Coverage-maximization in networks under resource constraints

Subrata Nandi,[1] Lutz Brusch,[2] Andreas Deutsch,[2] and Niloy Ganguly[1]
[1]*Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur 721302, India*
[2]*Center for Information Services and High Performance Computing, Technical University Dresden, 01062 Dresden, Germany*
(Received 18 October 2009; revised manuscript received 19 April 2010; published 16 June 2010)

Efficient coverage algorithms are essential for information search or dispersal in all kinds of networks. We define an extended coverage problem which accounts for constrained resources of consumed bandwidth $\mathcal{B}$ and time $\mathcal{T}$. Our solution to the network challenge is here studied for regular grids only. Using methods from statistical mechanics, we develop a coverage algorithm with proliferating message packets and temporally modulated proliferation rate. The algorithm performs as efficiently as a single random walker but $\mathcal{O}(\mathcal{B}^{(d-2)/d})$ times faster, resulting in significant service speed-up on a regular grid of dimension d. The algorithm is numerically compared to a class of generalized proliferating random walk strategies and on regular grids shown to perform best in terms of the product metric of speed and efficiency.

DOI: 10.1103/PhysRevE.81.061124 PACS number(s): 05.40.Fb, 64.60.aq, 89.70.−a, 89.75.Hc

I. INTRODUCTION

Information management in distributed systems ranging from social networks and the Internet to wireless *ad hoc*, peer-to-peer (P2P), sensor, and mesh networks requires search [1–5], dissemination [6], and gathering [7] of information. The corresponding algorithms maximize the expected number of distinct visited nodes, the *coverage* $C(T)$ of the network starting from a single node, within a service latency of T time steps [1–3,5,7–9].

Real network systems like the P2P networks, KaZaA [10] and Gnutella [11], typically use time-to-live (TTL)-constrained flooding [31] or random walk as coverage algorithm. For example, if TTL is set to 2, then a message packet can hop at most twice, after which it is discarded. Thus, by fixing the maximum TTL, real systems also place an implicit limit to the overall amount of bandwidth per service request. A systematic understanding of the implications of resource constraints on service performance as well as of the optimal utilization of the allocated resources is highly desired but currently lacking.

All real-world networks work under inherent resource constraints. Limiting resources range from battery power in wireless sensor or *ad hoc* networks to communication bandwidth in P2P networks. Most of the mentioned networks are unstructured, i.e., they lack a central index. If the nodes and message packets do not possess any memory of previously visited nodes then the coverage strategy has to rely on decentral forwarding of any message packet from its current node to a *randomly* selected neighbor node during one time step [1,4]. We assume that all inter-node connections possess the same bandwidth. Each service request is allotted once a bandwidth quota $\mathcal{B}$ which gets consumed by one unit each time a packet is forwarded.

Problem statement. We define the following fundamental and novel problem: Starting from a single node and without memory of visited nodes, maximize node coverage in a given time $\mathcal{T}$ under resource constraint $\mathcal{B}$.

Statistical mechanics perspective. To solve this problem, the message packets are modeled as random walkers and in the following studied from a statistical mechanics perspective. Two variants of random walk (RW) algorithms have already been applied for distributing message packets on networks: K-RW where the walker number $K \geq 1$ remains fixed [12] and proliferating random walk [$p(t)$-RW] where the walker number increases with time [13]. In the latter case, a walker self-replicates at its current node with rate $p(t) \in \mathcal{R}^+$ at time t such that on average each walker produces one offspring walker every $1/p(t)$ time steps. Previously developed algorithms have either set application-specific rules for walker proliferation or used a constant rate $p(t)=P_C$, where P_C satisfies the condition $\mathcal{B}=\sum_{t=1}^{\mathcal{T}} K(t)=\sum_{t=1}^{\mathcal{T}}(1+P_C)^{t-1}$ with service latency $T=\mathcal{T}$. Neither the K-RW nor the constant proliferation strategy, however, maximize coverage under constraints as we show below.

In $K(>1)$-RW, multiple random walkers are used to reach more distinct nodes than a single random walker can visit during the same time. However, $K>1$ random walkers visit nodes redundantly, that is, different walkers visit the same nodes, causing *mutual overlap* especially during the starting phase. Unlike multiple walkers, a single random walker only has its inherent overlap. Hence, given $\mathcal{B}$ there is an upper bound C_{max} to the amount of achievable coverage which equals C_1, i.e., the coverage obtained by 1-RW. By using a larger K, the allocated bandwidth $\mathcal{B}$ can be consumed faster at the cost of reduced coverage due to increased mutual overlap. A further problem is that after some time the K random walkers move "far" apart leaving unexplored area in between. Hence, for a given bandwidth $\mathcal{B}$ there is an inherent tradeoff between the coverage and the required time. To circumvent the mutual overlap of a $K(\gg 1)$-RW, a P_C-RW with small initial walker number and moderate P_C could be used. This however is a naive strategy and guarantees neither elimination of overlap in the early stage nor reduction of the uncovered area at the later stage as demonstrated by numerical simulation and quantitative comparison of different algorithms below.

To solve the above-posed fundamental coverage problem one now has to answer the question: Is C_{max} achievable only at $T=\mathcal{B}$ using a single walker or can it be achieved at $T=T_{min} \ll \mathcal{B}$ by multiple walkers with a suitably chosen proliferation strategy? We here develop such a proliferation strat-

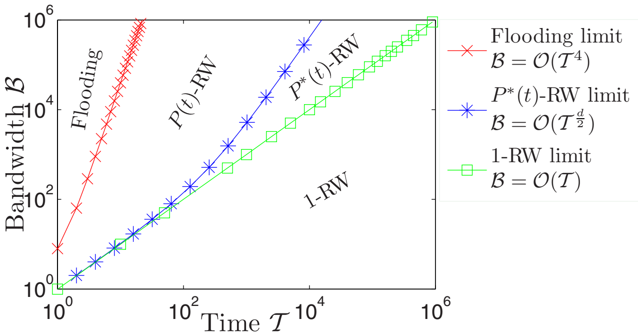

FIG. 1. (Color online) Phase diagram of resources $(\mathcal{T},\mathcal{B})$ allocated to a service on a network and the specific algorithm, as denoted, which yields maximum coverage $C(\mathcal{T})$ at given $(\mathcal{T},\mathcal{B})$. Flooding and 1-RW are optimal in the limit cases of large $\mathcal{B}$ and $\mathcal{T}$, respectively. The novel proliferation strategy $P^*(t)$-RW is preferred for constraints $\mathcal{O}(\mathcal{B}^{2/d}) \leq \mathcal{T} \leq \mathcal{B}$, i.e., between the phase boundaries marked with square (green) and asterisk (blue). $P(t)$-RW is preferred for $\mathcal{O}(\mathcal{B}^{1/d}) < \mathcal{T} < \mathcal{O}(\mathcal{B}^{2/d})$, i.e., between the asterisk (blue) and cross (red) phase boundaries. Phase boundaries are shown for a Euclidean grid with $d=4$, as an example.

egy for infinite d-dimensional Euclidean grids (with $d>2$) by exploiting the statistical properties of the K-random walk, that achieve C_{max} in $\mathcal{O}(\mathcal{B}^{(d-2)/d})$ times faster than a single random walker, implying significant speed-up. The new algorithm solves the twin problems of mutual overlap and unexplored area. Further, any real-world service operating with bandwidth $\mathcal{B}$ may need a higher speed-up than $\mathcal{O}(\mathcal{B}^{(d-2)/d})$ but a time constraint $\mathcal{T} < T_{min}$ will result in coverage $C(\mathcal{T}) < C_{max}$. In such a scenario, we show that $P^*(t)$-RW can be generalized to a class of strategies, termed $P(t)$-RW, which yield higher coverage than the naive strategies $\lceil\frac{\mathcal{B}}{\mathcal{T}}\rceil$-RW and P_C-RW.

The application ranges of these strategies are compared in the phase diagram $(\mathcal{T},\mathcal{B})$, Fig. 1, by denoting which strategy yields maximum node coverage $C(\mathcal{T})$ at given $(\mathcal{T},\mathcal{B})$. We find a stereotypic sequence of the best-performing algorithms as $\frac{\mathcal{B}}{\mathcal{T}}$ increases. For the extreme cases of large $\mathcal{B}$ and $\mathcal{T}$, the best-performing algorithms are already known as flooding and 1-RW, respectively. The proliferation strategy $P^*(t)$-RW will be shown to perform best for $\mathcal{O}(\mathcal{B}^{2/d}) \leq \mathcal{T} \leq \mathcal{B}$ while the generalized strategies $P(t)$-RW are best in the intermediate range.

II. *K*-RW DYNAMICS

In a pioneering work Larralde *et al.* [9,18] studied the dynamics of multiple $K \gg 1$ random walkers on an infinite d-dimensional Euclidean grid and derived an asymptotic expression for $C(\mathcal{T})$, which was supported by a more rigorous solution by Yuste *et al.* [19].

Table I summarizes the *increase* in coverage $\Delta C(t)$ during one time step at time t $(1 \leq t \leq T \leq \mathcal{T})$ of a K-RW on an infinite d-dimensional Euclidean grid [9,18,19]. The coverage does not increase uniformly but with qualitatively different behavior within each of the three regimes that are subsequently encountered as t increases beyond the *crossover* *times* t_1^c and t_2^c, respectively. In regime I, all the walkers clutter together and the mutual overlap probability is very high. However, unvisited neighbors of all the visited nodes are reached during the next step, resulting in a floodinglike coverage. For comparison, the pure flooding algorithm replicates packets at each occupied node as often as needed to forward one packet to each neighbor node. The considered K-RW enters regime II when t passes t_1^c as the walkers gradually move away from each other and less walkers co-occupy nodes. However, still some amount of mutual overlap persists. In regime III, the walkers are sufficiently separated such that mutual overlap almost vanishes. As a result, from time $t=t_2^c(K)$ onward, i.e., *crossover* from regime II to III, each walker behaves independently like a single random walker with nonoverlapping exploration space. We observe that $t=t_2^c(K)$ is an *optimal time* for efficient coverage since it yields a high coverage increment per time step due to low mutual overlap and low wastage of inter-walker space. The crossover time $t_2^c(K)$ is reached early if K is small.

TABLE I. Coverage increments $\Delta C(t)$ for $K \gg 1$ random walkers on a $d(>2)$-dimensional Euclidean grid [9,18,19]. ξ' and ξ are parameters. For $d=1,2$, the K random walkers show the same behavior in regimes I and II with identical t_1^c but regime III does not occur for $d=1$ and occurs at $t_2^c \sim e^K$ for $d=2$.

Regimes :	I	II	III
$\Delta C(t)$:	$d \times t^{d-1}$	$\frac{d}{2}[t \times \ln(K \times t^{1-\frac{d}{2}})]^{\frac{d}{2}-1}$	K
Time (t):	$---$	$-----------$	$--->$
Crossover time:	$t_1^c(K) = \xi' \times \ln K$ ($\Downarrow$, I→II)	$t_2^c(K) = \xi \times K^{\frac{2}{d-2}}$ ($\Downarrow$, II→III)	

Let us denote by $E(t)$ the efficiency during time step t, which is defined as the ratio of the coverage increments $\Delta C(t)$ to the number of walkers used during time step t. In regimes I and II efficiency $E(t)$ increases with time but reaches and keeps the peak value $E(t \geq t_2^c) = E_{max}$ as it enters regime III. E_{max} is the efficiency of a single random walker.

III. COVERAGE STRATEGY UNDER CONSTRAINT $\mathcal{B}$

The system with a constant number K of walkers will cross over from regime II to III at a time

$$t_2^c(K) = \xi \times K^{2/d-2}. \tag{1}$$

If no further walkers are introduced, the system will stay in regime III where overlaps do not decrease further but large parts of the network remain unexplored. However, starting with a small number of random walkers at $t=1$ and proliferating each walker at a suitable rate $P^*(t)$ at each time step, the system can always remain at the regime boundary as desired.

A. Calculation of $P^*(t)$

Let the per walker proliferation rate $P^*(t)$ produce $K(t)$ walkers at time t. Then $K(t)$ can be obtained from the following recurrence:

$$K(t+1) = K(t) \times [1 + P^*(t)]. \tag{2}$$

We consider a slowly changing walker number $K(t)$ and extend the dependency of $t_2^c(K)$ on K from Eq. (1) to $K(t)$ by adiabatic approximation and insert the requirement that $K(t)$ maintains the system at the regime boundary.

$$t_2^c(K) = t = \xi \times K(t)^{2/(d-2)} \Rightarrow K(t) = \left(\frac{t}{\xi}\right)^{(d-2)/2} \tag{3}$$

Hence, the initial walker count is $K(1)=\xi^{-(d-2)/2}$ which provides the *initial condition* for the recurrence in Eq. (2). Substituting the values of $K(t+1)$ and $K(t)$ as obtained from Eq. (3) into Eq. (2), $P^*(t)$ is calculated as

$$P^*(t) = \left(1 + \frac{1}{t}\right)^{(d-2)/2} - 1 \tag{4}$$

Expanding $P^*(t)$ in powers of $\frac{1}{t}$ and ignoring the higher order terms in $\frac{1}{t}$ in the asymptotic range $t \gg 1$, the proliferation rate takes the form

$$P^*(t) \approx \frac{d-2}{2} \times \frac{1}{t} \tag{5}$$

showing a fast decay with time. Hence, $K(t+1)-K(t) \ll 1$ and the adiabatic approximation in Eq. (3) is consistent.

Since $P^*(t)$-RW consumes the bandwidth $\mathcal{B}$ in T_{min} time steps, we can write $\mathcal{B}=\sum_{t=1}^{T_{min}} K(t)$ and with Eq. (3)

$$\mathcal{B} = K(1) \times \sum_{t=1}^{T_{min}} t^{(d-2)/2} \approx 2 \times \xi^{-(d-2)/2}\left(\frac{T_{min}^{d/2} - 1}{d}\right). \tag{6}$$

Therefore, $P^*(t)$-RW utilizes the given bandwidth $\mathcal{B}$ in a best possible way, providing the lower bound on the time to achieve C_{max} as $T_{min}=\mathcal{O}(B^{2/d})$. Given $\mathcal{B}$, we define the service *speed-up* S as the ratio of the time taken to obtain C_{max} by 1-RW to the time taken by $P^*(t)$-RW.

$$S = \frac{B}{T_{min}} = \mathcal{O}(B^{(d-2)/d}) = \mathcal{O}(T_{min}^{(d/2-1)}) \tag{7}$$

In the following, this approximative result for the service speed-up and the performance of the proliferation rate $P^*(t)$-RW are verified empirically.

B. Empirical verification

For calculating the values of S and $P^*(t)$, an empirical estimation of ξ is needed. We present an estimation of ξ followed by the verification of the performance of $P^*(t)$ and its speed-up, through simulation.

Estimating ξ empirically. Simulations are performed on a Euclidean grid with periodic boundary conditions and for $d=4$ with $|\mathcal{L}|=130^4$ nodes. For the random walk, the Mersenne random number generator has been used [20]. Simulation data were averaged over 10 000 realizations. K-RW simulation for different discrete values of K is performed to obtain the efficiency $E(t)$. Figure 2 shows the lin-log scale plot of the efficiency $E(t)$ during the time step t versus t for $K=1$ and 10. Similar behavior has been observed for any $K>1$ across different dimensions $d=3$, 4, 5, and 6

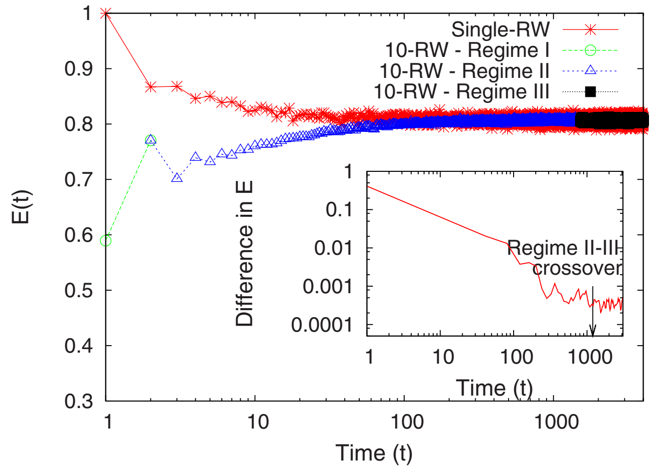

FIG. 2. (Color online) Efficiency $E(t)$ versus time t (on lin-log scale) for K-RW on a four-dimensional Euclidean grid of $|\mathcal{L}|=130^4$ nodes. $K=1$ is shown in red '$*$' throughout. Three regimes for $K=10$ are shown, regime I in green "○," regime II in blue "△" and regime III in black "□" which occur overlayed on red "$*$"s. For 10-RW the crossover time from regime II to III ($t_{10}^c=1525$) is estimated from the intersection of the asymptotics of regime II (decreasing difference) and regime III (fluctuating difference around a constant nonzero average), as shown in the inset by a log-log plot of E_1-E_{10}.

which confirms the results in Table I. The values of t_K^c are recorded for different walker number, $K=1$ to 30 at an interval of 3, and then a least square fit of Eq. (1) yields ξ.

Theoretically, as $\xi>1$ for grid dimensions $d=3\ldots6$, a continuous initial walker number should be $K(1)<1$ but the coverage process has to start with at least one random walker. To nullify this discretization effect in initial walker number, we modify $P^*(t)$ to $P^*(t+t')$ where t' is the earliest time with $K(t')\geq 1$ if simulated for continuous K with $P^*(t)$ and $K(1)=\xi^{-(d-2)/2}$. Substituting $K(t=t')=1$ in Eq. (3) yields $t'=\xi$.

Performance evaluation of $P^*(t)$. To better understand the behavior of $P^*(t)$, a generalized proliferation rate $P(t)$ is defined and analyzed by extending Eq. (4) and considering at least one initial walker.

$$P(t) = \left(1 + \frac{1}{t+\xi-1}\right)^{(\alpha\times(d-2)/2)} - 1. \tag{8}$$

Here, we introduce a new parameter α as a degree of freedom for exploring regimes II and III. This choice of extension lets the process operate, for $\alpha>1$, within regime II with constant efficiency and correspondingly constant distance from the regime boundary as verified later. Note that $P(t)|_{\alpha=1,\xi=1}=P^*(t)$.

It is expected that the coverage of $P(t)$-RW is similar to that of 1-RW for $\alpha\leq 1$. Figure 3 shows the relative difference in coverage between $P(t)$-RW and 1-RW and reveals that it is as low as 0.3% for $\alpha<1$ whereas it increases significantly for $\alpha>1$, independent of the network dimension. The small deviation for $\alpha\leq 1$ stems from the smooth transition between regimes II ($\alpha>1$) and III ($\alpha<1$). Next, we study the behavior of $P(t)$-RW [Eq. (8)] as a function of α

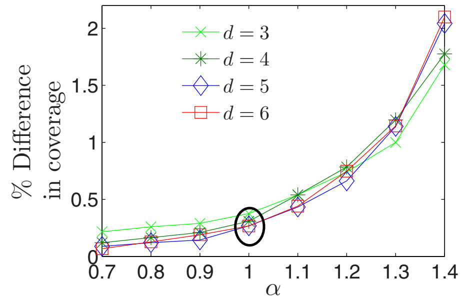

FIG. 3. (Color online) Performance of the generalized coverage algorithm $P(t)$ for different α and allocated bandwidth $\mathcal{B}=5\times10^4$, as a typical case. The performance of $P^*(t)$ at $\alpha=1$ is highlighted by a black "oval." The plot shows the relative difference in coverage $\frac{C_{max}-C(T)}{C_{max}}\times 100\%$ obtained using $P(t)$-RW and compared to 1-RW for $d=3$, 4, 5, and 6.

taking $d=4$ as a case study. The results are also valid for other dimensions.

For arbitrary constraints $(\mathcal{T},\mathcal{B})$, the performance of $P^*(t)$ can be assessed in the light of the coupled optimization problems of latency T and coverage C which are presented in comparison to 1-RW in Fig. 4. We observe that in order to increase the efficiency slightly further than that achieved by $P^*(t)$ by decreasing α below 1, one would have to spend a much longer time T [Fig. 4(b)]. On the other hand, efficiency decreases fast if one tries to consume the given $\mathcal{B}$ slightly faster by increasing α beyond 1 [Fig. 4(a)]. The same behavior has been found for all investigated network dimensions. To quantitatively assess this coupled optimization problem in a multidimensional solution space, a metric is required that combines time and efficiency. Though there may be many possible ways to combine, we define the combined metric M as

$$M(\alpha) = T(\alpha)[E_{max} - E^{avg}(\alpha)]. \tag{9}$$

M combines the time needed to consume the allocated bandwidth (first factor) with a measure for the wastage of bandwidth $[E_{max}-E^{avg}(\alpha)]$ as compared to the maximally achievable efficiency E_{max} of the 1-RW. The desired algorithm shall operate fast and waste little, hence it *minimize* M. Figure 4(c) shows that $M(\alpha)$ is moderate for small α (regime III) where a few walkers take a lot of time and is also large for large α (regime II) where many walkers waste bandwidth due to mutual overlap. In particular, $M(\alpha)$ increases rapidly for $\alpha>1$ as the efficiency decreases rapidly beyond $\alpha=1$. The *minimum* of $M(\alpha)$ is found at $\alpha=1$. These numerical results confirm, within the class of generalized algorithms given by Eq. (8), that P^*(t) with $\alpha=1$ provides the best performance as measured by the product metric of Eq. (9).

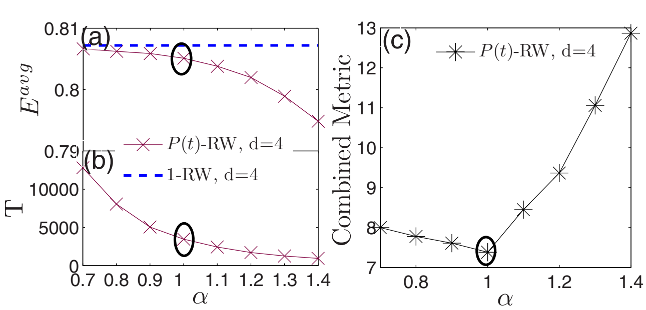

FIG. 4. (Color online) Performance of the generalized coverage algorithm $P(t)$ for different α, for $d=4$ and allocated bandwidth $\mathcal{B}=5\times10^4$, as a typical case. The performance of $P^*(t)$ at $\alpha=1$ is highlighted in each panel by a black "oval." (a) and (b) show mean efficiency $E^{avg}=\Sigma_{t=1}^{T}\frac{E(t)}{T}$ and the time T taken by a $P(t)$-RW to consume the given bandwidth $\mathcal{B}$, respectively. The dashed line corresponds to E_{max} and equals the mean efficiency of a 1-RW. (c) Combined metric $M(\alpha)=T(\alpha)(E_{max}-E^{avg})$ as function of α.

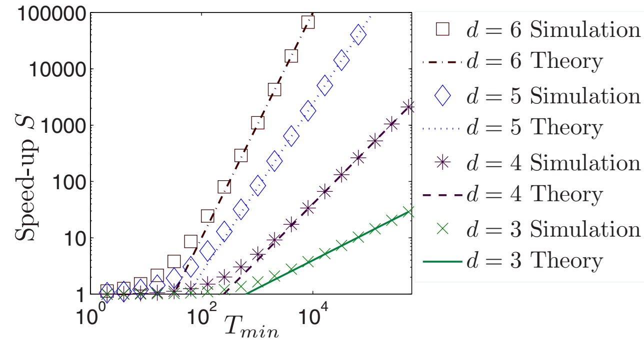

FIG. 5. (Color online) Speed-up $S=\frac{T_{1\text{-RW}}}{T_{P^*(t)\text{-RW}}}$ versus the minimal time $T_{min}=T_{P^*(t)\text{-RW}}$ to consume the given bandwidth $\mathcal{B}$ for different dimensions d of an Euclidean grid. The analytical result $S=\mathcal{O}(T_{min}^{(d/2-1)})$ is shown by solid lines [Eq. (7)] and symbols denote simulation results.

Verifying speed-up S. Figure 5 compares the analytical result for the service speed-up [Eq. (7)] with the simulation results. The speed-up is significant and the theoretical and simulation values match well except for small T_{min} where the approximation of Eq. (4) by Eq. (5) fails.

IV. COVERAGE STRATEGY UNDER CONSTRAINTS $\mathcal{B}$ AND $\mathcal{T}<T_{min}$

In many real-world network applications the service latency needs to be much smaller than $T_{min}=\mathcal{O}(\mathcal{B}^{2/d})$ whereas wastage of some bandwidth may be permissible. Under the given constraints, $P(t)$ can be expressed using Eq. (8) with $\alpha>1$. The parameter value α can be calculated numerically such that the condition $\mathcal{B}=\Sigma_{t=1}^{\mathcal{T}}\{K(t)[1+P(t)]\}$ holds true. Results show that when $\mathcal{T}<T_{min}$ and hence $\alpha>1$ is chosen, the efficiency of the $P(t)$-RW remains steady, although its value is slightly smaller than the steady efficiency of $P^*(t)$-RW implying that the process works at a fixed operation point in regime II. However, when $\alpha\gg1$, i.e., $\mathcal{T}\ll T_{min}$ the efficiency falls sharply with time implying that the strategy fails to maintain an operation point of constant efficiency, rather with time it is gradually shifted into regime I.

The performances of K-RW, P_C-RW and $P(t)$-RW are compared in Fig. 6 with $K=\lceil\frac{\mathcal{B}}{\mathcal{T}}\rceil$, P_C and α chosen such that $\mathcal{B}$ was consumed within $\mathcal{T}$. The performance of $P(t)$-RW proved superior, especially when $\mathcal{T}\ll T_{min}$. Simulation results for other dimensions exhibit similar behavior. The performance of $P(t)$-RW improves for higher bandwidth and lower dimensionality. Figure 1 summarizes the application ranges

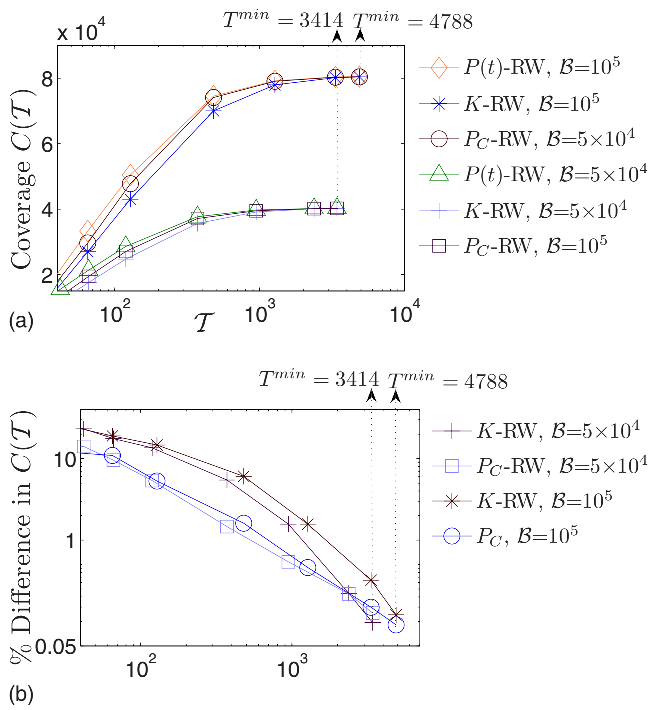

FIG. 6. (Color online) Performance comparison for coverage under two constraints $(\mathcal{T},\mathcal{B})$. (a) shows coverage $C(\mathcal{T})$ versus $\mathcal{T}$ for $P(t)$-RW, K-RW and P_C-RW, with d=4 for $\mathcal{B}$=10^5 and 5×10^4. $\mathcal{T}$ is varied from T_{min}=4788 and 3414, respectively, to very small values. (b) Relative difference in $C(\mathcal{T})$ obtained by P_C-RW and K-RW compared to $P(t)$-RW versus $\mathcal{T}$. For $\mathcal{T}\ll T_{min}$, $P(t)$-RW yields 10% and 20% more coverage than P_C-RW and K-RW, respectively. However, for $\mathcal{T}\simeq T_{min}$, the difference among the strategies is insignificant. K=$\lceil\frac{\mathcal{B}}{\mathcal{T}}\rceil$, P_C and α are chosen such that $\mathcal{B}$ is consumed within $\mathcal{T}$.

of the different coverage algorithms. Interestingly, no single algorithm is optimal for one fixed constraint and variable second constraint.

V. RELATED WORK

The underlying problem of coverage by a single random walker has been widely studied since the estimation of coverage $C(T)$ on a d-dimensional Euclidean grid in 1951 [8]. Since then, research efforts on randomized network coverage have concentrated on studying properties of $C(T)$ in infinite graphs [9,12,18,19] using both single and multiple random walkers. The subject has also become of growing interest to network scientists, however the main thrust has been to identify optimal topologies for speed-up, rather than algorithm design of strategies to achieve the speed-up for a given topology. The characterization of full [14–16] and partial [2] cover times [32] of finite graphs of size n has been investigated considering a single random walker. In a typical application scenario (like search) an object is replicated in multiple nodes, hence the partial cover time [PCT(c)] [33] is important. Avin *et al.* [2] proved that for 1-RW the upper bound on the PCT(c)=$\mathcal{O}(h_{max})$, where h_{max} is the maximum hitting time [34]. PCT is asymptotically smaller than the Matthews bound [16] on the cover time, intuitively meaning, on sufficiently large graphs, almost all the time used by a walker to cover the entire graph is spent trying to reach the last remaining $\log(n)$ nodes. For a finite size grid (d=4), PCT(c)=$\mathcal{O}[n\log(n)]$ which is larger than $\mathcal{O}(n)$ as obtained for an infinite grid due to the finite size effect, whereas the proposed proliferation algorithm takes $\mathcal{O}(\sqrt{n})$ time. Bisnik *et al.* [3] show that to obtain the best performance, the random walk parameters, the walker count K and TTL T must be a function of the estimated object popularity.

More recently, coverage maximization by using multiple (K) walkers [14,17] is of key interest for the computer science community. For example [17], determined the walker number for a K-RW strategy, that can speed-up the coverage reducing mutual overlap for a large class of special graphs, where all walkers start from a single node [17]. shows for d-dimensional grids, hypercubes and E-R random graphs that there exists a lower bound in speed-up which is linear in K when $K<\mathcal{O}(\log^{1-\epsilon}n)$, where n is the size of the graph and ϵ is an arbitrary constant $0<\epsilon<1$.

These results imply for a d-dimensional grid, that a suitably chosen K-RW can cover the grid with little mutual overlap within the time $\frac{T_1}{\log^{1-\epsilon}n}$, where T_1 is the time taken by a single walker. Compared to that, the proliferating random walk algorithm $P^*(t)$-RW proposed in this paper yields coverage with minimum overlap in much shorter time $\frac{T_1}{n^{(d-2)/d}}$. Hence, one can achieve significant speed-up even without sacrificing efficiency. Further, we have extended the algorithm to a class of proliferating random walk algorithms which can be used to efficiently cover the entire $(\mathcal{B},\mathcal{T})$ spectrum (Fig. 1). The derived scaling behavior of the phase boundaries can be used to estimate the effect of resource $(\mathcal{B},\mathcal{T})$ preallocation in terms of obtained coverage. Alternatively, the minimum latency $\mathcal{T}$ can be estimated if a certain desired level of coverage is required with a preallocated $\mathcal{B}$.

VI. DISCUSSION AND CONCLUSIONS

We have defined an extended coverage problem which takes into account the resource constraints in the form of consumed bandwidth $\mathcal{B}$ and latency time $\mathcal{T}$. Our work does not consider constrained local queue length and the corresponding problem of congestion which has been studied previously [21–23]. Here, low walker density emerges as a result of the optimality request which lets the algorithm operate at the border of regime III. Walkers behave as isolated entities within regime III and we expect no issues of congestion with our algorithm $P^*(t)$-RW.

The approach toward the design of the algorithm presented here for a Euclidean grid topology can immediately be adopted to search unstructured networks with almost homogeneous node degrees, e.g., sensor networks which are typically modeled as grids [1,2,7]. However, the precise details of Eq. (8) might require modifications if the same strategy shall be applied to other complex networks such as small world [27], power-law [29], and the Internet. Perhaps some of these applications will require replacement of the unbiased random walk by a biased random walk [30], thereby allowing the walkers to choose their next step with nonuniform probability among nearest neighbors. Initial simulations show that our strategy of maintaining the random walk process at the regime boundary is also successful for homogeneous small world topologies [28].

In addition to our analytical approach toward today's preallocation problems, our proposed coverage algorithm can also be used for upcoming sophisticated applications like service differentiation, where each node will get a different quality of service based on subscription level or its history of cooperation [24–26].

We have designed, using methods from statistical mechanics, the coverage algorithm $P^{*}(t)$-RW such that it exploits the advantages of both random walk regimes II and III by keeping the process at the regime boundary. This algorithm yields similar efficient coverage as a single random walker but $S=\mathcal{O}(\mathcal{B}^{(d-2)/d})$ times faster, resulting in significant service speed-up. In a class of all functions given by Eq. (8), numerically it has been shown that $P^{*}(t)$-RW provides the best performance as measured by metric M. However, it will be interesting to mathematically investigate whether this is the optimal strategy in a class of all possible $P(t)$-RW.

ACKNOWLEDGMENTS

We are grateful for stimulating discussions with H. Hatzikirou, B. Mitra, and F. Peruani.

[1] J. Ahn, S. Kapadia, S. Pattem, A. Sridharan, M. Zuniga, J. Jun, C. Avin, and B. Krishnamachari, Comput. Commun. Rev. **38**, 17 (2008).
[2] C. Avin and C. Brito, Proceeding of 3rd ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'04) (unpublished).
[3] N. Bisnik and A. A. Abouzeid, Comput. Netw. **51**, 1499 (2007).
[4] S. S. Dhillon and P. V. Mieghem, *Proceedings of 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)* (unpublished).
[5] B. Wu and A. D. Kshemkalyani, Comput. Commun. **31**, 4138 (2008).
[6] A. O. Stauffer and V. C. Barbosa, IEEE/ACM Trans. Netw. **15**, 425 (2007).
[7] I. Mabrouki, X. Lagrange, and G. Froc, Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools (Inter-Perf '07) (unpublished).
[8] A. Dvoretzky and P. Erdos, in Proceedings of the 2nd Berkley Symposium on Mathematical Statistics and Probablity, University of California Press, Berkley, 1951.
[9] H. Larralde, P. Turnfio, S. Havlin, H. E. Stanley, and G. H. Weiss, Nature (London) **355**, 423 (1992).
[10] http://www.kazaa.com/
[11] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (unpublished).
[12] G. H. Weiss, *Aspects and Applications of Random Walk* (North-Holland, Amsterdam, 1994).
[13] H. Guclu and M. Yuksel, IEEE Trans. Parallel Distrib. Syst. **20**, 667 (2009).
[14] C. Cooper and A. Frieze, Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (unpublished).
[15] J. Jonasson and O. Schramm, Electron. Commun. Probab. **5**, 85 (2000).
[16] P. Matthews, Ann. Probab. **16**, 189 (1988).
[17] N. Alon, C. Avin, M. Kouck, G. Kozma, Z. Lotker, and M. R. Tuttle, Proceedings of the 20th ACM Annual Symposium SPAA'08 (unpublished).
[18] H. Larralde, P. Turnfio, S. Havlin, H. E. Stanley, and G. H. Weiss, Phys. Rev. A **45**, 7128 (1992).
[19] S. B. Yuste and L. Acedo, Phys. Rev. E **61**, 2340 (2000).
[20] Mersenne random number generator has been used. Mersenne Twister homepage: http://www.math.sci.hiroshima-u.ac.jp/mmat/MT/emt.html
[21] W. X. Wang, B. H. Wang, C. Y. Yin, Y. B. Xie, and T. Zhou, Phys. Rev. E **73**, 026111 (2006).
[22] K. W. Kwong and D. H. K. Tsang, J. Supercomput. **36**, 265 (2006).
[23] M. Tang, Z. Liu, X. Liang, and P. M. Hui, Phys. Rev. E **80**, 026114 (2009).
[24] I. Stoica and H. Zhang, Proceedings of 8th Network and Operating System Support for Digital Audio and Video (NOSSDAV '98) (unpublished).
[25] M. Gupta and M. Ammar, *Lecture Notes in Computer Science* (Springer, New York, 2003), p. 2816.
[26] L. Mekouar, Y. Iraqi, and R. Boutaba, Journal of Peer-to-Peer Networking and Applications **2**, 2 (2009).
[27] D. J. Watts and S. H. Strogatz, Nature (London) **393**, 440 (1998).
[28] B. Bollobs, *Random Graphs*, 2nd ed. (Cambridge University Press, Cambridge, England, 2001).
[29] Albert-Lszl Barabsi and E. Bonabeau, Sci. Am. **288**, 50 (2003).
[30] A. Fronczak and P. Fronczak, Phys. Rev. E **80**, 016107 (2009).
[31] The flooding algorithm replicates packets as often as needed to forward one packet to each neighbor node.
[32] The expected number of steps needed for a single random walker to visit all the vertices of a finite graph.
[33] The expected time to cover $\lfloor c\times n\rfloor$ nodes of a finite graph G, where $0<c<1$.
[34] h_{max} is the maximum $h(u,v)$ over all ordered pairs of nodes and $h(u,v)$ is the expected time for a random walk starting at node u to arrive at node v for the first time.