Random walk with jumps in large-scale random geometric graphs ☆

Leonidas Tzevelekas [a,*], Konstantinos Oikonomou [b], Ioannis Stavrakakis [a]

[a] National and Kapodistrian University of Athens, Department of Informatics and Telecommunications, Athens, Greece
[b] Ionian University, Department of Informatics, Corfu, Greece

ARTICLE INFO

ABSTRACT

The *information dissemination* problem in large-scale networking environments like wireless sensor networks and ad hoc networks is studied here considering random geometric graphs and random walk based approaches. A new type of random walk based agent is proposed in this paper and an analytical expression with respect to *coverage* (i.e., the proportion of the network nodes visited by the random walk agent) as a function of the number of the agent movements is derived. It is observed that the cover time of many of already existing random walk based variants is large in random geometric graphs of low degree (as it is commonly the case is wireless environments). As this inefficiency is attributed (as discussed in the paper) to the inability of existing random walk based solutions to move away from already likely covered areas, a mechanism for directional movement (i.e., jumping) of the random walk based agent is proposed and studied, that allows the agent to jump to different network areas, most likely not covered yet. The proposed mechanism (Jumping Random Walk) is studied analytically and via simulations and the parameters (of the network topology and the mechanism) under which the proposed scheme outperforms existing random walk based variations are determined.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

One of the main challenges associated with large-scale, unstructured and dynamic networking environments is that of *efficiently reaching out* to *all* or *a portion* of the network nodes (i.e., *disseminating information*) in order to provide, e.g., software updates or announcements of new services or queries. The high dynamicity and the sheer size of such networking topologies ask for the adoption of decentralized approaches to information dissemination [1–4]. In this paper, the problem of efficiently disseminating information (or queries) across a large-scale, resource-limited, ad hoc-structured wireless network, such as a wireless sensor network, is considered. One of the simplest approaches employed for disseminating information in such environments, is the traditional *flooding* approach. Under flooding [5–8], each time a node receives a message for the first time from some node, it forwards it to all its neighbors except from that node. Despite its simplicity and speed (typically achieving the short-

est cover time, upper bounded by the network diameter), the associated large message overhead is a major drawback.

As flooding is considered not to be an option for large-scale, wireless sensor networks (WSNs) due to strict energy limitations of individual sensor nodes, solutions based on variations of the random walk based information dissemination paradigm are viewed as reasonable choices for searching and/or routing in WSNs [9–11]. Furthermore, there has been a significant body of work in adopting random walks for search or information dissemination in large peer-to-peer (P2P) networks [12–15]. The random walk based information dissemination paradigm possesses several good characteristics such as simplicity, robustness against dynamic failures or changes to the network topology, and lack of need for knowledge of the network physical and topological characteristics. A *Random Walk agent* (RW-agent) doing a simple random walk within a network of wireless sensors moves from neighbor node to neighbor node in a purely random manner, frequently revisiting previously covered nodes in a circular manner. Even when backtracking (returning to the node it just came from) is not allowed, some circular movement in the topology can not be eliminated; these revisits constitute overhead and impact negatively on the cover time [16] of the process. Such a poor behavior of the RW-agent is attributed to the random manner of its movement, combined with some problematic topological characteristics of large-scale wireless ad hoc networks, such as cliques and bottlenecks. To make sure that there is consistency in terminology in this paper, we note here

that the term RW-agent denotes always an agent performing a simple random walk without backtracking in the network, unless otherwise stated.

Large-scale, random geometric graphs (RGGs) have been studied in the past in relationship with percolation theory, statistical physics and hypothesis testing [17]. Recently, the $G(N, r_c)$ RGG has received significant attention due to its applicability in modelling wireless ad hoc and sensor networks, where N is the number of network nodes and r_c is the *connectivity radius* of individual nodes in the field. The network connectivity of the RGG depends on (a) the connectivity radius r_c; and (b) the geometric position of nodes. In particular, any nodes having geometric (euclidean) distance below the connectivity radius r_c are considered to be bi-directionally connected. Naturally, connectivity radius r_c should be large enough, such that the network is connected, i.e., there are no isolated nodes within the network. Such a network connectivity model is substantially different from the well-known scale-free network model arising in many natural and man-made systems, like the Internet, the World Wide Web, citation networks and some social networks. Many such networks fall into the class of scale-free networks, meaning they have power-law (or scale-free) degree distributions. The Barabasi-Albert model is one of several proposed models that generates scale-free networks [18]. The rules of growth and preferential attachment used for creating such networks, allow for any two nodes (no matter how far away from each other) to be connected with non-zero probability. Existence of long-haul links (links connecting nodes residing far apart in physical distance), although perfectly valid in power-law graphs, do not appear in RGGs, due to physical limitations associated with the connectivity radius r_c.

In this paper the Jumping Random Walk (J-RW) mechanism, originally proposed in [19], is shown both analytically and experimentally to be an efficient random walk based information dissemination/ retrieval mechanism for large-scale RGGs. The contributions in this paper, in addition to presenting the J-RW mechanism, include analytical results about the coverage of the RW-agent and J-RW-agent and a new set of simulation experiments to validate the analytical findings on coverage. The proposed J-RW scheme entails the well-known, key benefits of random walk based mechanisms, like simplicity, lack of need for centralized control and robustness to topology changes, while providing a "boost" in performance, i.e. accelerating the coverage process within a RGG (modelling a large-scale WSN). The latter is achieved by introducing a second state of operation of the J-RW-agent, in which the random walk based movement paradigm is replaced by a non-random "directional" movement paradigm. Note that the other state of operation of the J-RW-agent is similar with that of the RW-agent (i.e., it executes a random walk without backtracking). It turns out that this modification in the agent's movement improves significantly the cover time in RGGs by allowing the J-RW-agent to traverse some "virtual" long links in the topology, which are otherwise absent in RGGs. A similar phenomenon has been described in [20] in the context of social networks. Individuals in that pioneering work were assumed to live on the vertices of a grid and thus they know their neighbors for some number of steps in all directions (links to local neighbor nodes are established). They also have a number of acquaintances distributed more broadly across the grid, representing a few, "long range" links of nodes. The author shows that networks constructed under such criterion can accurately model a social network as formed by a group of human beings through their relationships or by computers communicating in the World Wide Web. There can be also a decentralized algorithm (individuals may use only local information) for such network models that is capable of finding short paths between two individuals with high probability.

2. The RW-agent

A credible alternative to flooding for disseminating information in an unstructured, large-scale networking environment, is the random walk based information dissemination paradigm. In random walk based approaches, initiator nodes (representing an end user in P2P networks or a sink in WSNs) employ the random walk agent that will move randomly in the network, one hop/node per time slot, informing (or querying) all the nodes in its path. Random walks in large-scale P2P networks have been shown to possess a number of good properties for searching and/or distributing of information within the network. In particular, authors in [21] have shown that whenever uniform sampling from the set of nodes of a P2P network would have been a good algorithmic approach, the random walk method is an excellent candidate (i) to simulate uniform sampling, moreover (ii) the number of simulation steps required can be as low as the number of samples in independent uniform sampling. Another paper [22] proposes a random walk based search protocol for large-scale P2P Gnutella-like systems, where the biased random walk approach is used for directing queries towards high-capacity nodes (typically best able to answer the queries). Random walks are introduced here mainly for scalability purposes, in contradiction to flooding that can easily overload the network, in case of multiple, simultaneous queries. This work differs in that the scope is only on information propagation/querying in large-scale wireless sensor networks modelled as a random geometric graph.

Random walk based solutions for use in wireless sensor networks have been proposed in [10,11]. In particular, authors of [10] present data gathering in structured wireless sensor networks, where the sensor locations form a regular grid or hex-grid structure. They present analytical findings of the node occupation probability and the first-passage probability of nodes, and through those calculate the latency and energy associated with the system to report from N nodes with N random walks to a set of $\sqrt{N}$ sinks. This work differs from that paper in that the sensor node locations are assumed random, forming a RGG, and also there is a single random walk agent traversing the network for information dissemination/retrieval purposes. Furthermore, a recent theoretical result on speeding up distributed algorithms that use random walks as a subroutine is shown in [11]. The authors propose an algorithm which can compute a random walk sample of length l in fewer than l rounds on an undirected unweighted network. Previous algorithms computing random walk samples of length l do so naively, i.e., in $O(l)$ rounds. This work differs from the paper described previously in that the authors use a multiplicity of short random walks to prefetch information and thus accelerate the sampling process in the network, whereas in this work a different methodology is used without a need to invoke multiple random walk agents in the network, thus possibly causing congestion.

The overhead of random walk based solutions is considered to be much smaller than that of flooding approaches, at the expense of a significant increase in cover time. *Cover time* (*Partial Cover time*) is the expected time taken by a random walk to visit all (a portion of) nodes of a network. The generally relatively large (compared to flooding) cover time achieved under random walk based approaches depends on the network topology. For instance, it is $O(N \ln(N))$ for the fully connected graph (best-case scenario) and $O(N^3)$ for clique topologies (worst case scenario) [16,23]. Random walks on the RGG $G(N, r_c)$ have been shown to have optimal cover time $O(N \ln(N))$ and optimal partial cover time $O(N)$ with high probability given that the connectivity radius of each node r_c fulfills a certain threshold property, i.e. given that $r_c^2 \geqslant \frac{c8 \ln(N)}{N}$, [24]. Generally, it has been shown that C is lower for high connectivity network topologies, such as complete graphs, and it is higher in

network topologies presenting bottlenecks. In the latter case, the number of revisits of already covered nodes (which affect the induced overhead and cover time) becomes particularly high.

The analysis and simulation experiments in this paper utilize a RW-agent variation which is specialized and differs from the simple random walk information dissemination scheme. We employ as our RW-agent the *random walk without backtracking* scheme. In this particular random walk based packet forwarding scheme, the RW-agent is never allowed to backtrack to the immediately previously covered node in the network. This feature of the RW-agent is highly compatible with the proposed changes in random walk based information dissemination as they are explained in the subsequent section. Given a network $G(V, E)$ and a node $v \in V$ with *connectivity degree* $\delta(v)$ the RW-agent employing the random walk without backtracking scheme will choose the next hop node $u \in V$ arbitrarily among the neighbors of node v with probability $p_u = 1/(\delta(v) - 1)$ if $(v, u) \in E$, where u is any neighbor of v except the one from last hop, and $p_u = 0$ for all other $u \in V$. It is evident that the random walk without backtracking scheme is a slight modification (improvement) over the simple random walk based information dissemination paradigm, where the next hop node is chosen uniformly at random among all neighboring nodes of the currently visited node.

The RW-agent moving in accordance with the previously described mechanism, will eventually visit or *cover* all network nodes after some time (cover time). Let $C_r(t)$ be the fraction of network nodes covered (or visited) after t time units or movements of the RW-agent (i.e., the RW-agent start moving at $t = 0$), for a particular realization (sample path) of the walk and for a given initiator node. $C_r(t)$ will be referred to hereafter as the *coverage* at time t. Clearly, $C_r(t)$ depends on the network size, the network topology, the initiator node and other factors. If T_r denotes the cover time, then $C_r(T_r) = 1$; clearly $C_r(0) = 0$. As time increases, the RW-agent is expected either to move to a node that has not been covered previously (thus, $C_r(t)$ increases) or to move to an already covered node (thus, $C_r(t)$ remains the same). Therefore, $C_r(t)$ is a non-decreasing function ($C_r(t_1) \leqslant C_r(t_2)$, for $t_1 < t_2$).

The number of exchanged messages in random walk based solutions is much smaller than that under flooding approaches (each movement of the RW-agent corresponds to one message transmission), at the expense of a significant increase in cover time. For example, in the case of the traditional random walk in a fully connected network (complete graph), the number of messages is $O(N \ln(N))$ [23], while under flooding it approaches $O(N^2)$. On the other hand, cover time for the random walker is $O(N \ln(N))$, while under flooding it is upper bounded by the network diameter. Note here that the number of messages exchanged in the network until the information dissemination process has completed is an important metric of performance, because they represent precious energy resources and/ or delays in the network.

3. The J-RW-agent

3.1. Motivation

Fig. 1a illustrates a random walk based agent movement initiated from the initiator node depicted inside the dotted ellipsis. The random walker spends some time revisiting nodes in the depicted "upper-left" network part, while nodes in other network parts are left unvisited. Suppose now that after a few time units – long enough to "cover" a certain network part – the random walker moves to a "new" (most likely uncovered) network part ("bottom right" network part in Fig. 1b). It is more likely than before to cover nodes that have not been visited previously by the agent, and therefore, accelerate the overall network cover process.

One possible way for the agent to move away from a certain network region would be to carry out a number of consecutive directional movements, implementing a jump. This directional movement mechanism or *jumping*, initially proposed in [19], can be realized by switching occasionally away from the random walk without backtracking operation and engaging an operation implementing a directional movement. That is, such a RW-agent (to be referred to as the Jumping Random Walk agent (J-RW-agent)) operates under two states: *State* 0 under which it implements the typical random walk without backtracking mechanism, and *State* 1 during which the directional move is implemented; the time spent in *State* 1 (freezing state) will be referred to as the freezing (the direction) period.

The J-RW mechanism moves the agent – at the end of the freezing period – to networks that are expected (due to the directional freeze) to be geographically more distant than those reached by the RW-agent after the same number of movements. That is, the introduction of the freezing state implements in essence jumps, defined as the physical distance between the nodes hosting the RW-agent at the beginning and the end of the freezing period.

The improvement in the cover time may be viewed as a consequence of "sampling" the network more uniformly, by moving the "sampling" agent into remote and likely new (not yet sampled) areas, as opposed to keeping the agent wandering around a certain locality according to the RW mechanism and (over)sampling predominately a certain locality. When a network graph has long links (that can take an agent into a remote network region in a topological sense), it has been shown in [21] that a RW-agent produces a uniform sampling of the network nodes.

In wireless environments like sensor networks, the physical and the network topologies are typically correlated: a long path between two nodes in the network topology corresponds to a large physical distance between these nodes in the physical topology. In essence, the proposed J-RW mechanism applied over a network with no long (physical) links (like a WSN) creates virtual long links in this network and results in an environment that is equivalent to that of applying the RW-agent over a network with some long links. Thus, the proposed J-RW mechanism is expected to result in a more uniform sampling of the network nodes, which – as argued earlier – leads to a better cover time.

Besides the improved cover time, the increased uniformity of the (node) sampling under the J-RW mechanism may be on its own another important property of the proposed information dissemination scheme when considered in conjunction with certain specific and fairly common applications such as those related to sensing the environment. In such applications and due to the typically high spatial correlation of nearby nodes, a dissemination of a query on the state of the field may target only a portion of the network nodes to conserve energy, [16]. Since the J-RW mechanism possesses the uniform sampling capability as argued earlier, it is expected that the query dissemination and collected responses would better represent the state of the sensor field and contain less redundant information. For such environments it is reasonable to base the evaluation of information dissemination schemes on the partial cover time as opposed to the (100%) cover time.

3.2. Description

The proposed J-RW mechanism is based on two underlying states. State transitions of the J-RW-agent are assumed to occur at discrete times, prior to each forwarding decision of the J-RW-agent. The simple 2-state Markov chain, as shown in Fig. 2 governs the type of movement the agent executes in the network. When in *State* 0, the J-RW-agent operates as the already described RW-agent. Each time the steering markov chain transitions to *State* 1, the geographic direction of the imminent jump in the network is
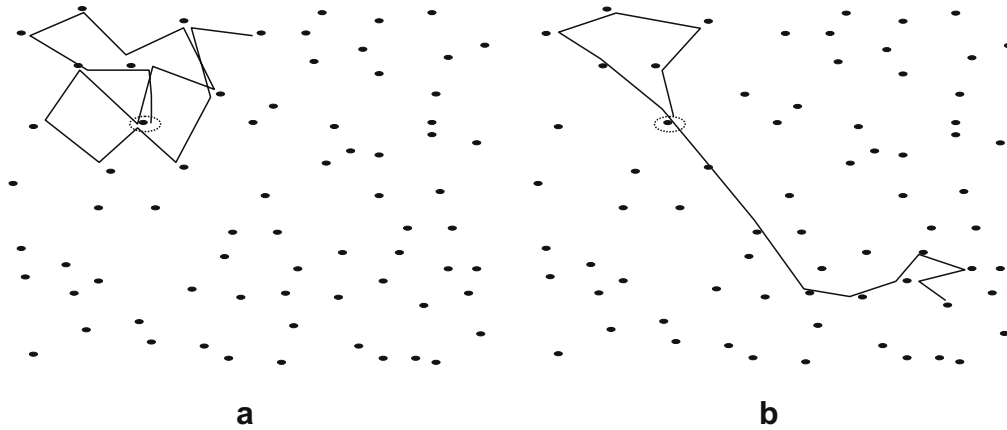
Fig. 1. An agent (starting from the initiator node depicted in dotted ellipsis) under: (a) the random walk without backtracking mechanism and (b) the J-RW mechanism.
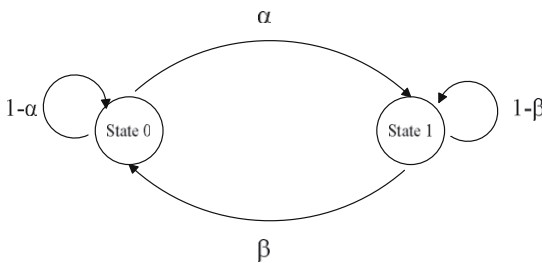

Fig. 2. Markov chain mechanism for controlling J-RW-agent movements.

fixated; it can be calculated as a vector whose direction coincides with the direction of the line connecting the current node and the node visited by the agent in the previous discrete time, in the direction away from the previously visited node. The J-RW-agent implements a directional walk during all subsequent node visits (while the markov chain stays in *State* 1), by selecting as the next node to visit to be the neighbor of the current node that is the closest (in direction) to the fixated vector. The J-RW-agent completes the jump in the network when the steering markov chain returns to *State* 0.

The directional walk may be easily implemented through a simple look up table involving the geographic locations of the neighbours of a node; this table determines the next node to forward the agent to under the directional walk, given that the agent came to this node from a given neighbour. The geographic information can be easily retrieved either at the time of deployment in the case of a static sensor field (with provisions for second, third, etc. choices when lower order choices are not available due to battery depletion), or after the deployment of the field with the help of a limited scope localization protocol run occasionally.

Let α (β) denote the transition probabilities from *State* 0 to *State* 1 (*State* 1 to *State* 0) and let $T_0 = 1/\alpha$ ($T_1 = 1/\beta$) denote the mean time (in discrete times of our reference time, or number of visits to nodes) that the agent spends in *State* 0 (*State* 1). Clearly, β (or, T_1) determines the length of time over which the directional walk is continuously in effect and, thus, the mean length of the induced jump. Similarly, α (or, T_0) determines the length of time over which the RW mechanism is continuously in effect. It should be noted that α and β should be carefully selected so that the mix of the two distinct operations is effectively balanced. β should be such that the implemented jump is sufficiently large to move the agent away from the current locality that is likely to be covered by the operation at *State* 0, and on the other hand, it should not be too large in which case it would leave uncovered large areas or require

the random walk operation to operate long enough (at the increased cost of revisits) to cover the large areas between the start and the end of the jump. Similarly, α should be such that the time spent at *State* 0 be balanced so as to not over-cover or under-cover the current locality.

As previously for the RW mechanism, coverage and cover time under the J-RW mechanism may be defined in a similar manner, denoted by $C_j(t)$ and T_j, respectively. $C_j(t)$ is a non-decreasing function of t taking values between 0 (for $t = 0$) and 1 (for $t \geqslant T_j$).

4. Coverage analysis

This section analyzes coverage under the random walk without backtracking mechanism in order to extract useful information regarding the performance of the RW-agent and consequently to use these results for further understanding of the particulars of the J-RW mechanism. The analysis followed in this section is different than any other previous analysis in the best of the authors' knowledge.

4.1. Coverage for the RW-agent

The main aim here is to derive an analytical expression for $C_r(t)$, which will serve as a tool for further understanding of random walk based information dissemination. Lets assume that the network topology is fully connected (i.e., all nodes are connected to all other nodes). This is actually the case for large values of r_c in RGG. For example, for nodes scattered in the $[0,1] \times [0,1]$ 2-dimensional plane, any value of $r_c \geqslant \sqrt{2}$ ensures that there is a link among any pair of nodes.

In such a network, each time the RW-agent decides to move to a new neighbor node at time t (thus, arriving at time $t + 1$), coverage $C_r(t)$: (a) may increase ($C_r(t + 1) = C_r(t) + 1/N$), provided that the new node has not been covered previously; or (b) remain the same ($C_r(t + 1) = C_r(t)$), provided that the new node has already been covered. Note that at time t, in a fully connected network the RW-agent may select one out of $N - 2$ network nodes (i.e., all network nodes excluding the one the agent came from and the one that is currently located at). Since $1/N$ corresponds to the coverage contribution of the node the agent came from and $1/N$ to the coverage contribution of the node that is currently located at, then $C_r(t) - 2/N$ is the coverage corresponding to the remaining $N - 2$ nodes and eventually, $(N - 2) \times (C_r(t) - 2/N)$ corresponds to the number of nodes that have already been visited by the agent (excluding the one the agent came from and the one that is currently located at). For large values of N (which is typically the case

considered in this paper), $(N - 2) \times (C_r(t) - 2/N) \approx NC_r(t)$. Consequently, the probability to choose a node that has not been visited previously (and consequently increase coverage) is equal to the probability of selecting one out of the $N - NC_r(t)$ nodes that have not been visited previously, or $N(1 - C_r(t))/N = 1 - C_r(t)$. It is easily derived now that (on average) the increment of the coverage after a RW-agent moves at time t, is given by the probability $1 - C_r(t)$ that it moves to a node not visited before multiplied by $1/N$ which is the contribution to coverage by each node that is visited for the first time. That is,

$$C_r(t + 1) - C_r(t) = \frac{1}{N}(1 - C_r(t)). \tag{1}$$

Eq. (1) can be expressed in a more convenient form by switching from discrete to continuous time. Let t be continuous and let $\widetilde{C}_r(t)$ denote the corresponding continuous and increasing function of $C_r(t)$. The difference $C_r(t + 1) - C_r(t)$ can be approximated by $\frac{C_r(t_1) - C_r(t_0)}{t_1 - t_0} = \frac{dC_r(t)}{dt}$, for $t_0 < t_1$. Based on Eq. (1),

$$\frac{d\widetilde{C}_r(t)}{dt} = \frac{1}{N}(1 - \widetilde{C}_r(t)). \tag{2}$$

The derivative $\frac{d\widetilde{C}_r(t)}{dt}$ corresponds to the rate at which $\widetilde{C}_r(t)$ increases. Obviously, for $t = 0$ (i.e., the RW-agent is about to start moving in the network), $\frac{dC_r(t)}{dt} = 1$, since during its first step the RW-agent will move to a node definitely not covered previously. $\frac{dC_r(t)}{dt}$ will eventually become zero, when all nodes are covered or, $\widetilde{C}_r(t) = 1$.

Eq. (2) is a first class differential equation, and the solution satisfying the previous properties of $\widetilde{C}_r(t)$ (e.g., increasing, $0 \leqslant \widetilde{C}_r(t) \leqslant 1$), is given by,

$$\widetilde{C}_r(t) = 1 - e^{-\frac{t}{N}}. \tag{3}$$

For convenience of the presentation, very frequently in the sequel, the normalized version $\widetilde{C}_r(t/N) = 1 - e^{-t}$ will be used instead of $\widetilde{C}_r(t)$.

In the literature, a well known analytical result for the complete graph case, is that cover time T_r is of the order of $N \ln(N)$, [23]. Using Eq. (3) it is now easy to derive that $\widetilde{C}_r(N \ln(N)) = 1 - e^{-\frac{N \ln(N)}{N}} = 1 - \frac{1}{N}$. For large values of N it is clear that $\widetilde{C}_r(N \ln(N)) \rightarrow 1$. Another interesting result is that for $t = N$, $\widetilde{C}_r(N) = 1 - e^{-\frac{N}{N}} = 1 - e^{-1} = 0.632$, which basically means that for movements equal to the number of the network nodes, on average 63.2% of the network nodes are visited by the RW-agent. These analytical findings are confirmed later by simulation results presented in Section 5.

Eq. (3) was derived assuming a fully connected network. By reducing r_c in RGG, the number of neighbor nodes decreases and therefore a RW-agent has fewer choices to move than before. Therefore, the fraction of nodes that have (not) been visited previously, is expected to deviate from $C_r(t)$ (or $1 - C_r(t)$). As r_c decreases even further, it is getting more and more difficult for the agent to move to a node not previously visited (due to the fewer

choices of movement), thus increasing the number of revisits and eventually, decreasing the actual rate under which $C_r(t)$ increases. For example, bottlenecks tend to appear in the network topology, [17], likely "forcing" a RW-agent to keep (re)visiting a comparably small number of nodes for a long time.

In order to account for the aforementioned decrease in the increase rate of $\widetilde{C}_r(t)$, it is assumed in the sequel that derivative $\frac{d\widetilde{C}_r(t)}{dt}$ is given by Eq. (4),

$$\frac{d\widetilde{C}_r(t)}{dt} = \frac{k}{N}(1 - \widetilde{C}_r(t)), \tag{4}$$

where $k \leqslant 1$ is a positive constant related to the topology characteristics (i.e., connectivity radius r_c and number of nodes N). The solution of Eq. (4) is given by,

$$\widetilde{C}_r(t) = 1 - e^{-\frac{kt}{N}}. \tag{5}$$

The case of $k = 1$ corresponds to the fully connected network topology (i.e., large values of r_c) as it is concluded from Eq. (3). Therefore, smaller values of r_c (however large enough for the network to be connected) should result in smaller values of k. This will be further explored and evaluated using simulation results presented later in Section 5.

Fig. 3a depicts $\widetilde{C}_r(t/N)$ as it is given by Eq. (5) as a function of t/N for various values of k. It is interesting to note that as k decreases, cover time increases. For example, for $t = \frac{N}{k} \ln(N)$, $\widetilde{C}_r(\frac{N}{k} \ln(N)) = 1 - \frac{1}{N}$ which tends to zero for large values of N. Let $\widetilde{T}_r = \frac{N}{k} \ln(N)$ be referred to hereafter as the *asymptotic* cover time for the RW-agent (i.e., $\lim_{N \rightarrow +\infty} \widetilde{C}_r(\widetilde{T}_r) = 1$). Given that $k < 1$, it is evident that as r_c decreases (the network has fewer links), k decreases and therefore, the asymptotic cover time $\widetilde{T}_r$ increases by a factor $\frac{1}{k}$.

This observation is graphically presented in Fig. 3. It is evident that the first derivative of coverage is high at the beginning and then it becomes significantly small, particularly for values of k close to 1. Along with $\widetilde{C}_r(t/N)$ in Fig. 3a, a (dotted) line corresponding to $\widetilde{C}_r(t/N) = t/N$ is also depicted. This line corresponds to the best (even though frequently not realistic) dissemination information scenario (i.e., the – artificial – case in which a node not previously visited is reached after each step). It is interesting to note that the particular case of $k = 1$ is the one most close to this best scenario. This is basically the case since for some time at the beginning (small values of t/N) $\frac{dC_r(t/N)}{dt/N}$ remains close to 1 for $k = 1$, as it is also depicted in Fig. 3b. Afterwards, $\frac{dC_r(t/N)}{dt/N}$ decreases (more rapidly than for cases of smaller k) due to frequent revisits of the RW-agent. Simulation results are presented later in Section 5 demonstrating the accuracy of the previous analysis.

4.2. Coverage for the J-RW-agent

There are basically two issues that need to be explored with respect to the J-RW mechanism presented in Section 3: (a) for how
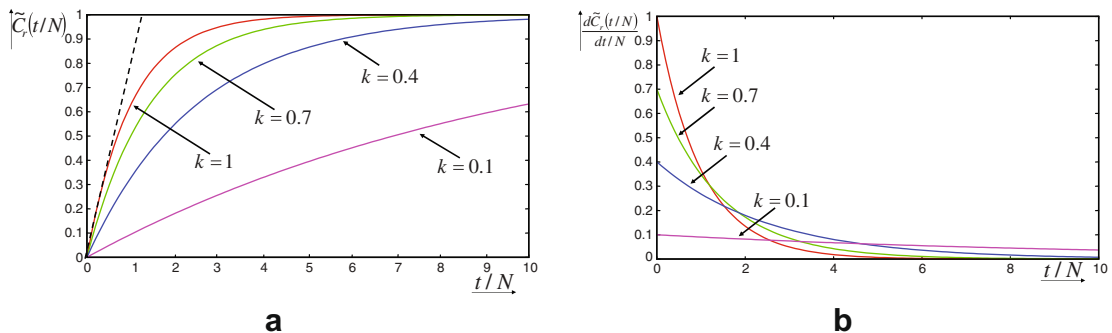

a

b

Fig. 3. $\widetilde{C}_r(t/N)$ and $\frac{d\widetilde{C}_r(t/N)}{dt/N}$ as a function of t/N for various values of k.

long should the J-RW-agent stay in *State* 0 (governed by α); and (b) for how long should the J-RW-agent stay at *State* 1 (governed by β).

Let us assume that α is close to 0. This is a trivial case in which the J-RW mechanism resembles the RW mechanism. This choice is expected to be a suitable one in topologies of highly connected nodes (large number of neighbors) such that the agent is allowed to move to nodes residing away from the already covered network parts. This may happen due to its own probabilistic movement within the network. An obvious example is the fully connected topology, but as it will be shown later using simulation results, this is also applied for topologies of smaller connectivity (i.e., smaller values of r_c) than the fully connected topology.

On the other hand, values of α close to 1, mean that the agent will not stay long in *State* 0. This eventually means that the agent will not be allowed a large time period to cover a certain network part. In this case, the agent will be mostly "jumping" (depending on the value of β) to different network parts thus failing to exhaustively cover network areas. This resembles a RW-agent moving over an overlay topology of longer links.

The connectivity of the topology (which for the case of RGGs is related to r_c) plays an important role for the investigation of the appropriate values of β with respect to coverage under the J-RW mechanism. Highly connected topologies (i.e., comparably high values of r_c but not necessarily close to the particular value that the topology is fully connected) are characterized by significantly small diameters, [17]. In such networks, a RW-agent is not expected to be limited within certain network parts and therefore, jumping would not improve coverage. In such cases, β should be close to 1. On less connected topologies (i.e., topologies of small values of r_c but large enough for the network to be connected) it is expected the RW-agent to frequently revisit nodes due to the topology's structure (e.g., bottlenecks). In such a case, β should be small enough to allow the agent to move to different network parts. On the other hand, too small values of β (e.g., β close to 0) will eventually result to an agent mostly operating in *State* 1 (i.e., jumping) at the expense of exploring more thoroughly the visiting part of the network.

As it can be concluded from the previous discussion, coverage under the J-RW mechanism is related to r_c, α and β. However, an analytical expression for the coverage considering r_c, α and β is difficult to be derived and its further investigation will be based on simulations presented in the following section. Let $C_j(t)$ ($\widetilde{C}_j(t)$) denote the coverage under the J-RW mechanism in a similar way as $C_r(t)$ ($\widetilde{C}_r(t)$) denotes coverage under the RW mechanism. After a long time, it is reasonable to assume that the J-RW-agent will have moved to all different network parts and have covered (on average) the same proportion of network nodes within each visited network part. Therefore, at time t it is expected that (on average) the fraction of non-visited neighbor nodes of the node that the agent is located at, will be $N \times (1 - C_j(t))$. Following an analysis similar to the one presented for the case of the RW mechanism,

the following analytical expression for coverage under the J-RW can be written

$$\widetilde{C}_j(t) = 1 - e^{-\frac{k't}{N}}, \qquad (6)$$

where k' is a positive constant depending on the particulars of the topology (i.e., r_c) and the J-RW mechanism (i.e., α and β), as it will be also shown in the following section using simulation results.

5. Simulation results and evaluation

A simulation program exploiting the capabilities of the Omnet++ simulation platform, [25], was created for the simulation purposes. The aim of the simulation results presented in this section is twofold: to confirm the analytical findings of the previous section and to shed more light on the behavior of the J-RW mechanism (mostly in comparison to the RW mechanism) for cases not covered by the analysis.

There are multiple simulation runs executed under specific sets of parameters for the network and the investigated schemes. During each simulation run there is a large-scale node set up, with node population varying from 100 to 3000 nodes depending on the case. The nodes are placed at random locations on a square plain $[0,1] \times [0,1]$. The random positions (x_u, y_u) of each node $u \in V$ are chosen within the set $[0,1]$ using the uniform probability distribution. Each node u is aware about its own position: (x_u, y_u). Each node is connected to some other node if the euclidean distance among them is less or equal to r_c. Clearly, for $r_c \geqslant \sqrt{2}$, the resulting network is fully connected. Depending on N (the size of the network), the lower bound of r_c for which the topology remains connected varies (typically decreases as N increases). Four different values of r_c (0.05, 0.1, 0.5 and 1.0) are considered in the sequel for those topologies of $N = 1000$. Note that all four values are less than $\sqrt{2} \approx 1.4$, since a fully connected network is not representative of the wireless environment (e.g., wireless sensor networks) that is considered here. Networks with values of $N \geqslant 10^4$ were impossible to be simulated due to restrictions in memory imposed by our simulator.

Coverage is the main focus in the result to be presented. These results correspond to one experimentation instance (apart from some cases which are explicitly mentioned) and not averaged values. Averaging would have given a macroscopic view but it would also hide important details.

5.1. The RW-agent

An important contribution in the analytical part of this paper, is the derivation of the coverage in a fully connected network (i.e., $r_c \geqslant \sqrt{2}$), shown in Eq. (3). Fig. 4a presents coverage as a function of t (i.e. the number of movements of the random walker) in fully connected network topologies of 100, 500, 1000 and 3000 nodes. For each network topology, the analytically derived value $\widetilde{C}_r(t)$ is
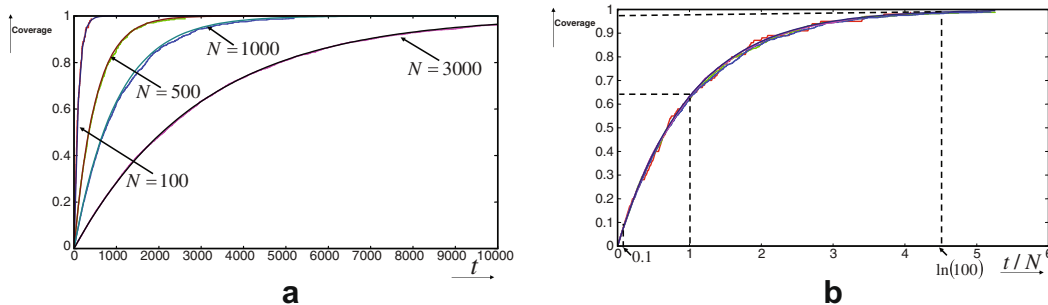

Fig. 4. Coverage and normalized coverage under the random walk without backtracking mechanism for fully connected topologies of various network sizes.

also depicted. It is important to note that the analytical findings are almost identical to the simulation results. Fig. 4b presents (normalized) coverage as a function of t/N. As before, it is demonstrated that $\tilde{C}_r(t/N)$ is almost identical to $C_r(t/N)$ as it is concluded by observing simulation results for the case of a fully connected topology.

The fact that all curves follow the same pattern as it is illustrated in Fig. 4b allows for certain observations. First, it is obvious that for about $0.1N$ movements (see dotted line at $t/N = 0.1$ depicted in Fig. 4b), the random walker has a very good performance (the number of revisits remains small) in the sense that a new movement most likely results in visiting a node that has not been visited before. At time $t = N$, it is interesting to see that for almost all cases, 63.2% of the total number of network nodes has been covered as it was expected from the analysis presented in the previous section. For $N = 100$, $t = N \ln(N)$ corresponds to $t/N = \ln(100) \approx 4.6$ in Fig. 4b. As expected from the analysis (and also depicted in Fig. 4b), coverage is about $1 - 1/N = 99\%$, which is very close to that depicted using simulation results.

Fig. 5 presents simulation results for various topologies derived for $r_c = 0.05,\ 0.1,\ 0.5$ and 1.0. The first observation is that for the
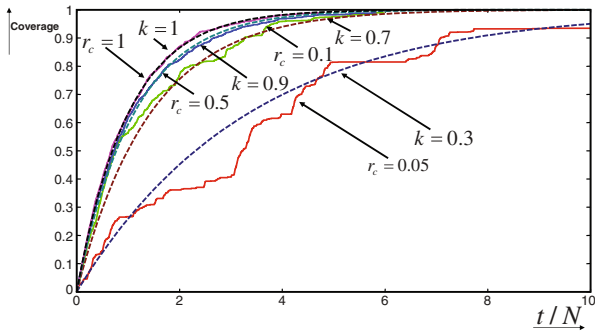
appropriate value of k (i.e., $k = 0.3$ for $r_c = 0.05$, $k = 0.7$ for $r_c = 0.1$, $k = 0.9$ for $r_c = 0.5$ and $k = 1.0$ for $r_c = 1.0$), the analytical expression for coverage, given by Eq. (5) (i.e., $\tilde{C}_r(t) = 1 - e^{-kt}$) approximates well the simulation results. Another interesting observation is that for $r_c > 0.1$, the appropriate value of k is very close to 1.0, This is basically due to the fact that the number of neighbors (on average) increases much faster than r_c (it is $\pi r_c^2 N$ on average) and, thus, a small increase in r_c results in a large increase of the node degree (number of neighbors). For $N = 1000$ and $r_c = 0.1$, there are (on average) about 31 neighbor nodes for each node which implies that the topology is highly connected. As r_c increases further, it is interesting to observe that the coverage approaches closely $1 - e^{-t}$, even for cases for which r_c is significantly smaller that $\sqrt{2} \approx 1.4$.

5.2. The J-RW-agent

Fig. 6 presents simulation results under the J-RW mechanism for a network of 1000 nodes and various values of r_c and α. β has been kept constant and equal to 0.4, which means that as soon as *State* 1 is assumed (i.e., directional movement) the agent moves (on average) for 2–3 nodes towards a certain direction (more details are provided in the description of the J-RW mechanism in Section 3) before *State* 0 is assumed. In Fig. 6a, coverage under the RW mechanism is clearly depicted and it is less than the coverage under J-RW for any value of α (e.g., 0.2, 0.4, 0.6 and 0.8). Note that the case depicted in Fig. 6a corresponds to a topology that is not highly connected ($r_c = 0.05$), thus even a relatively small value of $\beta = 0.4$ results in the J-RW doing significantly long jumps to get a performance improvement.

As the topology becomes more connected (r_c increases), the advantage of the J-RW mechanism is less obvious. For example, for the case depicted in Fig. 6b ($r_c = 0.1$), coverage under the random walk without backtracking mechanism is still smaller than that under the J-RW mechanism (for any value of α), even though not that smaller as before, while for the case depicted in Fig. 6c ($r_c = 0.5$), coverage under the random walk without backtracking mechanism
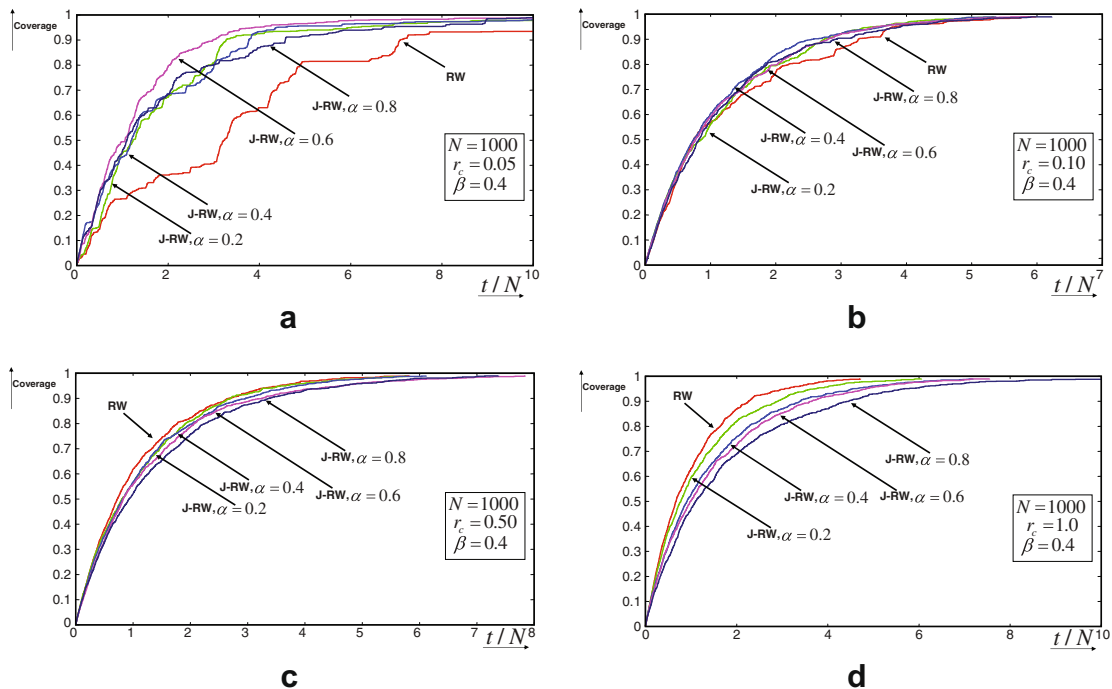

Fig. 5. Coverage for topologies of 1000 nodes and various values of r_c (0.05, 0.1, 0.5 and 1.0). $\tilde{C}_r(t)$ is also depicted for corresponding values of k (0.3, 0.7, 0.9 and 1.0).

Fig. 6. Coverage for various values of α ($\beta = 0.4$) as a function of t/N.
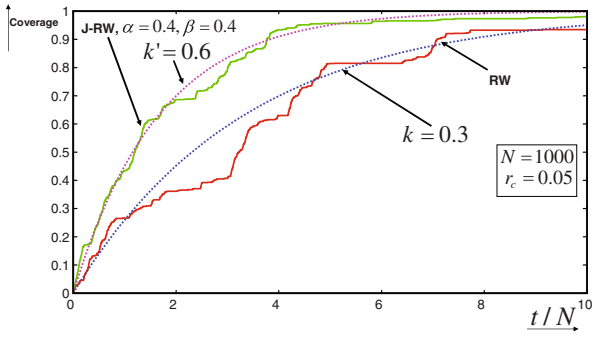
Fig. 7. Simulation and analytical results for a topology of 1000 nodes under the random walk without backtracking and the J-RW mechanism.

is now larger than that under the J-RW mechanism (for any value of α). As r_c increases further, coverage under the random walk without backtracking mechanism is clearly higher than that under the J-RW mechanism for the specific combination of values of α and β. This is clearly depicted in Fig. 6d for the case of $r_c = 1.0$ and can be attributed to the fact that the random walk without backtracking mechanism can now fully exploit the increased connectivity of the graph. In particular, subsequent movements of the RW-agent in highly connected graphs are similar to J-RW-agent movements in low connectivity graphs (in terms of how far in physical distance the agent moves). Thus, there is no coverage benefit when introducing J-RW agent in highly connected graphs (as compared with RW-agent), on the contrary the J-RW mechanism of "locking" in *State* 1 tends to push the agents away towards the physical boundaries of the examined network and thus result in an actual decrease in network coverage.

Another interesting aspect is the shape of the curve corresponding to the J-RW mechanism. As it can be observed in Fig. 6, it clearly follows the pattern of the analytical expression $1 - e^{-\frac{k'}{N}t}$, for suitably selected values of k', as it was already mentioned in Section 4. This is more clearly seen in the simulation results depicted in Fig. 7 along with plots for $\widetilde{C}_r(t/N)$ and $\widetilde{C}_j(t/N)$. It is obvi-

ous that simulation results follow the same pattern for the J-RW mechanism as it is also the case (also shown before) for the random walk without backtracking mechanism.

The particular value of β that was used for the simulation results depicted in Fig. 6 was fixed ($\beta = 0.4$). Fig. 8 presents simulation results (coverage) as a function of both α and β (their values in the range [0.2,0.8]) at time $t = N$ and Fig. 9 at time $t = N \ln(N)$.

The results depicted in Fig. 8a correspond to $r_c = 0.05$. About 25% of the network is covered under the random walk without backtracking mechanism while coverage for this case under the J-RW mechanism varies from 30% to 55% depending on the particular values of α and β. For $r_c = 0.1$, as depicted in Fig. 8b, coverage under the random walk without backtracking mechanism is again smaller than that under the J-RW mechanism for any selection of α and β. In Fig. 8c ($r_c = 0.5$), it is interesting to see that coverage under the random walk without backtracking mechanism is greater than that under the J-RW mechanism, apart from those cases that β is high (e.g., 0.8) and α is small (e.g., 0.2). In such cases the high value of β results in relatively small jumps within the network, which are small enough to be effective in such a highly connected network. Any smaller value of β would result in larger jumps within the highly connected graph, 'pushing' again the J-RW agents towards the boundaries of the network and reducing performance. Thus the J-RW-agent is able to move now between different 'neighbor areas' and explore them efficiently.

As r_c increases further, as it is the case in Fig. 8c for $r_c = 1.0$, it is obvious that there is no combination of α and β in the given range of values such that coverage under the J-RW mechanism be greater than coverage under the random walk without backtracking mechanism. At this point it is important to note that the J-RW mechanism becomes equivalent to the random walk without backtracking mechanism for $\alpha = 0$ and $\beta = 1$ (which means that the mechanism always stays at *State* 0).

Simulation results for $t = N \ln(N)$ are depicted in Fig. 9. The observations are identical to those regarding Fig. 8 apart from the fact that coverage is close to 1 (which is expected given the analytical results). Similarly to the case of $t = N$, for $t = N \ln(N)$ coverage under the J-RW mechanism is larger than that under
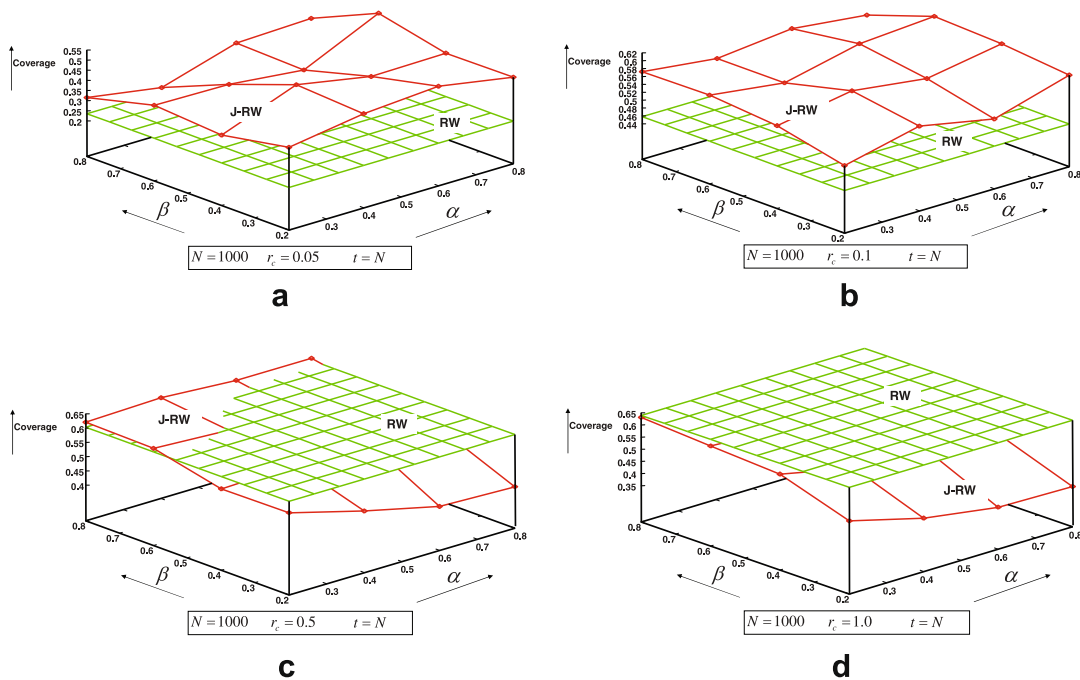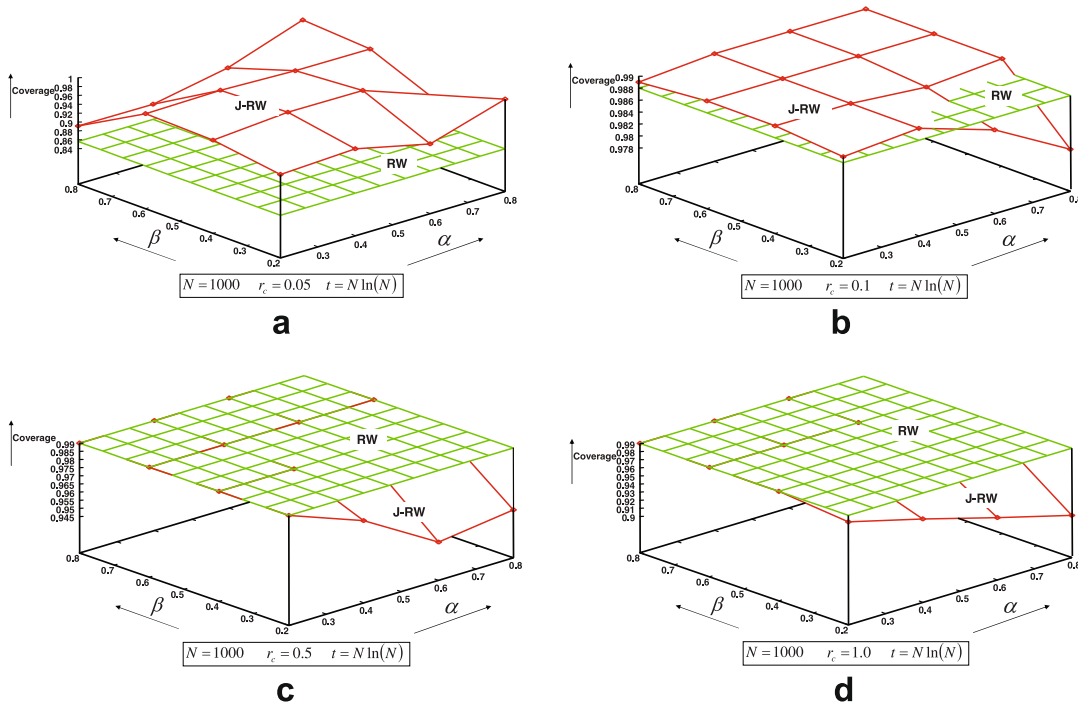

Fig. 8. Coverage for various values of α and β at $t = N$.

Fig. 9. Coverage for various values of α and β at $t = N \ln(N)$.

the RW for topologies of small values of r_c. It is interesting to observe in Fig. 9a that for large values of both α and β (e.g., 0.8) coverage under the J-RW can be close to 98%, which is a significant improvement when compared to coverage under the random walk without backtracking mechanism that is close to 86%. In all other cases and particularly those for which coverage under the random walk without backtracking mechanism is close to 100%, coverage under the J-RW mechanism is close to 100% for values of β not that small (long jumps should be avoided in such highly connected topologies). Note that values for α and β around 0.5, appear to be an appropriate selection, as it is observed from the simulation results depicted in both Figs. 8 and 9, since: (a) for topologies of small r_c, coverage under the random walk without backtracking mechanism is greater than coverage under the J-RW mechanism; and (b) for all other cases, coverage is almost the same under either mechanism.

6. Conclusions

Random walk based solutions have been proposed for information dissemination in large-scale wireless sensor and ad hoc networks. However, such random walk based agents are prone to inefficiencies due to frequent revisits to already covered nodes, resulting in relatively large network cover time. In this paper a new class of random walk based agents, referred to as the Jumping Random Walk agent has been introduced. As this new class contains the (classical) random walk without backtracking agent (for $\alpha = 0$) it can only improve on the performance of the RW-agent or – in the worst case – achieve that of the RW-agent.

The coverage (i.e., the percentage of network nodes visited by the disseminating agent) as a function of time is studied both analytically and through simulations in this paper. Geometric random graph topologies are used to model the underlying network topology, which is a suitable abstraction for the aforementioned wireless environment. The analytical results are confirmed through simulations and it is shown that the J-RW-agent, other than the RW-agent yields the best performance in not highly connected topologies. The comparative performance of the RW-agent and other members from the proposed J-RW-agent class is thoroughly investigated through numerous results and discussion that shed light into the intrinsic capabilities of these mechanisms and the network conditions that affect their performance.

References

[1] S. Dolev, E. Schiller, J. Welch, Random walk for self-stabilizing group communication in ad hoc networks, IEEE Trans. Mobile Comput. 5 (7) (2006).

[2] C. Avin, B. Krishnamachari, The power of choice in random walks: an empirical study, in: Proceedings of the 9th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, (MSWiM), Malaga, Spain, 2006.

[3] D. Braginsky, Rumor routing algorithm for sensor networks, in: WSNA '02: 1st ACM Workshop on Wireless Sensor Networks and Applications, Atlanta, Georgia, 2002.

[4] U. Feige, A spectrum of time–space trade-offs for undirected s-t connectivity, J. Comput. Syst. Sci. 54 (2) (1997) 305–316.

[5] A. Segal, Distributed network protocols, IEEE Trans. Inform. Theory IT (29) (1983) 23–35.

[6] B. Williams, T. Camp, Comparison of broadcasting techniques for mobile ad hoc networks, in: Proceedings of MOBIHOC'02, Lausanne, Switzerland, 2002, pp. 194–202.

[7] K. Oikonomou, I. Stavrakakis, Performance analysis of probabilistic flooding using random graphs, in: 1st IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC'07), Helsinki, Finland, 2007.

[8] M.B. Yassein, M. Ould-Khaoua, S. Papanastasiou, On the performance of probabilistic flooding in mobile ad hoc networks, in: 11th International Conference on Parallel and Distributed Systems (ICPADS'05), 2005.

[9] D. Kogias, K. Oikonomou, I. Stavrakakis, Study of randomly replicated random walks for information dissemination over various network topologies, in: Proceedings of WONS 2009), Snowbird, Utah, USA, 2009.

[10] I. Mabrouki, X. Lagrange, G. Froc, Random walk based routing protocol for wireless sensor networks, in: Proc. of 2nd international Conference on Performance Evaluation Methodologies and Tools (ValueTools'07), Nantes, France, 2007.

[11] A.D. Sarma, D. Nanongkai, G. Pandurangan, Fast distributed random walks, in: Proceedings of the 28th ACM Symposium on Principles of Distributed Computing (PODC), Alberta, Canada, 2009.

[12] N. Bisnik, A.A. Abouzeid, Optimizing random walk search algorithms in p2p networks, Computer Networks 51 (6) (2007).

[13] D. Tsoumakos, N. Roussopoulos, Adaptive probabilistic search for peer-to-peer networks, in: Proceedings of 3rd IEEE International Conference on P2P Computing, Linkoping, Sweden, 2003.

[14] T. Lin, H. Wang, Search performance analysis in peer to peer networks, in: 3rd International Conference on Peer-To-Peer Computing (P2P'03), 2003, pp. 82–83.

[15] Q. Lv, P. Cao, E. Cohen, K. Li, S. Shenker, Search and replication in unstructured peer-to-peer networks, in: ACM SIGMETRICS 02, 2002, pp. 258–259.

[16] C. Avin, C. Brito, Efficient and robust query processing in dynamic environments using random walk techniques, in: Proceedings of the IPSN 2004, Berkeley, California, 2004.

[17] M. Penrose, Random Geometric Graphs, Oxford Studies in probability, Oxford University Press, 2003.

[18] R. Albert, A.-L. Barabási, Statistical mechanics of complex networks, Rev. Mod. Phys. 74 (1) (2002) 47–97, doi:10.1103/RevModPhys.74.47.

[19] L. Tzevelekas, I. Stavrakakis, Improving partial cover of random walks in large scale wireless sensor networks, in: 3rd IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications(AOC'09), Kos, Greece, 2009.

[20] J. Kleinberg, The small-world phenomenon: an algorithmic perspective, in: STOC'00: Proceedings of the 32nd ACM Symposium on Theory of Computing, Portland, Oregon, United States, 2000, pp. 163–170.

[21] C. Gkantsidis, M. Mihail, A. Saberi, Random walks in peer-to-peer networks, in: Proceedings of INFOCOM'04, Hong Kong, 2004.

[22] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, S. Shenker, Making gnutella-like p2p systems scalable, in: SIGCOMM '03: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM, New York, NY, USA, 2003, pp. 407–418.

[23] L. Lov'asz, Random walks on graphs: a survey, combinatorics, Paul Erdos is eighty, J. Bolyai Math. Soc. II (2) (1996) 353–397.

[24] C. Avin, G. Ercal, On the cover time of random geometric graphs, in: ICALP 2005, 2005, pp. 677–689.

[25] %3chttp://www.omnetpp.org%3e.