

# Replicated Random Walks for Service Advertising in Unstructured Environments

Dimitris Kogias, Konstantinos Oikonomou, and Ioannis Stavrakakis

**Abstract** Service advertisement is a key design issue in modern dynamic and large-scale networking environments such as unstructured peer-to-peer networks. The intrinsic capability of a *single random walker* of *stretching* the information dissemination over widely spread network areas (compared to *flooding*), is explored and exploited in this paper, along with the introduction of random walkers which can *replicate* themselves. Two *replication policies* are also introduced in this paper: the *Topology Independent Policy* that creates replicas according to an exponentially decreasing probability (creating more replicas at the beginning of the advertising process), and the *Topology Dependent Policy* in which replication decisions are based on some locally available topological information (aiming at creating replicas at the dense network areas). The discussion and the results in this paper reveal intrinsic comparative properties of flooding and the single random walker, as well as the advantages that the random walker replication can bring in improving the overhead, speed and coverage of the advertising process.

**Acknowledgements** This work has been supported in part by the project ANA (Autonomic Network Architecture) (IST-27489), the PENED 2003 program of the General Secretariat for Research and Technology (GSRT) co-financed by the European Social Funds (75%) and by national sources (25%) and the NoE CONTENT (IST-384239).

---

Dimitris Kogias · Ioannis Stavrakakis  
National And Kapodistrian  
University of Athens, Department of Informatics & Telecommunications  
Panepistimiopolis, Ilissia 15 784, Athens, Greece  
Tel: +30 210 7275315, Fax: +30 210 7275333  
e-mail: {dimkog, ioannis}@di.uoa.gr

Kostantinos Oikonomou  
Ionian University  
Department of Informatics  
Corfu, Greece  
e-mail: okon@ionio.gr

---

*Please use the following format when citing this chapter:*

Kogias, D., Oikonomou, K., Stavrakakis, I., 2008, in IFIP International Federation for Information Processing, Volume 265, Advances in Ad Hoc Networking, eds. Cuenca, P., Guerrero C., Puigianer, R., Serra, B., (Boston: Springer), pp. 25-36.

## 1 Introduction and Motivation

The evolution of the *Peer-to-Peer* (P2P) networks has attracted a great deal of attention lately, becoming an environment suitable for numerous networked services (e.g. file or service sharing) mostly due to their low cost sharing of these services. These environments are highly dynamic and of large scale since a great number of users join and leave the network at any time, exchanging a large amount of data and sharing a huge number of services/files. *Unstructured* P2P networks, e.g. Gnutella, [1], have become popular, mostly because they are easily formed without the need for any sophisticated configuration methods. These environments present some major design challenges due to the aforementioned lack of structure. One key challenge (that is actually the focus of this paper) is *service discovery*. That is, the design of the mechanism that allows user nodes to *discover* the *service location* or, equivalently, the location of a *service node* (i.e., the node that hosts a particular service).

The effectiveness or the required intensity of a service discovery mechanism depends strongly on the service location information availability within the network. If only the service node knows about the service location, then the service discovery mechanism should launch a service location search process that will need to “hit” the service node itself in order to retrieve the service location information. If, on the other hand, more nodes in the network are aware of the service location, then the *service searching* task during the service discovery can be less intense or more effective (e.g., time-wise), as it would be sufficient for the searching agent (or packet) to hit one of the (more than one) nodes that possess the service location information. Informing other than the service node of the location of the service is the function of the *service advertising* process or mechanism that basically disseminates the service location in the network.

At the end of a service advertising process a number of network nodes becomes aware of the service location. This set of (informed) nodes will be referred to here as the *advertising network* that has been generated by the specific advertising process. It is important to emphasize that the effectiveness or efficiency of the service discovery process will depend not only on the size of the advertising network but also on its *stretch*; loosely speaking (to be defined in a specific way later), the stretch of an advertising network would reflect how spread out within the network are the nodes of the advertising network. It would, also, reflect the distance in hops of an average network node (which could decide to search for the service location and look for a node of the advertising network) from the advertising network (the smaller the distance the larger the stretch). Between two advertising networks with the same number of nodes, the one with the larger stretch would be considered to be more effective than the one that generated the advertising network of smaller stretch.

The focus on this paper is to explore service advertising (or information dissemination, in general) schemes for unstructured, large scale networks. More specifically, this paper investigates the effectiveness of (multiple) random walkers or *agents* in spreading the service location information effectively, with respect to certain criteria to be introduced later. One way to carry out service advertising is through traditional *flooding*, e.g., [2], [3]. In this case, the advertising network is the

entire network (largest coverage possible) and the *advertising network completion time* is small (upper bounded by the network diameter); on the negative side, though, flooding induces large message overhead. To reduce the large message overhead of traditional flooding, *probabilistic flooding* (where a message is forwarded with some probability less than 1) can be employed, [4], [5], [6], [7], [8], [9]. This scheme is capable of reducing the number of messages, at the expense of some increase in the advertising network completion time and possible decrease in the size of the advertising network, compared to (traditional) flooding, [6]. Another way to reduce the overhead of the traditional flooding is through *controlled flooding*, where the advertisement is confined to an area of some (predefined) number of hops (say  $K$ ) away from the service node; if  $K$  is very large this scheme approaches the traditional flooding.

A very different - from flooding - approach to implementing service advertising is through the employment of a *single random walker*, e.g. [10], [11], [12]. Under this approach, a single agent moves randomly in the network informing the network nodes on its path about the service location. As it will be discussed later, this approach tends to create advertising networks of typically larger stretch than those generated by some equivalent (in terms of overhead) flooding approach. On the other hand, completion time is higher (equal to the number of messages).

Flooding and single random walker can be viewed as two rather “extreme” dissemination approaches. For example, when the amount of affordable messages in the network (denoted as  $H$ ) - to be viewed also as the amount of the “advertising budget” - is large (larger than the number of network links), flooding is the best choice (the advertising network is the entire network with small completion time), as opposed to the single random walker. On the other hand, when  $H$  is relatively small, a single random walker is expected to create advertising networks of larger stretch than under flooding at the expense of larger completion time.

This good performance of flooding for a large advertising budget, as discussed above, may be attributed to the large number of “agents” that work in parallel and independently in order to create the advertising network. On the other hand, only one agent undertakes the task of creating the advertising network under the single random walker approach, which makes the creation slow; furthermore, it does not facilitate the wider spreading of the coverage (that could materialize due to the large advertising budget), as it would need to be implemented through one agent only in a sequential manner.

The introduction of “multiple” random walker approaches could enhance the performance of single random walker, in the sense that the size, the stretch and the completion time of an advertising network could be improved. To this end, the idea of *agent replication* is introduced in this paper that allows an agent, moving in the network according to the random walker paradigm, to create a replica or a *child* of itself. The introduction of a new agent in the network is expected to cover (probabilistically) a different area (by initiating its independent path) than that of its *parent* agent. For a given advertising budget, the larger the number of agents generated, the smaller their individual advertising budget would be, since it would be a portion of the original and the coverage would be limited to nearby areas only. Consequently,

the effective number of replicas should depend on the total advertising budget and increase with it. Clearly, the employed *agent replication policy* shapes accordingly the resulting advertising network and its completion time.

In view of the above discussion, a replication policy, to be referred to hereafter as *Topology Independent Policy* or *TI-Policy*, is proposed in this paper under which agent replication takes place according to some probability that decreases exponentially after each replication. When the initial value  $p$  of this probability is relatively large, it would create replicas relatively soon; the exponential decrease ensures, though, that replications will be rare afterwards, thus allowing agents to move for some time and cover wider network areas. On the other, small values of probability  $p$  would result in a behavior similar to the single random walker case.

Another replication policy that would make sense could be one that creates replicas in the *dense* network areas (i.e., network areas of nodes with relatively high number of neighbor nodes). In such areas it is reasonable to assume that new agents are more likely to follow a different “network direction” compared to that of their parent agent, thus potentially increasing the stretch and size of the coverage area. On the other hand, in less dense network areas, replication should be avoided since it is likely to result in an (undesired) “overlapping” of the covered network areas by different agents. To this end, the *Topology Dependent Policy* or *TD-Policy* is also presented in this paper.

The performance of the proposed algorithms is evaluated through simulations and compared to that of single random walker and flooding. It is shown that, as long as advertising budget  $H$  is not extremely high (in which case flooding should be employed) or low (in which case single random walker should be employed), there is a range of values for  $q$  and  $p$  for which the proposed policies are capable of covering a certain part of the network faster (i.e., smaller completion time) and more efficiently (i.e., stretched).

For the rest of the paper, a formal definition of the network system is given in Section 2. The proposed replication policies are described in Section 3 and simulation results are presented in Section 4. Finally, a summary and the conclusions can be found in Section 5.

## 2 System Definition

Let the undirected connected graph  $G(V, E)$  represent a network with a certain set of nodes  $V$  and a set of bidirectional edges  $E$  among nodes. Let  $d(u, v)$  denote the number of hops over a shortest path between node  $u$  and node  $v$ ; obviously,  $d(u, u) = 0$ . Let  $S_u$  denote the set of neighbor nodes of node  $u$ . Let  $|X|$  denote the number of elements or size of a particular set  $X$ . The number of nodes  $|V|$  in the network will also be noted as  $N$  ( $N = |V|$ ).

The service location node, denoted by  $s$ , is the initiator of the associated service advertising process aiming at disseminating service location information over a subset  $E_a$  of the network links  $E$  ( $E_a \subseteq E$ ) and eventually informing a subset  $V_a$

of the network nodes  $V$  ( $V_a \subseteq V$ ). The network consisting of nodes in  $V_a$  and links in  $E_a$ , denoted as  $G_a(V_a, E_a)$  is the advertising network. Obviously,  $G_a(V_a, E_a)$  is a connected network.

Many different advertising networks can be defined in a certain network, depending on the algorithm employed by the advertising process. The interest is to *stretch* the advertising network in order for the disseminated information to come as close as possible to all network nodes. Of course, in the exceptional case that information reaches all network nodes, the meaning of stretching becomes obsolete and no service searching is needed. In the general case, however, not all network nodes are part of the advertising network and, thus, a searching mechanism needs to be employed. A simple searching process that is assumed here, is implemented through controlled flooding of depth of  $L$  hops away from the node initiating the search. Clearly, if the advertising process is capable of creating an advertising network such that  $\min d(u, v) \leq L$ , for some node  $u \in V$  and at least one  $v \in V_a$ , then the searching process of node  $u$  will be successful.

Let the  $L$ -property of any network node  $u$  be defined as the existence of at least one node of the advertising network in at most  $L$  hops away from the particular node  $u$ . Let *coverage* of a particular advertising process for a certain  $L$ , denoted as  $C(L)$ , be defined as the proportion (%) of the network nodes for which the  $L$ -property is satisfied. For two different advertising networks (e.g.,  $G_a(V_a, E_a)$  and  $G'_a(V'_a, E'_a)$ ) of the same number of nodes ( $V_a \neq V'_a$ , and  $|V_a| = |V'_a|$ ), if  $C_1(L) > C_2(L)$ , then the advertising network  $G_a(V_a, E_a)$  is more stretched than the advertisement network  $G'_a(V'_a, E'_a)$ , since the information is closer (under the  $L$ -property notion) to more network nodes.

For example, under flooding if the advertising budget is large (exceeds a minimum value that depends on the network topology and size), all network nodes will be part of the advertising network. Under a single random walker, though, previous statement is not true, since no matter how large the advertising budget is, the “full” coverage cannot be guaranteed. For example, when the requirement is  $C(0) = 100\%$ , ( $L = 0$ ), and the available number of messages  $H$  is high, then under flooding it is “deterministically” assured that  $C(0) = 100\%$ , as well as a small completion time. Under single random walker on the other hand, a large completion time would be required (equal to the number of messages  $H$ ) but it is not guaranteed (only with high probability) that  $C(0) = 100\%$ .

There are some interesting properties about the  $L$ -property and coverage  $C(L)$ . For example, if for a node the  $L_1$ -property is satisfied for some  $L_1$ , then the  $L_2$ -property is also satisfied for any  $0 \leq L_1 < L_2$ . If the 0-property is satisfied for some node  $u$ , then node  $u$  belongs also to the advertising network ( $u \in V_a$ ). If the  $L$ -property is satisfied for all network nodes, then  $C(L) = 100\%$ , and vice versa. Furthermore, for any two values of  $L$ ,  $0 \leq L_2 < L_1$ ,  $C(L_1) \geq C(L_2)$ . Since  $L = 0$  is an exceptional case that has been addressed extensively in the literature, e.g., [10], [11], [12], the focus in this paper is mainly on  $L > 0$ .

The number of messages  $H$  used to create an advertising network as well as its completion time are two performance metrics (along with coverage) that will be used in the remaining of this paper. An interesting observation is that for the excep-

tional case of  $C(L) = 100\%$  (for some value of  $L > 0$ ) if minimization of the number of messages is required, the resulting advertising network is a *connected  $L$ -distance dominating set*, [13], of the initial  $G(V, E)$  network. The creation of such a set in a graph, is a *NP-complete* problem that requires *global knowledge*. Obviously, such solutions are not suitable for the considered unstructured, large-scale, and dynamic network environment. In any case, dominating sets are not within the scope of this paper.

### 3 Replication Policies

The service advertising policies presented in this section employ the *random walk* paradigm in order for an agent  $e$ , carrying information about the service location to be forwarded in the network, thus, creating an advertising network. The initial agent (denoted as  $e_0$ ) is created at the source node  $s$ . When an agent moves from a node  $u$  to another neighbor node  $v \in S_u$ , it is assumed that this particular movement takes place in one time unit (or time slot) and it corresponds to one message. When an agent arrives at node  $v$  from node  $u$  ( $v \in S_u$ ), it does not choose node  $u$  as its next hop destination, unless node  $u$  is the only neighbor node of  $v$ . Note that this rule does not prevent the agent from re-visiting a certain network node in the future. The initial agent has an advertising budget of  $H$  hops (corresponding to  $H$  messages). After each movement the remaining budget for agent  $e$ , denoted by  $h_e$ , is decreased by 1.

The core idea behind the policies proposed in this section is to allow for *creating replicas* of the agents (only one replica each time in this paper). Any new agent  $e_x$  (referred to as *child agent*) continues to move in the network just like its *parent agent*  $e_y$ . The number of allowed messages of the parent agent  $h_{e_y}$  (i.e., its advertising budget), is divided equally (or almost equally since  $h_{e_y}$  is an integer) among the child and the parent agent.

When only one agent is employed in the network, it is expected to achieve a certain value of coverage,  $C(L)$ , for some  $L > 0$  and for a certain number of messages  $H$ , in (completion) time equal to  $H$  time units. If a child agent is created (assuming half of the allowed number of messages of its parent agent), the completion time is expected to be reduced, while coverage  $C(L)$  may: (a) remain the same – for example, when both agents cover the same coverage area as the single agent; (b) increase – for example, when the agents cover different coverage areas and therefore, are capable of stretching the advertising network more than the case when a single agent was employed; (c) decrease – for example, when both agents cover the same coverage area but for a shorter time due to the division of the number of messages.

From the previous discussion it becomes evident that the coverage and/or completion time can be improved through *agent replication*. It should also be noted that *replication* may also have undesired results (e.g., coverage is decreased). For example, replication should not be too frequent but also not too rare. Frequent replications results in agents with small advertising budget (since  $h_e$  is divided after the repli-

cation of agent  $e$ ) that is mostly spent to move in largely overlapping, small areas. On the other hand, rare replication results in a behavior that is similar to a single random walker (i.e., large completion time, etc).

A reasonable approach is to allow replication to take place frequently at the beginning when they will have enough “steam” to move and create a stretched advertising network, while becoming rarer afterwards (giving multiple agents sufficient time to create stretched and less overlapping coverage areas). A second approach would be to allow frequent replications to take place in the “dense” areas of the network (assuming that it would be more likely for a child agent to cover different coverage areas than that of its parent agent), while not allowing for frequent replicas in the less “dense” areas of the network (otherwise, most likely both the parent and the child agent would cover the same area). The replication policies presented in the sequel employ the aforementioned approaches.

### 3.1 *The Topology Independent Policy (TI-Policy)*

A simple first idea is to have the agents replicate themselves according to some fixed probability  $p$ . If  $p$  is large, replication would be rather excessive, as each agent would have limited budget (i.e., initial budget is divided to them) and cover mostly overlapping areas. On the other hand, if  $p$  is very small replication would occur after a considerably long time, when no messages (budget) are left for the new agent to take advantage of. As already mentioned, a reasonable approach would be high replication at the initial stages of the advertising process and lower afterwards.

The proposed TI-Policy allows for the replication of an agent  $e$  according to the replication probability  $p(k_e) = p^{2^{k_e}}$ , where  $p$  is a constant probability (as before), and  $k_e$  is the number of replications that have taken place in the past for agent  $e$  and all its parent agents. For the initial entity  $e_0$ ,  $k_{e_0} = 0$ . Obviously, as the number of replications increases, the probability for a new replication decreases rapidly. This particular policy is easily applied requiring only that  $k_e$  is maintained along with agent  $e$  and made available to its child agents (if any).

### 3.2 *The Topology Dependent Policy (TD-Policy)*

The main aim under the TD-Policy is to exploit information that can be easily gathered by an agent during its movement in the network, in order to decide whether to replicate or not. As already mentioned, the goal is to allow replication in dense network areas and avoid it in non-dense areas. The information related to the topology, that is used under the TD-Policy, is actually the number of neighbor nodes  $|S_u|$  of the particular node  $u$  that is agent's  $e$  current location. Agent  $e$  updates during its movement a certain parameter  $Q_e$  that corresponds to the total number of neighbor nodes for all nodes that agent  $e$  has visited so far during its walk in the network.

Initially, for any agent  $e$ ,  $Q_e = 0$ . When agent  $e$  moves to some node  $u$  that does not belong to the advertising network  $V_a$  (but it will eventually become part of it since the agent just moved to it),  $Q_e$  is increased by  $|S_u| - 1$  ( $-1$  has the meaning that the link over which the agent arrived should not be considered). If node  $u \in V_a$ , this implies that at least one agent (maybe the same one) has already visited node  $u$  and the corresponding number of neighbor nodes has already been taken into account for replication purposes and therefore,  $Q_e$  does not change its value. Replication of an agent  $e$  takes place if at some node  $u$ ,  $Q_e$  becomes larger than a threshold value  $q$ . When replication takes place for entity  $e$ , a child entity  $e'$  is created and  $Q_e = Q_{e'} = 0$ .

The particular value of threshold  $q$  is to be explored. Small values of  $q$  result in frequent replications resulting in an advertising network that would not be stretched, as already mentioned. On the other hand, large values of  $q$  would keep the number of replications low but they may occur late in the sense that replication may take place long after the agent is outside the dense area.

## 4 Simulation Results

In this section simulation results are presented evaluating the previously presented algorithms. For the simulation purposes, a simulation program that has been written in C programming language is used for topologies of 500 nodes according to the *Random Geometric Graph* model. Each node is randomly put in the  $[0, 1] \times [0, 1]$  area, and communication radius  $r$  is selected so as to create a connected network ( $r = 0.08$ ); the resulting network has 2323 links among its 500 nodes. For the rest of this section  $L = 2$ . The presented results correspond to average values for 10 runs for each experiment.

In Table 1 simulation results regarding coverage and completion time are presented for flooding and single random walker. There are a number of conclusions that may be drawn from these results. As expected, as the advertising budget  $H$  increases, the coverage and the completion time increase. When it is sufficiently large (e.g.,  $H \geq 2000$ ), both approaches achieve high coverage but the completion times are dramatically shorter under flooding. Consequently, when  $H$  is very large (e.g.,  $H \geq 4000$ ), flooding is to be preferred, as it achieves about as high coverage as under the single random walker with drastically shorter completion time.

Referring again to Table 1 one can notice that when  $H$  is small (e.g.,  $H \leq 1000$ ) the single random walker achieves drastically higher coverage compared to flooding. This clearly demonstrates and is due to the larger stretch of the advertising network created by the single random walk compared to flooding. Consequently, when the advertising budget is small and completion time not as critical, the single random walker is to be preferred. This holds true in general for larger amounts of advertising budgets as well, but the difference in the stretch diminishes (e.g.,  $H = 3000$ ) or it could be reversed (e.g., for  $H = 4000$  flooding achieves a bit higher coverage) due



to the dominating impact of the “noise” in (or probabilistic impact of) the spreading processes when coverage reaches its limit.

**Table 1** Coverage  $C(L)$  (%) and completion Time for  $L = 2$  for the Flooding and Single Random Walker (both running until  $H$  expires).

$H$	$C(L)$ (%)						Completion Time					
	100	500	1000	2000	3000	4000	100	500	1000	2000	3000	4000
Flooding	6.8	32.4	47.8	81.8	98.2	100	3	8	11	14	18	22
SRW	24.9	56.5	85.5	95.4	99.3	99.4	100	500	1000	2000	3000	4000

Another conclusion is that the completion time under a single random walker is fairly high (in fact it coincides with  $H$ ). The main conclusion drawn from the results shown in Table 2 (where results under TI-policy are presented) is that the introduction of replicas in the random walk helps reduce the completion time significantly at a generally very small loss (or occasionally no loss or even gain) in coverage. Thus, when both coverage and completion time are of interest and the advertising budget is neither very small nor very large, the proposed policies that introduce replicas of the random walker are more effective. Notice that for  $H = 500$  and  $p = 0.05$  the achieved coverage is 68.5% and completion time is 238.7, compared to 56.5% and 500, respectively under the single random walk. Thus, for small  $H$  both the completion time and coverage are improved, indicating that the generated replicas not only yield the benefits of faster built up of the advertising network due to the multiplicity of the agents, but they also help build a stretchier advertising network. For  $H = 4000$  the completion time decreases significantly (1030.6 versus 4000 in the single random walker case) at a slight loss in coverage (97.2% versus 99.4%), indicating a slight reduction in the stretchiness of the advertising network that may be attributed to the increased chance that the replicas cover some overlapping areas when the “virgin” area in which to move and cover decreases for large  $H$ . Finally notice from Table 2 that as expected, both coverage and completion time increase as  $H$  increases. Furthermore, for the same  $H$ , the coverage of the network is smaller for large values of  $p$  (e.g.,  $p = 0.9$ ) than the corresponding values when  $p$  is small (e.g.,  $p = 0.05$ ), as expected. This trend is also reflected in the completion time, which decreases as  $p$  increases due to the increased number of replicas.

Table 3 presents the results under the TD-Policy approaches for various values of  $q$  and  $H$ . Both coverage and completion time increase as  $H$  increases. The coverage of the network for the same  $H$  is generally smaller for small values of  $q$  (i.e., where more replicas are created) as is expected. When the values of  $q$  become large enough (in which case replicas are very unlikely to be created), then the performance is similar to the single random walk. The observation made regarding the TI-policy and the results on Table 2 are also valid here, where a small  $p$  in the TI-policy corresponds to large  $q$  in the TD-policy.

Table 4 presents a comparison between the single random walker and the TI-Policy and TD-Policy from another perspective. This time the resources are not

**Table 2** Coverage  $C(L)$  (%) and completion time under the TI-Policy for  $L = 2$  and various values of  $p$  and  $H$ .

$p/H$	$C(L)$ (%)						Completion Time					
	100	500	1000	2000	3000	4000	100	500	1000	2000	3000	4000
0.05	13.3	68.5	79.5	94	98.6	97.2	64.7	238.7	453.6	724.1	1030.6	1362.9
0.1	20.2	56.8	80.5	91.7	98.2	98.5	54.3	240.8	346.9	601.5	862	1069
0.3	19.1	52.3	77.6	91.1	96.9	99.3	35.7	133.8	246.7	427.9	511.9	658
0.5	15.2	45.3	68.5	90.9	94.5	96.4	29.3	82.1	148.4	264.6	380.9	456.7
0.7	14.7	38.6	58.4	82	90.7	98.5	21	61.2	91.2	156.6	214.4	280
0.9	11.6	24.1	38.3	57.4	73.7	80.4	13.5	30.1	47	76.1	103.4	128.4

**Table 3** Coverage  $C(L)$  (%) and completion time under the TD-Policy for  $L = 2$  and various values of  $q$  and  $H$ .

$q/H$	$C(L)$ (%)						Completion Time					
	100	500	1000	2000	3000	4000	100	500	1000	2000	3000	4000
10	12.3	30.9	49.8	75.6	87.3	94.6	22.2	74.3	121.9	287.9	343.4	366.4
20	9.8	38.1	61.8	83.2	93.4	98.4	29.6	103.4	154.4	286.4	421.9	516.1
30	14.3	44.6	70.8	86.5	95.2	98.9	32.5	120.5	214.6	299.3	496.5	651
50	16.4	45.6	66.5	92.2	97.5	98.1	48.5	157.3	249.3	344.1	577.8	755.5
150	25.2	55.3	80.6	95.5	97.1	99.1	65.5	214.5	337	483.9	659	797
250	18.3	65.3	77.7	91	98.3	99.2	91.8	241.3	350.3	555.1	782.8	931.9
400	17.4	67.1	86.2	95.8	98.3	99.3	99.8	291.6	409	667.1	971.6	1186.5
600	19.6	64.5	77.7	94.9	97.2	99.1	100	374.7	548.3	791.2	1107	1421.3
2000	28.5	68.2	82.1	95.8	96.9	99.9	100	491.6	885.1	1315.4	1852.6	2371.5

kept constant, but the time and the coverage are kept constant instead. In more details, the first column (labelled: SRW  $ct$ ) corresponds to the completion time of the single random walker. The coverage under the TI-Policy and the TD-Policy for the same completion time is shown in the next two columns. The fourth column (labelled: SRW  $C(L)\%$ ) corresponds to the coverage under the single random walker for the completion time of the same row and the first column. The last two columns correspond to the completion time under the TI-Policy and the TD-Policy, in order to achieve the same coverage as the one corresponding to the single random walker case.

Table 4 clearly shows that for same completion time both proposed policies outperform the single random walker with respect to coverage. For example, for completion time 500 (SRW  $ct = 500$ ) the coverage is 82.6%, 94.6% and only 56.5% under the TI-policy, TD-policy and the single random walker, respectively. Similarly, for the same coverage both proposed policies outperform the single random walker with respect to completion time. For example for coverage 99.3% (SRW  $C(L)\% = 99.3$ ) the completion time is 709.8, 541.6 and as high as 3000 under the TI-policy, TD-policy and the single random walker. The increased coverage for the same completion time demonstrates again the larger stretchiness of the advertising

network that can be achieved within a given available completion time. For applications for which service advertisement needs to be done in a (strict) timely manner (i.e., completion time is bounded), the proposed policies can achieve significantly higher coverage. It should be noted that the advertising budget actually utilized is not the same under the three policies, though. Nevertheless, it is for lower than flooding.

**Table 4** Coverage  $C(L)$  (%) for various values of the completion time (column: SRW  $ct$ ) and coverage (column: SRW  $C(L)$ %) under Single Random Walker, TD-Policy and TI-Policy ( $p = 0.1$  and  $q = 250$ ).

SRW $ct$	$C(L)$ (%)		SRW $C(L)$ %	Completion Time	
	TI-Policy	TD-Policy		TI-Policy	TD-Policy
<b>100</b>	29.82	36.12	<b>24.9</b>	70.6	60.4
<b>500</b>	82.6	94.6	<b>56.5</b>	172.4	202.6
<b>1000</b>	99.16	99.96	<b>85.5</b>	374.4	367
<b>2000</b>	100	100	<b>95.4</b>	506.8	379.6
<b>3000</b>	100	100	<b>99.3</b>	709.8	541.6
<b>4000</b>	100	100	<b>99.4</b>	972.6	572.2

Eventually, the simulation results presented in this section have shown in the first place that a single random walker is a suitable alternative to flooding, when the number of allowed messages is limited. This is due to the capability of the single random walker to create stretched advertising networks. For both proposed replication policies it was shown that an improvement in the completion time is achieved generally at the expense of a slight decrease in coverage, when the number of messages is kept constant; sometimes, though, coverage may also be improved in addition to the completion time. Finally it has been shown that the proposed policies improve the coverage of the single random walker when completion time is fixed, or they can achieve a given coverage in shorter completion time.

## 5 Summary and Conclusions

In this paper the problem of service advertising is studied for unstructured network environments. An advertising process disseminates information to the network nodes in an attempt to make information about a certain service location a priori available in as wide (or stretchable) network area as possible, to ease searching for it. A generic, simple, service searching approach is subsumed here: each node is considered as capable of employing a simple searching mechanism that searches for the particular information at most  $L$  hops away from the particular node.

Flooding and single random walk were shown to be at rather extreme points with respect to coverage, completion time and number of required messages. However, as shown in this paper, a single random walker is capable of creating a more “stretched”

advertising network than flooding, especially when the advertising budget is small, resulting in improved completion times and sometimes even coverage.

This capability of the random walk approach was exploited further in the sequel in the context of two proposed policies, by allowing agents to move in the network according to the random walk paradigm and replicate themselves when certain criteria are met (with some probability  $p$  that decreases exponentially or by using information related to the topology).

Simulation results, for both proposed replication policies when compared to single random walker, have shown that an improvement in the completion time is achieved generally at the expense of a slight decrease –or even increase– in coverage, when the number of messages is kept constant. Finally it has been shown that the proposed policies improve the coverage of the single random walker when completion time is fixed, or they can achieve a given coverage in shorter completion time.

## References

1. Gnutella. Gnutella rfc. <http://rfc-gnutella.sourceforge.net/>, 2002.
2. A.Segall, "Distributed network protocols", *IEEE Transactions on Information Theory*, vol. IT-29, pp.23-35, 1983.
3. B.Williams, T.Camp, "Comparison of broadcasting techniques for mobile ad hoc networks", *MOBIHOC 2002*, pp.194-205, 2002.
4. F.Banaei-Kashani and C.Shahabi, "Criticality-based analysis and design of unstructured peer-to-peer network as "complex" systems", in *Proceedings of the Third International Symposium on Cluster Computing and the Grid*, pages 351-358, 2003.
5. V. Dimakopoulos and E.Pitoura, "On the performance of flooding-based resource discovery", *IEEE Transactions on Parallel and Distributed Systems*, 17(11):290-297, November 2006.
6. K.Oikonomou and I.Stavrakakis, "Performance analysis of probabilistic flooding using random graphs", in *The First International IEEE WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC 2007)*, Helsinki, Finland, 18 June, 2007.
7. Y.Sasson, D.Cavin, and A.Schipper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks", in *Swiss Federal Institute of Technology (EPFL), Technical Report IC/2002/54*, 2002.
8. D.Tsoumakos and N.Roussopoulos, "Adaptive probabilistic search for peer-to-peer networks", in *3rd IEEE International Conference on P2P Computing*, 2003.
9. M.Bani Yassein, M.Ould-Khaoua and S.Papanastasiou, "On the Performance of Probabilistic Flooding in Mobile Ad Hoc Networks", in *11th International Conference on Parallel and Distributed Systems (ICPAD '05)*, 2005.
10. N.Alon, C.Avin, M.Koucky, G.Kozma, Z. Lotker, M.Tuttle, "Many Random Walks Are Faster Than One", *ArXiv e-prints. Vol. 705*, May 2007.
11. C.Gkantsidis, M.Mihail and A.Saberi, "Random Walks in Peer-to-Peer Networks", in *Proceedings of IEEE INFOCOM*, 2004.
12. C.Gkantsidis, M.Mihail and A.Saberi, "Hybrid Search Schemes for Unstructured Peer-to-Peer networks", in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 3, pp. 1526-1537, Miami, Fla, USA, March 2005.
13. T. Haynes, S. Hedetniemi and P.Slater, "Fundamentals of Domination in Graphs", *Marcel Dekker, New York*, 1998.
14. P.B. Mirchandani and R.L. Francis, "Discrete Location Theory", John Wiley and Sons, 1990.