Census: A protocol for visiting all nodes in MANETs using biased random walks

Vinod Kulathumani[1], Kenneth Parker[2], Mukundan Sridharan[2], and Anish Arora[2]

[1]West Virginia University, Morgantown WV 26506
[2]The Samraksh Company, Dublin OH 43017

***Abstract*—This paper describes Census, a protocol for node visitation in MANETs. Given a set of tokens, the goal of Census is to ensure that every node is visited by at least one token. Census is a gossip style protocol where the random walks of the tokens are assisted by short, local gradients that guide the tokens towards hitherto unvisited nodes. This achieves fast convergence while eschewing routing structures such as spanning trees that incur a high messaging overhead to maintain under mobility. Analytical bounds on convergence time and message overhead are characterized for single token and multiple token scenarios. The scalability and robustness of Census are demonstrated using simulations in networks ranging from 150 to 4000 nodes.**

***Index Terms*—random walk, MANET, statistical aggregation, gossip, local gradients**

I. Introduction

THIS paper focuses on the problem of node visitation in mobile ad-hoc networks, i.e.., given a set of tokens, the objective is to ensure that every node in the network is *visited* by at least one token. We say that a node is *visited* by a token, when it gets exclusive access to the token – this visitation period can be utilized to add node-specific information into the token. There are many applications for such a service such as voting, computing aggregates (max, min average), statistical counting (i.e., estimating fraction of nodes that satisfy a state predicate). Example scenarios include computing average sensor measurements, counting battalions and ammunition in military networks, and computing aggregate traffic densities in vehicular networks. The service could also be used to provide every node an access to a critical resource in a mutually exclusive manner, such as the use of a shared high-bandwidth link. Note that the problem of node visitation is different from that of token dissemination [1, 2] over the entire network where it suffices for every node to have simply *heard* at least one token.

In static networks with stable links, node visitation and even subsequent token aggregation can be potentially realized by traversing on fixed routing structures such as trees or on network backbones [3, 4, 5]. However, in mobile networks and networks with frequent link changes, topology driven structures like spanning trees are likely to be unstable and incur a high communication overhead for maintenance. We are therefore led to describe in this paper, *Census*, a protocol that exploits the simplicity of random walks to achieve token coverage in mobile ad-hoc networks. Random walks are attractive for MANETs because they are inherently stable in the presence of network dynamics, have no critical points of failure, avoid structure maintenance, and have very little state overhead. In a pure random walk, a node that holds a token picks a random node in its neighborhood and transfers the token to that node. This process is repeated until all nodes have been visited. However, the *cover time* for random walks (time to visit all nodes) is typically high. In order to expedite the cover time using random walks, in this paper we explore the idea of partially guiding random walks using the following two biasing mechanisms: (i) local bias and (ii) multi-hop gradient bias. We have considered a MANET with an underlying motion model that is Brownian in nature for our analysis.

a. **Local bias**: If there are one or more unvisited nodes in the neighborhood (i.e., within the communication range) of a token, the token is passed to one of the unvisited nodes picked chosen at random. Thus, if there is an unvisited node in the neighborhood, it will get a preference in receiving the token. We show that local bias, by itself, achieves fast coverage with a cover time of $O(N.\log(N))$. However, we find that when the fraction of already visited nodes in the network rises beyond a certain threshold, the scheme exhibits a slowdown. This is because when all the nodes within the communication range of a token holder are already visited, the scheme reduces to a canonical random walk until an unvisited node becomes a neighbor. While the order of convergence in relation to N remains $O(N.\log(N))$, the slowdown creates a long tail in the convergence and significantly increases cover time.
b. **Multi-hop gradient bias**: To prevent the random walks from getting stuck in regions of visited nodes while there are still unvisited nodes to be explored, we set up short, temporary gradients to pull the token towards unvisited nodes. For the gradient assisted random walk solution, we show a cover time of $O(N \log N/d)$, where d is the average network density. While the convergence order in terms of network size is same as that of local biasing, the gradient bias avoids the long tail and reduces both cover time and the token passing overhead.

Both forms of biasing exhibit a linear speed up when multiple tokens are used. With gradient bias, the cover time is

This work was supported in part by Defense Advanced Research Products Agency (DARPA)'s Fixed Wireless at a Distance program under contract FA8750-12-C-0278. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. Approval for public release was granted by DARPA on April 25, 2014.

$O(N \log N/kd)$, when k tokens are used in an N node network. Thus, the cover time is O($\sqrt{N} \log N / d$) when $\sqrt{N}$ tokens are used and O($\frac{N}{d \log N} \log(\frac{N}{\log N})$ when $\log(N)$ tokens are used. Note the total token overhead remains the same even when multiple tokens are used.

The gradient assisted random walk introduces a gradient message overhead of $O(N \log N/k)$, to pull tokens towards unvisited nodes. However, the gradient overhead is compensated by reducing the required number of token transfers, while also significantly reducing the total cover time. Also note that the gradient message overhead decreases linearly with number of tokens, while the token message overhead is unaffected by the number of tokens. Hence, when local biasing is used with more tokens, the total message overhead will stay high. But when using gradient bias, the overall message overhead will be much lower.

We corroborate all of our results using ns-3 based simulations of mobile networks ranging from 150 to 4000 nodes. We also quantify the impact of number of tokens and network density on the cover time and message overhead. Our simulation results demonstrate the scalability and robustness of using partially guided random walks for token coverage.

A. Related work

Census utilizes a random walk construct on the tokens to cover the network. There have been a number of studies on cover times for random walks in graphs. Results on the cover time range have been shown to vary from $O(N \log(N))$ for complete graphs to $O(N^3)$ for lollipop graphs. [6,7,8]. Typically, cover times are lower for dense, highly connected graphs and tend to increase as connectivity decreases [12]. A speed-up by a factor of k has been shown when k independent random walks are utilized in the graph [9,10,11]. However, note that all these results have been obtained in the context of static graphs. Cover times for biased random walks in time-varying graphs (relevant for mobile networks) have not been studied to the best of our knowledge, which is the focus of this paper.

That being said, our results on convergence times for node visitation appear to be related to cover times of random walks on certain ***static** geometric graphs* [12], in which nodes are placed uniformly on a unit square and two nodes are connected if and only if their Euclidean distance is less than some radius *r*. It is shown in [12] that if the communication radius is greater than a certain threshold, then the expected cover time is $O(N \log(N))$. Specifically, the critical communication radius proven in [9], translates to a node degree of the order of $\theta(8\, log(N))$. In this paper, we study networks whose average densities do not grow with network size. For such mesh networks (modeled as geometric graphs with uniform degree of connectivity), the expected cover time is known to be $O(N\, log^2 N)$ [12]. By biasing random walks, we show that this bound can be improved to $O(N. \log(N))$. It is shown in [12] that the lower bound on cover time for any geometric graph in $O(N \log(N))$: *the biasing ideas in this paper are shown to achieve this lower bound.*

We note that in previous work, there has been *empirical evidence* of obtaining an $O(N. \log(N))$ cover time for static mesh networks by exploiting some form of choice in random walks [13]. The process, *RWC(d)*, described in [13], selects d neighbors uniformly at random at each step and moves to the least visited vertex in that set. In [13], an experimental study of the process *RWC(d)* is performed on geometric graphs and an improvement in cover time is noted with *RWC(2)* and *RWC(3)*. The locally biased random walk explored in this paper is quite different than random walks with choice. In a locally biased random walk, unvisited nodes are preferred whenever they are available, and if there are no unvisited neighbors, the token is moved to any randomly selected neighbor. Nodes do not keep track of the number of times they have been visited, but rather just keep track of *whether* they have been visited. The motive behind local biasing is to simply move opportunistically towards unvisited nodes whenever they are within range. Moreover, while the results in [13] are empirical, in this paper we have analytically shown that biasing a random walk results in an optimal cover time of $O(N. \log(N))$ for random geometric graphs in a mobile setting. We also show that complementing the local bias with a multi-hop gradient bias can further reduce the cover time by avoiding a long tail, while maintaining the same order of convergence.

Finally, we note that the idea of applying biased random walks for problems such as node visitation, aggregation and counting in mobile ad-hoc networks has not been explored before.

B. Outline of the paper

In Section II, we describe the system model and state the problem. In Section III, we describe the Census protocol. In Section IV, we present an analytical characterization of convergence time and message overhead for biased random walks. In Section V, we provide simulation results. In Section VI, we discuss some implementation considerations for Census in a MANET. We conclude in Section VII.

II. Model and problem statement

We consider a mobile network of N nodes deployed over a two dimensional region whose width and height scale as $\theta(\sqrt{N}\ x \sqrt{N})$ and whose communication range and density are constant irrespective of network size N. We assume a random walk mobility model [15, 16] for the nodes *(not to be confused with the random walk of the tokens in the protocol)*. In this mobility model, at each interval a node picks a random direction uniformly in the range $[0, 2\pi]$ and moves with a constan speed randomly chosen in the range $[v_{min}, v_{max}]$ for a constant distance γ. At the end of each interval, a new direction and speed are calculated. This model is Brownian in its characteristics; the Brownian model can be described as a scaling limit of this motion model under small step sizes [14]. The random walk motion model results in node locations that are uniformly distributed across the network [15]. Therefore, although the density of the nodes is time varying, we assume that over time the average number of nodes per unit disk communication range is d.

One or more tokens are introduced at uniformly distributed, random locations within the network. The objective of our protocol is to pass the tokens around the network such that every node in the network is visited by at least one token. We say that a node is *visited* by a token, when it gets exclusive access to the token.

III. CENSUS PROTOCOL

The Census protocol consists of three components: (i) token passing, (ii) gradient setup, and (iii) termination detection. In this section, we describe each of these components. To accomplish these tasks, each node stores three variables, *visited*, *holder* and *level*. The variable *visited* is a Boolean which keeps track of whether a node has been visited by any of the tokens. Initially, *visited=0* at all nodes. When a token first arrives at a node, *visited* is set to 1. Tokens are assumed to be initiated at a random set of nodes. All nodes in which a token is initiated are marked as visited by default and the token value is initialized to the data at the corresponding node. The variable *holder* is used to identify nodes that currently hold a token. When a gradient bias is used, each node also participates in a gradient setup process to attract tokens towards unvisited nodes. To do so, each node uses the state variable *level* where $0 \le level \le 1$. Nodes that are unvisited are at *level=1*. Nodes that hold a token set *level=0* as soon as they receive a token.

A. Token passing

1) Token passing with local bias only

For the random walks with only a local bias, a token holder announces that it has a token. Nodes that hear this message send a request at a random slot within a chosen interval T_r. A timer T_r is started at the token holder to accept requests for the token. The token holder picks a random unvisited node if at least one unvisited node sends a request. Otherwise, the token holder picks a random visited node. The token is transferred to the chosen node. The node that receives the token marks itself as visited if it was unvisited so far. If the token is used for data aggregation, an already visited node may not add its information again to a token. This concludes the procedure for token passing using random walks with only a local bias. The token is continued to pass iteratively using this procedure. There is no deterministic method for termination detection when using random walks with only local bias.

2) Token passing with gradient bias

For the random walks with a gradient bias, a token holder announces that it has a token. Nodes that hear this message send a request at a random slot within a chosen interval T_r. A timer T_r is started at the token holder to accept requests for the token. All nodes with *level>0* randomize their response time and reply to the token announcement along with their current level. Nodes with *level>0* are nodes that have either not been visited (*level=1*) or nodes that have been visited and are now part of a gradient ($0 < level < 1$). The token holder stores all requests received during time T_r. The replies are sorted based on the *level* of the requestors and the token is sent to the node with the highest *level*. When multiple requestors exist with the same *level*, the token recipient is chosen randomly among that set. Thus if any unvisited node requests a token, the token will be sent to that node. If all nodes that have currently requested the token have been visited, the token is sent to the node with the highest value of *level*, which is expected to be the node

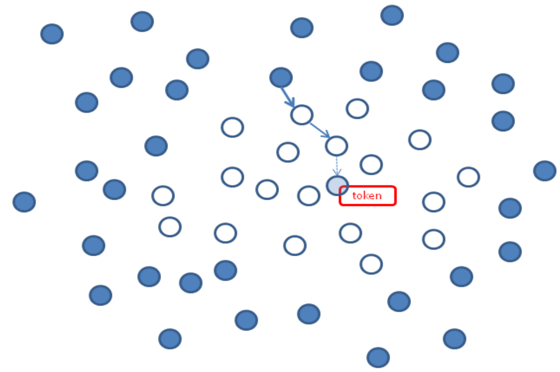

Figure 1: During token passing phase, the token may be surrounded by an island of visited nodes (white circles), i.e., all neighboring nodes have already been visited. Nodes that have not yet been visited (indicated by dark circle) periodically set up a gradient using the set of visited nodes and attract the token towards them. The token moves towards the closest unvisited node that is currently advertising by following the node with the highest value of *level*.

that is closest to an unvisited node. As soon as a token reply has been sent, the node resets *holder* to 0.

B. Gradients

1) Gradient setup

During the Census operation a token can get stuck inside a region where all its neighbors have already been visited. To recover from such a scenario, a gradient is setup in the network to attract tokens towards unvisited nodes, i.e., nodes with *level=1* (See Figure 1). This is done as follows. Nodes with *level=1*, for which none of their neighbors currently hold a token and have at least one neighboring node with *level=0*, initiate a gradient setup by broadcasting a gradient message. Nodes with *level=0* that receive a gradient message from update their level to half of sender's *level* and rebroadcast the gradient message. Thus, gradient broadcasts propagate only till the region where nodes with non-zero level are present, filling up the gap between an unvisited node and other nodes with non-zero levels.

2) Gradient refresh

To account for node mobility, gradients have to be periodically refreshed. To do so, when a node updates its *level* from zero to some non-zero value < 1, it starts a timer proportionate to the new *level* and when the timer expires it resets its *level* back to 0. Thus nodes with higher values of *level* are refreshed slower than smaller values. This heuristic is based on two reasons. (1) Gradients should preferably not be refreshed before a token is able to climb up a gradient and reach an unvisited node. By refreshing at a rate proportional to the value of *level*, a token gets more time to reach closer to the source of the gradient. (2) Nodes that are far away from an unvisited node (closer to the bottom of the gradient) should prevent blocking of gradient setup from unvisited nodes that are nearby, for extended periods of time.

C. Termination detection

When using Census with only locally biased random walks, there is no deterministic way to detect termination. As the network gets closer to being completely visited, the interval between reaching unvisited nodes increases. This may be used to design an approximate threshold for termination detection.

However, when using *Census* with gradient bias, termination can be deterministically detected if the network is connected, i.e., there are no partitions. This procedure is described here. When all nodes have been visited, the gradient setup will be terminated because the gradient setup is only initiated by nodes that have not been visited. As a result, a node that holds a token will continue to get only a level 0 reply for its token announcement. If a gradient is being setup, there would be at least one neighboring node with a value of *level* > *0*. Therefore, when nodes holding the token get a level 0 reply from all its neighbors over an interval greater than the gradient refresh interval, the holder nodes can conclude that all nodes in the network have been visited.

Note, however, that if the network is temporarily partitioned and if unvisited nodes are not connected to any of the nodes carrying a token, termination will be falsely detected.

IV. Census Analysis

In this section, we quantify the expected bounds on convergence time and message overhead for *Census* in the presence of different number of tokens.

Theorem 1: *Both the expected convergence time and the expected number of token transfers in Census with gradient bias in a connected, mobile network of N nodes with average density d and a single token are* $O(N(1+\frac{log(N)}{d}))$.

Proof: There are on average d neighbors for each node. Let z denote the fraction of nodes that have been visited at some given time. Note that the expected number of unvisited neighbors is greater than 1 as long as $zd \le 1$, i.e. $z \le 1/d$. Thus for a fraction $\left(1-\frac{1}{d}\right)$of the nodes, the average distance traveled by a token is 1.

Once the fraction of visited nodes exceeds $1/d$, i.e., $z > 1/d$, the gradients will be used to pull the token towards unvisited nodes. Now, note that there are $4d$ nodes within a 2 hop distance of a token holder. As long as one of these nodes is unvisited, the token will be pulled towards that node. The expected number of unvisited neighbors in a 2 hop range is greater than 1 as long as $z \le 1/4d$. Thus for a fraction $\left(\frac{1}{d}-\frac{1}{4d}\right)$of the nodes, the average distance traveled by a token is 2.

Continuing up to a maximum distance of $\sqrt{N}$, the total average distance traversed by a token before visiting all nodes and the average time for complete coverage is given by the following expression

$$\left(N-\frac{N}{d}\right).1+\left(\frac{N}{d}-\frac{N}{4d}\right).2+\cdots+\left(\frac{N}{\left(\sqrt{N}-1\right)^2 d}-\frac{N}{d}\right).\sqrt{N}$$

$$= N+\frac{N}{d}\left(1+\frac{1}{2}+\frac{1}{3}+\ \ldots+\frac{1}{\sqrt{N}}\right)$$

$$\approx N+\frac{N}{d}(\log(N)) \quad \{\textit{Euler's harmonic series approximation}\}$$

$$= O\left(N\left(1+\frac{\log(N)}{d}\right)\right.$$

■

Corollary 1a: *Both the expected convergence time and the average number of transfers per token in Census with gradient bias in a connected, mobile network of N nodes with average density d and k tokens are* $O(\frac{N}{k}(1+\frac{log(N)}{d}))$.

Proof: Note that when k tokens are used for visiting all nodes, each token is on average responsible for an area of N/k, from which the result follows.

■

Using Corollary 1a, we observe that with $\sqrt{N}$ tokens, the expected convergence time is $O(\sqrt{\mathrm{N}}(1+\frac{log(\mathrm{N})}{\mathrm{d}}))$. When $\log(N)$ tokens are used, the expected convergence time is $O(\frac{\mathrm{N}}{\log(\mathrm{N})}(1+\frac{log(\mathrm{N}/\log(\mathrm{N}))}{\mathrm{d}}))$.

Theorem 2: *Both expected convergence time and the expected number of token transfers in Census with local bias in a connected, mobile network of N nodes with average density d and a single token are* $O(N(1+log(N/d)))$.

Proof: There are on average d neighbors for each node. Let z denote the fraction of nodes that have been visited at some given time. Note that the expected number of unvisited neighbors remains greater than 1 as long as $zd \le 1$, i.e. $z \le 1/d$. Thus for a fraction $\left(1-\frac{1}{d}\right)$of the nodes, the average distance traveled by a token is 1.

Once the fraction of visited nodes exceeds $1/d$, i.e., $z > 1/d$, random walk with local biasing exhibits a slow down before reaching the next unvisited node. This is because the token will be randomly traversing an area of already visited nodes. Unlike the gradient bias, the token does not have a path to reach an unvisited node directly. Now, note that there are $4d$ nodes within a 2 hop distance of a token holder. As long as one of these nodes is unvisited, the token will find it with an average visiting time of *4d*. The expected number of unvisited neighbors in a 2 hop range is greater than 1 as long as $z \le 1/4d$. Thus for a fraction $\left(\frac{1}{d}-\frac{1}{4d}\right)$of the nodes, the average distance traveled by a token is 4d.

Continuing up to a maximum search area of pd, where $pd = N$, the total average distance traversed by a token before visiting all nodes and the average time for complete coverage is given by the following expression

$$\left(N-\frac{N}{d}\right).1+\left(\frac{N}{d}-\frac{N}{4d}\right).4d+\cdots+\left(\frac{N}{\left(\sqrt{p}-1\right)^2 d}-\frac{N}{pd}\right)pd$$

$$= N-\frac{N}{d}+N\left(3+\frac{5}{4}+\frac{7}{9}+\ \ldots+\frac{2\sqrt{p}+1}{(\sqrt{p}-1)^2}\right)$$

$$= N-\frac{N}{d}+\ N.\sum_{1=1}^{(p-1)}\frac{2i+1}{i^2}$$

$$< N+\ N.\sum_{1=1}^{(\sqrt{p}-1)}\frac{2}{i}+\ N.\sum_{1=1}^{(\sqrt{p}-1)}\frac{1}{i^2}$$

$$= O(N(1+\log(\sqrt{p}-1))$$

$$= O(N(1+\log(N/d)))$$

■

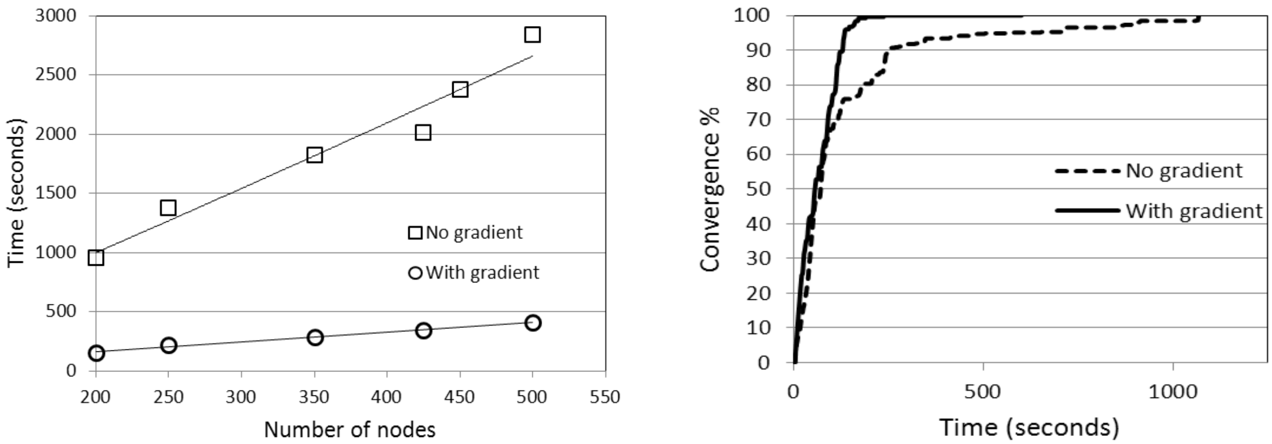

Figure 2: Impact of gradients: (left) Token coverage time as a function of network size with gradient bias and with only local bias. The order of convergence is almost linear for both the cases, but the coverage times are about 6-8 times lower with the gradient bias. (right) The convergence pattern is shown on a network with 250 nodes on the same mobility trace, with and without gradient bias. When using only local bias, a slowdown is observed around the 75% mark and the slowdown progressively increases from that point onwards. The gradient biasing is able to maintain a steady rate of convergence throughout.

Corollary 2a: *Both the expected convergence time and the average number of transfers per token in Census with local bias in a connected, mobile network of N nodes with average density d and k tokens are* $O(\frac{N}{k}(1+log(\frac{N}{d})))$.

Proof: Note that when k tokens are used for visiting all nodes, each token is on average responsible for an area of N/k, from which the result follows.

■

In comparison with the results of Theorem 1, we observe that the order of convergence time remains the same with the local bias. However, a speedup by a factor of d (the average network density) is obtained for *Census* with gradient bias.

We now derive the bounds on the overhead for setting up gradients.

Theorem 3: *The expected gradient message overhead for Census with gradient bias in a connected, mobile network of N nodes with density d and one token are* $O(N(1+log(N/d)))$.

Proof: Following the lines of Theorem 2, we note that for a fraction $\left(\frac{1}{d}-\frac{1}{4d}\right)$of the nodes, the average gradient set up cost will be $4d$since the information from this fraction of nodes will be advertised in a 2 hop neighborhood to pull a token. And for a small fraction $\left(\frac{1}{(\sqrt{p}-1)^2 d}-\frac{1}{pd}\right)$ of the nodes, the gradient will be advertised across the breadth of the network with a cost of zd, where $pd = N$. The result then follows from summing up the series as shown in the proof of Theorem 2.

■

Corollary 3a: *The expected gradient message overhead in Census with gradient bias in a connected, mobile network of N nodes with density d and k tokens is* $O(\frac{N}{k}(1+log(\frac{N}{d})))$.

Proof: Following the lines of Theorem 2, we note that for a fraction $\left(\frac{1}{d}-\frac{1}{4d}\right)$of the nodes, the average gradient set up cost will be $4d$. However, when k tokens are used for visiting all nodes, the maximum size of gradients reduces by a factor of k. So, for a small fraction $\left(\frac{1}{(\sqrt{p}-1)^2 d}-\frac{1}{pd}\right)$ of the nodes, the gradient will be advertised across the breadth of the network with a cost of pd, where $pd = N/k$. The result then follows from summing up the series as shown in the proof of Theorem 2.

■

Thus, we note that there is an extra overhead to pull the tokens towards unvisited nodes, but this is compensated by reduction in the number of required token transfers and reduction in convergence time. Also note that the *gradient message overhead decreases linearly with number of tokens, while the token message overhead is unaffected by the number of tokens.* Hence, when local biasing is used with more tokens, the total message overhead will stay high. But when using gradient bias, the overall message overhead will be much lower. We quantify these results using simulations in the following section.

V. CENSUS SIMULATIONS

In this section, we describe the results of our simulation using ns-3. We set up MANETs ranging from 125 to 4000 nodes with varying number of tokens. The tokens are initiated at random location within the network. The random 2D-walk mobility model is used where the nodes move in a certain direction for a fixed distance and then choose a new random direction. The node speeds are randomly chosen in the range of 2-4m/s. Note that the topology is time varying and there can be temporary partitions in the network. However, the deployment area and communication range are chosen such that the average neighborhood size remains constant irrespective of network size. Specifically, we test using two different densities with an average neighborhood size of 7 and 10 respectively.

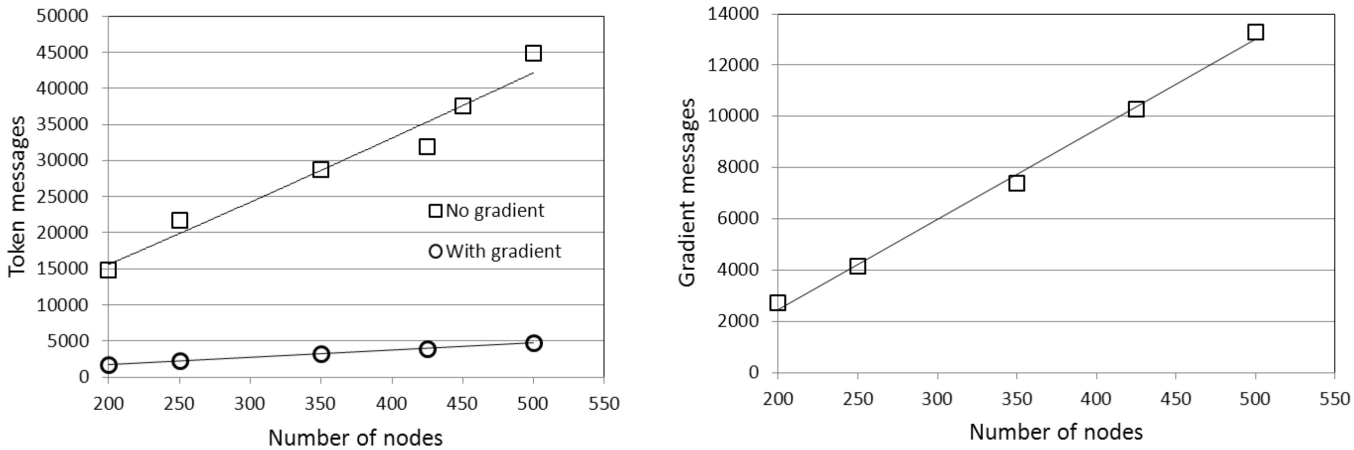

Figure 3: Message overhead: (left) Token message overhead as a function of network size with gradient bias and with only local bias. This graph follows the same pattern as the convergence time as described in our analysis. The token message overhead is about 6-8 times lower with the gradient bias. (right) Gradient message overhead (when using the gradient bias) as a function of network size. This graph is approximately linear with the network size. We also observe that the sum of the token and gradient messages in the gradient bias scheme is lower than the token message overhead for the local bias scheme.

A. *Impact of gradients*

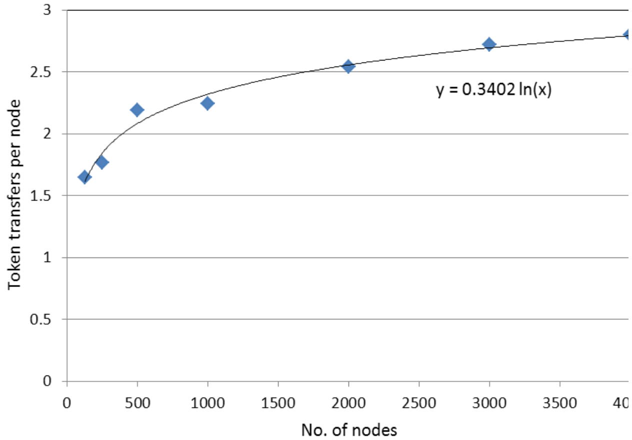

Figure 4: The number of token transfers per node when using gradient bias.

In this section, we compare the performances of Census with local and gradient bias respectively. In this experiment, we have used a single token and networks ranging from 200 to 500 nodes. The average neighborhood size is 7. Figure 2(a) shows the coverage time as a function of number of nodes. We observe that the order of convergence is almost linear with N. The gradient bias improves the convergence time by a factor of about 6-8. In Figure 2(b), we compare the convergence pattern for both these schemes on a network with 250 nodes on the same mobility trace. It is observed that initially both the biasing schemes proceed at almost identical rates towards convergence. When the fraction of visited nodes reaches about 75%, the local bias scheme starts to slow down. The rate of slow down progressively increases as the convergence gets closer to 100%. On the other hand, *Census* with gradient bias proceeds at a steady rate.

Next, we compare the message overhead. Figure 3(a) shows the token message overhead for both these schemes. Token messages include the announcement, request and handoff. As expected, this overhead follows a similar pattern to the convergence time and we see a significant reduction of messages with the gradient bias. In Figure 3(b), we plot the gradient message overhead for Census with gradient bias. We see that this is roughly linear. We also observe that the sum of the token and gradient messages in the gradient bias scheme is lower than the token message overhead for the local bias scheme. At the same time, we see a reduction in convergence time.

In Figure 4, we show the number of token transfers that take place normalized to the number of nodes in the network, using the gradient bias scheme. This ratio ranges from 1.6 to 2.7 as the network size goes from 125 to 4000 nodes. This growth is logarithmic, matching our analysis in Theorem 1. This shows that the redundant token transfers are quite low and demonstrates the efficiency of using the gradient bias to support random walks.

B. *Using $\sqrt{N}$ and $log(N)$ tokens*

In this section, we quantify the impact of using multiple tokens, where the number of tokens is a function of the network size. We use *Census* with gradient bias.

In Figure 5(a), we show the impact of using $\sqrt{N}$ tokens. The network sizes that we simulate are 125, 250, 500, 1000, 2000 and 4000. The corresponding number of tokens used in the network is 11, 15, 22, 31, 42 and 62 respectively. We observe that the coverage time grows only as $O(\sqrt{N})$, matching our analysis.

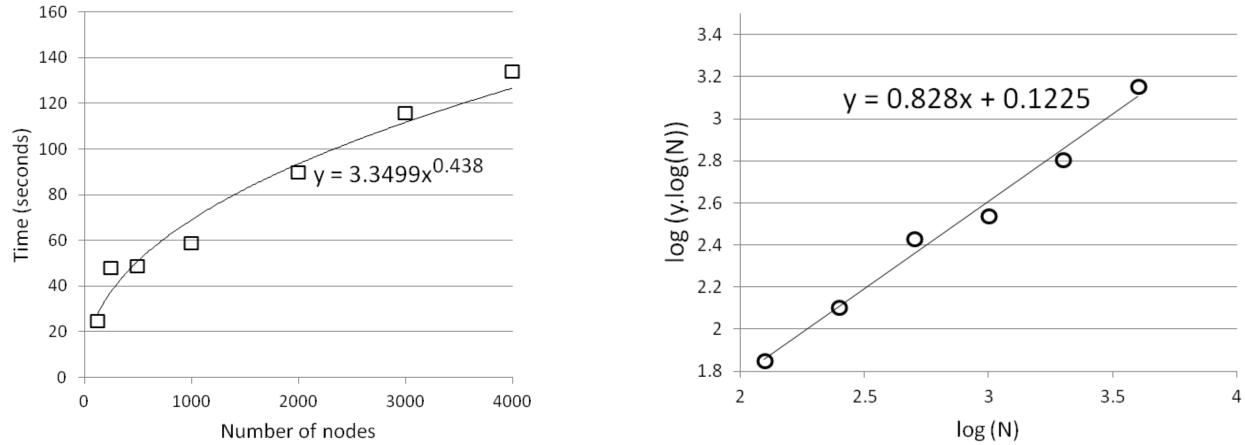

Figure 5: Using $\sqrt{N}$ and $log(N)$ tokens: (left) Token coverage time as a function of network size when using $\sqrt{N}$ tokens. We observe that the coverage time grows only as $O(\sqrt{N})$. The best fit trend line is also shown. (right) Token coverage time as a function of network size when using $\log(N)$ tokens. The x-axis in this graph is log(N). The y-axis shows $\log(y.\log(N))$, where y is the coverage time. The graph is observed to have a slope of approximately 1, showing that $y = O(N/\ logN)$.

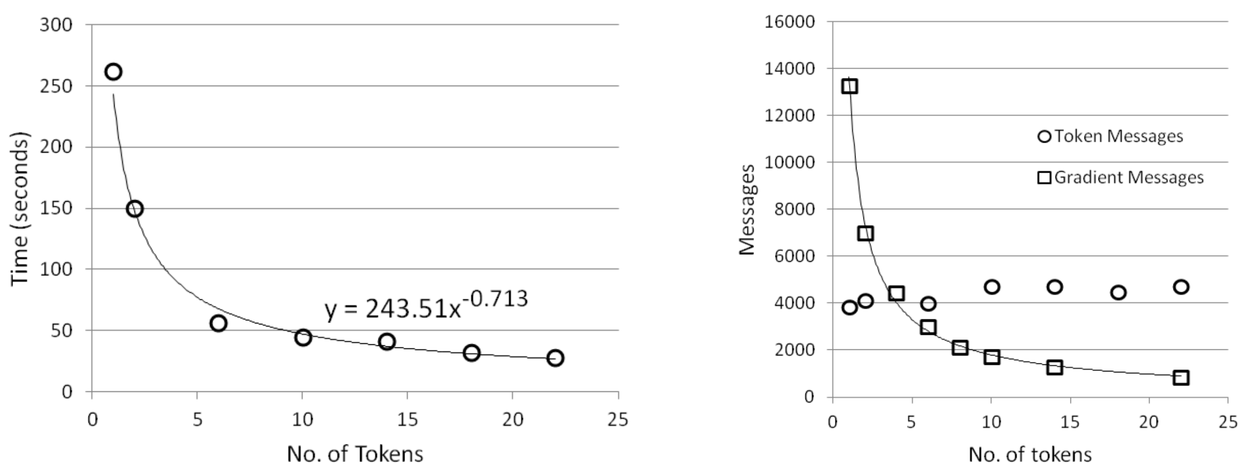

Figure 6: Impact of multiple tokens: (left) Token coverage time as a function of number of tokens in a network of 500 nodes. We observe that the coverage time falls almost linearly with the number of tokens. The best fit trend line is also shown. (right) Token message overhead and gradient message overhead as a function of the number of tokens. The token message overhead stays almost constant. The gradient message overhead falls linearly with the number of tokens.

In Figure 5(b), we show the impact of using $\log_2(N)$ tokens. The network sizes that we simulate are 125, 250, 500, 1000, 2000 and 4000. The corresponding number of tokens used in the network is 7, 8, 9, 10, 11 and 12 respectively. To better understand the trend, we show a log-log plot. The x-axis in Figure 5(b) is log(N). The y-axis shows the values of $\log(y.\log(N))$, where y is the coverage time. The graph is observed to have a slope of approximately 1, showing that $y = O(\frac{N}{\log(N)})$.

C. *Impact of multiple tokens*

In this section we quantify the impact of using multiple tokens. We choose a network size of 500 nodes and vary the number of tokens from 1 to $\sqrt{500}$.

In Figure 6(a), we plot the coverage time as a function of the number of tokens. It is observed to fall almost linearly with the number of tokens, matching our analysis.

In Figure 6(b), we quantify the impact of multiple tokens on message overhead. We observe that the total number of token messages stay roughly constant. However, the number of gradient messages decreases linearly with number of tokens, matching our result in Corollary 3a. This result is significant because it shows that the gradient message overhead introduced by the gradient bias can be significantly reduced, while still retaining the benefit of lower coverage time and lower token overhead compared to local bias. For example, in Figure 6(b), we see that with 10 tokens, the gradient message overhead is reduced to about 2000, while the token overhead remains about 4000 messages. For the local bias scheme, even with 10 tokens, the total token overhead will stay at around 45000 messages as seen in Figure 3(a) for the single token case.

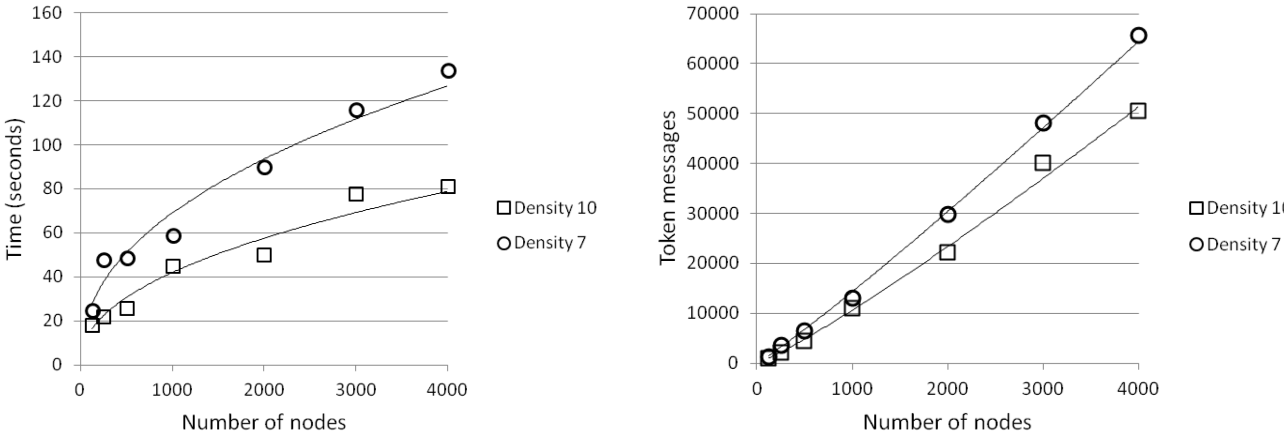

Figure 7: Impact of density: (left) Token coverage time as a function of network size when using $\sqrt{\text{N}}$ tokens in networks with average neighborhood size of 7 and average neighborhood size of 10. We observe reduction in coverage times at higher density. (right) Token message overhead as a function of network size when using $\sqrt{\text{N}}$ tokens in networks with average neighborhood size of 7 and average neighborhood size of 10. We observe reduction in token message overhead at higher density.

D. *Impact of network density*

Finally, we show the impact of network density (average neighborhood size) on the coverage time and message overhead of *Census* with gradient bias.

In Figure 7(a), we compare the coverage time with $\sqrt{N}$ tokens in a network with average density of 7 and average density of 10 neighbors. The graph shows that as density increases, the coverage time decreases, matching our result in Theorem 1.

In Figure 7(b), we compare the token message overhead for both these densities. This graph also shows improvement in networks with higher density.

VI. IMPLEMENTATION CONSIDERATIONS

In this section, we discuss two issues that are not related to the core idea of using random walks for token coverage, but nevertheless are important in the context of implementing *Census* in a MANET.

A. *Reliable token transfer*

Reliable token transfer is critical for successful operation. If a token is released by a node, but the intended recipient did not receive the token reply message, the token is lost. At the same time, if the sending node relies on acknowledgements to release a token, it is possible that the acknowledgements are lost and duplicate tokens are created. For applications where duplicate counting is not permitted, this is a problem.

This issue can be addressed in practice by using acknowledgments in conjunction with checkpoints. The procedure is described below.

As soon as a token reply has been sent, the sender releases the token (the node resets *holder* to zero). At the same time, it remains in a waiting state for acknowledgements from the recipient. If an acknowledgement is not received within a time T_a, the token send message is repeated up to a maximum of K re-tries. If the recipient receives the token multiple times, it simply repeats the acknowledgement message. However, if the token sender does not receive the acknowledgement even after K retries, it creates a *checkpoint* for the token: (a) the aggregate computed thus far is appended to the token along with the token id, (b) a fresh token id is created (unique ids can be created by simply assigning a node's id to the token during creation) and (c) the token aggregate is reset. It is possible that the token was actually successfully passed, but even in this case the checkpoint will not create duplicate counting. At the same time, the process ensures that data is not lost.

B. *Token exfiltration for computing overall aggregate*

This section pertains to specific applications of Census such as aggregation and statistical counting where the goal is collect aggregates at one or more nodes. In these applications, once the initiated tokens have visited all nodes, it is necessary to ex-filtrate the tokens to a given location such as the operating base station or to one or more nodes in the network. As such, token exfiltration is orthogonal to that of node visitation and can be achieved using multiple methods. However, for the application of Census to be meaningful in the context of aggregation, a structure-free method is required. We describe two simple ideas here.

One solution is to simply flood the aggregate tokens across the network in $O(D)$ time (where D is the network diameter) with an $O(Nk)$ message overhead where k is the number of tokens. This leads to a potential question: why not use flooding or diffusion based approaches all the way. Note that the cost of disseminating data from each node to all other nodes is $O(N^2)$ where N is the number of nodes in the network. To reduce this cost, the flooding of individual data can be avoided by opportunistically aggregating information before rebroadcasting and thus essentially *diffusing* the aggregate information across the network. The hurdle in doing so is that the knowledge of nodes whose information has already been included in the aggregate is needed so as to avoid duplicate counting. The inclusion of already counted node ids in the diffused messages cause the message size to grow as O(N), thus effectively making the overall message cost as $O(N^2)$. By using a fixed number of k tokens to first compute the aggregates and then flooding the aggregates, the cost is only $O(Nk)$.

Another potential solution is to transmit the k aggregated tokens using a long distance transmission link (such as cellular or satellite) in hybrid MANETs where the long links are used for infrequent, high priority data.

VII. Conclusions and Future Work

In this paper, we explored the possibility of using random walks for the token coverage problem in MANETs, where the goal is for one or more tokens to visit every node in the network. Noting that a simple random walk has high cover times, we introduced two forms of biasing to partially guide random walks. A local one-hop bias, where the token prefers an unvisited neighbor whenever available, itself reduces the coverage time significantly to an order complexity of $O(Nlog(N))$. However, when a critical fraction of nodes have already been visited, the scheme exhibits a slow-down and creates a long tail before complete convergence. To redress this shortcoming, we introduced a temporary multi-hop gradient bias to pull the tokens towards unvisited nodes. While the scheme still has a similar convergence time of $O(Nlog(N))$, it avoids a long tail and significantly reduces the coverage time as well as the token passing overhead. Our analysis is supported by simulations in ns-3 for networks ranging from 125-4000 nodes, thus demonstrating the scalability of random walks for the token coverage problem in MANETs. We also described how Census can be applied towards one shot aggregation problems in MANETs by providing complementary techniques for token ex-filtration.

Moving forward, we expect to realize some optimizations for the gradient setup. For instance, our analysis in this paper shows that the gradients start having an impact only after a critical fraction of nodes has been visited. To exploit this, we would like to explore on-demand gradient setup (where token holders request a gradient when needed) and event triggered gradients where nodes estimate locally when it would be productive to setup gradients. We would also like to analyze in more detail, partial cover times for both the biasing techniques and understand the message overhead and productivity as a function of convergence percentage. This study will be useful in scenarios where visiting the entire network is not a requirement. Finally, we would like to extend our evaluation to other mobility models and network topologies such as random way point, Manhattan grid, Bonn, Linear and Gauss-Markov.

References

[1] Y. Chen, S. Shakkottai, and J. Andrews, “On the role of mobility on multi-message gossip”, IEEE Transactions on Information Theory, 56(12), pp. 3953—3970, 2013

[2] P. Levis, N. Patel, D. Culler, and S. Shenker, “Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks”, In *Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation - Volume 1* (NSDI'04), 2004.

[3] V. Naik, A. Arora, P. Sinha, H. Zhang, “Sprinkler: A Reliable and Energy Efficient Data Dissemination Service for Extreme Scale Wireless Networks”, *IEEE Transactions on Mobile Computing,* 6(7), pp. 777-789, 2007.

[4] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss and P. Levis, "Collection Tree Protocol," in *7th ACM Conference on Embedded Networked Sensor Systems*, 2009.

[5] S. Madden, M. Franklin, J. Hellerstein and W. Hong, "TAG: A tiny aggregation service for ad hoc sensor networks," in *ACM/USENIX Symposium on Operating Systems Design and Implementation*, 2002.

[6] L. Lovascz, "Random walks on graphs: A survey," *Combinatorics, Paul Erdos 80,* 1993.

[7] U. Feige, "A tight lower bound for the cover time of random walks on graphs," *Random structures and algorithms,* vol. 6, no. 4, pp. 433--438, 1995.

[8] U. Feige, "A tight upper bound on the cover time for random walks on graphs," *Random structures and algorithms,* vol. 6, no. 1, pp. 51--54, 1995.

[9] N. Alon, C. Avin, M. Koucky, G. Kozma, Z. Lotker and M. Tuttle, "Many Random Walks are faster than one," in *20th Annual symposium on parallel algorithms and architectures*, 2008.

[10] A. Frieze and C. Cooper, "The cover time of random regular graphs," *SIAM Journal of Discrete Mathematics,* vol. 18, no. 4, pp. 728--740, 2005.

[11] C. Cooper, A. Frieze and T. Radik, "Multiple random walks in random regular graphs," *SIAM Journal of Discrete Mathematics,* vol. 23, no. 4, pp. 1738--1761, 2009.

[12] G. Ercal and C. Avin, "On the cover time of random geometric graphs," in *Automata, Languages, and Programming*, 2005.

[13] C. Avin and B. Krishnamachari, "The power of choice in random walks: An empirical study," in *International Symposium on Modeling, Analysis and Simulation of Wireless Mobile Systems,* 2006.

[14] G. F. Lawler and V. Limic. 2010, “*Random Walk: A Modern Introduction*”. Cambridge University Press, New York, NY, USA

[15] J.-Y. Le Boudec and M. Vojnovic, “Perfect simulation and stationarity of a class of mobility models,” in Proc. IEEE INFOCOM, 2005, pp. 2743–2754

[16] P. Nain, D. Towsley, B. Liu, and Z. Liu, “Properties of random direction models,” in Proc. IEEE INFOCOM, 2005, pp. 1897–1907