}_{priv} + max_1 + max_3$



(a) (b)

Fig. 2

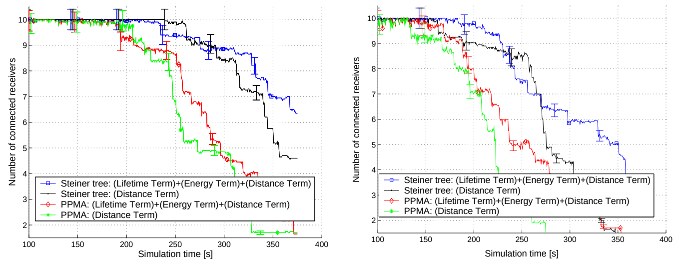
TREE LIFETIME FOR PPMA AND STEINER ALGORITHM FOR SMALL MULTICAST GROUPS IN A *medium* MOBILITY (FIG. 2(A)) AND *high* MOBILITY ENVIRONMENT (FIG. 2(B))



(a) (b)

Fig. 3

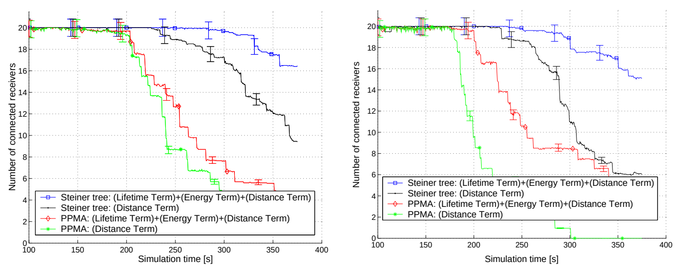
TREE LIFETIME FOR PPMA AND STEINER ALGORITHM FOR *large* MULTICAST GROUPS IN A *medium* MOBILITY (FIG. 3(A)) AND *high* MOBILITY ENVIRONMENT (FIG. 3(B))



(a) (b)

Fig. 4

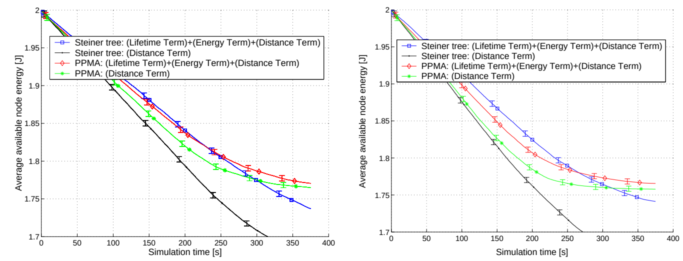
NUMBER OF CONNECTED RECEIVERS FOR PPMA AND STEINER ALGORITHM FOR *small* MULTICAST GROUPS IN A *medium* MOBILITY (FIG. 4(A)) AND *high* MOBILITY ENVIRONMENT (FIG. 4(B))



(a) (b)

Fig. 5

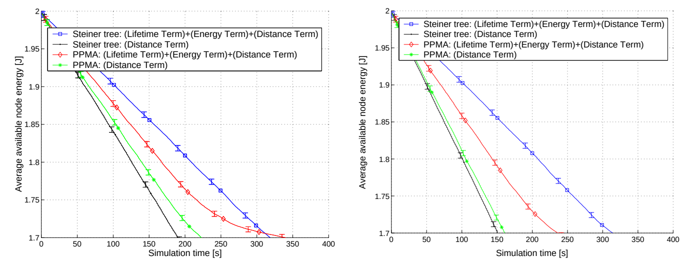
NUMBER OF CONNECTED RECEIVERS FOR PPMA AND STEINER ALGORITHM FOR *large* MULTICAST GROUPS IN A *medium* MOBILITY (FIG. 5(A)) AND *high* MOBILITY ENVIRONMENT (FIG. 5(B))



(a) (b)

Fig. 6

AVERAGE AVAILABLE NODE ENERGY FOR PPMA AND STEINER ALGORITHM FOR *small* MULTICAST GROUPS IN A *medium* MOBILITY (FIG. 6(A)) AND *high* MOBILITY ENVIRONMENT (FIG. 6(B))



(a) (b)

Fig. 7

AVERAGE AVAILABLE NODE ENERGY FOR PPMA AND STEINER ALGORITHM FOR *large* MULTICAST GROUPS IN A *medium* MOBILITY (FIG. 7(A)) AND *high* MOBILITY ENVIRONMENT (FIG. 7(B))

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 campaign run on an ad hoc C++ based simulator.

REFERENCES

[1] L. H. Sahasrabuddhe and B. Mukherjee, "Multicast Routing Algorithms and Protocols: A Tutorial," IEEE Network, Jan/Feb 2000, pp. 90-102.

[2] R. T. Wong, "A dual ascent approach for Steiner tree problems on a directed graph," Mathematical Programming, Vol. 28, pp. 271-287, 1984.

[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, "Network Flows: Theory, Algorithms, and Applications, " Prentice Hall, Feb. 1993.

[4] D. Pompili, L. Lopez, and C. Scoglio, "DIMRO, a DiffServ-Integrated Multicast algorithm for Internet Resource Optimization in source specific multicast applications," IEEE ICC 2004.

[5] H. Takahashi and A. Matsuyama, "An approximate solution for the Steiner problem in graphs," Math. Japonica Vol. 6, pp. 573-577, 1980.

[6] C. Gui and P. Mohapatra, "Efficient Overlay Multicast for Mobile Ad Hoc Networks," Proc. of IEEE WCNC 2003, Mar. 2003.

[7] Sajama and Z. J. Haas, "Independent-tree ad hoc multicast routing (ITAMAR)," Mobile Networks and Applications, Vol. 8, No. 5, Oct. 2003.

[8] E. M. Royer and C. E. Perkins, "Multicast Operation of the

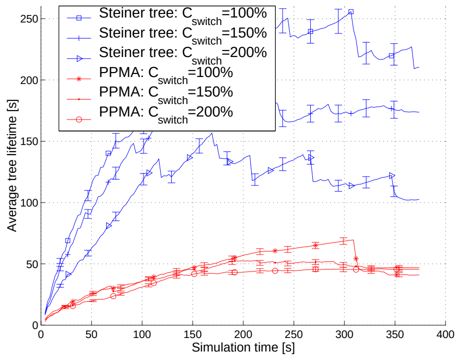


Fig. 8

TREE LIFETIME FOR PPMA AND STEINER ALGORITHM FOR *large* MULTICAST GROUPS IN A *high* MOBILITY, FOR THREE VALUES OF C_{switch}

Ad Hoc On-Demand Distance Vector Routing Protocol," ACM MobiCom, Aug. 1999, pp. 20718.

[9] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in Mobile Computing, Imielinski and Korth, Eds. Kluwer Academic Publishers, 1996, Vol. 353.

[10] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers," in Proceedings of ACM Sigcomm Conference on Communication Architectures, Protocols and Applications, pp. 234-244, Aug. 1994.

[11] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," IEEE Transactions on Wireless Communications, Vol. 1, No. 4, October 2002.

[12] T. Melodia, D. Pompili, and Ian F. Akyildiz, "Optimal Local Topology Knowledge for Energy Efficient Geographical Routing in Sensor Networks," Proc. of IEEE Infocom'04, Hong Kong SAR, PRC

[13] A. B. McDonald and T. F. Znati, "A Path Availability Model for Wireless Ad-Hoc Networks, IEEE Wrireless Communications and Networking Conference (WCNC99), New Orleans, LA, Sep. 21-24, 1999.

[14] A. B. McDonald and T. F. Znati, "Predicting Node Proximity in Ad-Hoc Networks: A Least Overhead Adaptive Model for Selecting Stable Routes," 2000 IEEE

APPENDIX

We assume the following PDF for $d_{ij}(t)$ to be:

$$P_{d_{ij}(t)}(x) = \frac{1}{\sqrt{2\pi}\cdot\sigma_d}\cdot e^{-\frac{1}{2}\cdot\left(\frac{x-\overline{d_{ij}(t)}}{\sigma_d}\right)^2}\cdot u_{-1}(x) \quad (28)$$

where $\overline{d_{ij}(t)}$ is the measured value. Also, we assume the conditional probability of $d_{ij}(t+\Delta t)$ given $d_{ij}(t)$ to be:

$$P_{d_{ij}(t+\Delta t)}(z) = \frac{1}{\sqrt{2\pi}\cdot\sigma}\cdot e^{-\frac{1}{2}\cdot\left(\frac{z-\overline{d_{ij}(t)}}{\sigma}\right)^2}\cdot u_{-1}(Z) \quad (29)$$

Our objective is the following probability:

$$Prob\{d_{ij}(t+\Delta t)\leq D^{max}\} \quad (30)$$

From the *Theorem of Total Probability*, it yields:

$$Prob\{d_{ij}(t+\Delta t)\leq D^{max}\} =$$

$$\sum_{\forall d_{ij}(t)} Prob\{d_{ij}(t+\Delta t)\leq D^{max}|d_{ij}(t)\}\cdot Prob\{d_{ij}(t)\} \quad (31)$$

Thus, in the continuous case:

$$\int_{-\infty}^{\infty}\left(\int_{-\infty}^{D^{max}}\frac{e^{-\frac{1}{2}\cdot\left(\frac{z-x}{\sigma_v}\right)^2}}{\sqrt{2\pi}\cdot\sigma_v}\cdot dz\right)\cdot\frac{e^{-\frac{1}{2}\cdot\left(\frac{x-\overline{d_{ij}(t)}}{\sigma_d}\right)^2}}{\sqrt{2\pi}\cdot\sigma_d}\,dx \quad (32)$$

We suppose σ_d is small with respect to $\overline{d_{ij}(t)}$, so that we can extend to $-\infty$ the limit of the external integral in eq. 33. This is a reasonable assumption if the measurement system has good accuracy, and the measurement is updated frequently enough. By exchanging the order of integration, we have:

$$\frac{1}{2\pi\cdot\sigma_v\cdot\sigma_d}\int_{-\infty}^{D^{max}}\left(\int_{-\infty}^{\infty}e^{-\frac{1}{2}\cdot\left(\frac{z-x}{\sigma_v}\right)^2-\frac{1}{2}\cdot\left(\frac{x-\overline{d_{ij}(t)}}{\sigma_d}\right)^2}dx\right)dz \quad (33)$$

we integrate:

$$\frac{1}{2\pi\cdot\sigma_v\cdot\sigma_d}\int_{-\infty}^{D^{max}}\left(\sqrt{2\pi}\cdot\frac{\sigma_v\cdot\sigma_d}{\sqrt{\sigma_d^2+\sigma_v^2}}\cdot e^{-\frac{1}{2}\cdot\left(\frac{z-\overline{d_{ij}(t)}}{\sqrt{\sigma_d^2+\sigma_v^2}}\right)^2}\right)dz \quad (34)$$

and, finally, we simplify:

$$\frac{1}{\sqrt{2\pi\cdot(\sigma_d^2+\sigma_v^2)}}\int_{-\infty}^{D^{max}}e^{-\frac{1}{2}\cdot\left(\frac{z-\overline{d_{ij}(t)}}{\sqrt{\sigma_d^2+\sigma_v^2}}\right)^2}dz \quad (35)$$

Thus, it has been proved that, under the given assumptions, the variance of $d_{ij}(t+\Delta t)$ is equal to $\sigma_d^2+\sigma_v^2$.