1 Introduction

In the near future, intelligent mobile wireless sensors will be extensively deployed in military operations, under-sea explorations, hazardous environments, etc. The objectives for the nodes in such mobile wireless sensor networks is to move rapidly, probe, process and transmit information to other nodes. At the end of the "operation" only a subset of the sensor network nodes are recovered (the rest are either lost or severely damaged). Information is retrieved from what is stored in the subset of recovered nodes. Since the sensors have limited computational power, the nodes have to balance between energy conservation and fault-tolerance while transmitting information.

Gossip algorithms lend themselves well to such applications where each node selects at random a neighbor and transmits a quantum of information (refered to as *gossip*). There are several articles in the open literature on gossip algorithms (such as Haas et al [4] on routing, Vogels et al [11] on robustness, and Servetto and Barrenechea [10] on scalability). A complete literature review is not performed due to space restrictions. Only a limited number of research articles such as Koldehofe [7] consider analytical models for performance analysis of gossip algorithms. In addition, only a few articles such as Kowalczyk and Vlassis [8] explicitly consider stopping criteria in their gossip algorithms.

The remainder of this paper is organized as follows. In Section 2 we propose a gossip algorithm suitable for applications described above. Then in Section 3 we model the information flow and node states using a Markov chain. By suitably modifying this Markov chain, we analyze the performance to obtain measures such as: time to transfer information and fraction of nodes receiving information in Section 4. We describe numerical results and findings in Section 5. Then we compare our algorithm to existing stopping-criterion based gossip algorithms in Section 6. Finally we present our concluding remarks and directions for future work in Section 7.

2 Gossip Algorithm Description

Consider a mobile wireless sensor network with N nodes. We propose a gossip algorithm for nodes to dissipate sensed information to as many nodes as possible in the network. We specifically concentrate on a scenario where the sensor nodes move around, they sense and process information which they periodically transmit to other nodes. The transmission times are far apart that the nodes would have significantly moved between two transmissions. Between successive transmissions, the nodes sense, process and store information. We assume the synchronous time model described in Boyd et al [1] and used in Karp et al [5] as well as Kempe et al [6] (extending it to the asynchronous time model described in Boyd et al [1] would be fairly straightforward and in fact easier). In the synchronous time model, all nodes at a prescpecified transmission time, synchronously transmit a set of information to one of their neighbors. Each node selects one neighbor and transmits only the information that the neighbor does not posses. To describe the algorithm as well as for analysis, we will consider only a single piece of information that we call *gossip* to study how fast and wide it can spread.

Note that the nodes (i) have local knowledge, (ii) have limited computational power, (iii) make distributed decisions, and (iv) move rapidly over time. With those in mind, we propose the following gossip algorithm. Consider a gossip that was originated at a certain node. During the next transmission phase the node picks one of its neighbors at random to transmit the gossip. Since the nodes are moving rapidly, we assume that the probability that a certain node is selected is $\frac{1}{N-1}$, even though all nodes may not be candidate neighbors. At the next transmission phase each of these two nodes that know the gossip, selects a neighboring node at random with probability $\frac{1}{N-1}$. In this manner during every transmission phase, every node that has the gossip (and has not stopped spreading it) selects one of the $N-1$ other nodes to check if it has the gossip. If the selected node already has the gossip, then the transmitting node not only does not transmit the gossip but also stops speading it; else it continues spreading the gossip. If two or more nodes attempt to transmit the gossip to a node that does not have the gossip, then only one of the nodes transmits the gossip but all the nodes involved continue to spread the gossip.

It is worthwhile to notice the following characteristics with respect to the proposed gossip algorithm: (i) there is an explicit stopping criterion for each node to stop spreading the gossip, i.e. when the node attempts to transmit to another node that already has the gossip; (ii) at each transmission phase, there are three types of nodes, those that are actively spreading the gossip, those that stopped spreading the gossip and those that do not have the gossip; (iii) the algorithm ends when there are no actively spreading nodes with gossip and it is not necessary that all nodes get the gossip eventually; (iv) when the algorithm ends, the number of transmissions that occured is one less than the number of nodes that have the gossip (thus computing power is used prudently). With these characteristics in mind, we can now model and analyze the process of gossip spreading via the algorithm proposed above.

3 Markov Model

As described in Section 2, at the beginning of a transmission phase, each node is in one of the following three states: knows gossip and is spreading it, knows gossip but stopped spreading, and does not know gossip. In that light, we define the following: let X_n be the number of actively spreading nodes at the n^{th} transmission phase; let Y_n be the number of nodes that have not heard the gossip upto the n^{th} transmission phase, and let Z_n be the number of nodes that have heard the gossip until the n^{th} transmission, but have stopped spreading it. Note that since $Z_n = N - X_n - Y_n$, we really need only X_n and Y_n to describe the state of the system and Z_n is defined purely for notational convenience. Also, $X_0 = 1$, $Y_0 = N-1$ and $Z_0 = 0$ is the initial state when one node has the gossip and the remaining $N-1$ do not know it. Further, in order to predict the state (X_{n+1}, Y_{n+1}) all we need to know is (X_n, Y_n). Therefore the process $\{(X_n, Y_n), n \geq 0\}$ is a discrete time Markov chain (DTMC).

For this DTMC model, the next step is to obtain the transition probabilities. Let $p_{i,j}(i-l+m, j-m)$ be the probability of transitioning from state $(X_n = i, Y_n = j)$ to state $(X_{n+1} = i-l+m, Y_{n+1} = j-m)$. That happens when l of the i nodes that are active in spreading the gossip end up attempting to spread to nodes that already have the gossip. Also, the remaining $i-l$ nodes spread the gossip to a set of m nodes out of the j nodes that do not have the gossip. There are some constraints such as $0 \leq m \leq \min(j, i-l)$ and $0 \leq l \leq i$. In order to obtain an algebraic expression for $p_{i,j}(i-l+m, j-m)$, consider the following matching problem: there are W eligible women and M eligible men in a society and each of the W women selects a man at random (assuming all men are equally desirable!) and writes a letter.

Lemma 1 *The number of combinations of m of the M men receiving letters (that means $M-m$ do not receive letters) is $\Omega(M,W,m)$ and is given by*

$$\Omega(M,W,m) = \binom{M}{m}\left[\sum_{x=1}^{m}(-1)^{m-x}\binom{m}{x}x^W\right].$$

Proof. By principle of mathematical induction. Details are omitted. ■

Such combinatorial problems have received attention in the literature, especially the stable marriage problem (see Gusfield and Irving [3]). Now, using the above Lemma 1, we can obtain an algebraic expression for $p_{i,j}(i-l+m, j-m)$, the probability that $X_{n+1} = i-l+m$ and $Y_{n+1} = j-m$ given that $X_n = i$ and $Y_n = j$. Let $Z_n = k$ where k is the number of nodes that have stopped spreading the gossip. Clearly, $k = N-i-j$. Also $X_{n+1} = i-l+m$ and $Y_{n+1} = j-m$ imply that $Z_{n+1} = k+l$. The following theorem describes the transition probability $p_{i,j}(i-l+m, j-m)$.

Theorem 1 *For $m = 0$, (and $l = i$ hence)*

$$p_{i,j}(i-l+m, j-m) = \left(\frac{i+k-1}{N-1}\right)^l$$

and for $1 \leq m \leq \min(j, i-l)$ and $0 \leq l \leq i$,

$$p_{i,j}(i-l+m, j-m) =$$

$$\binom{i}{l}\left(\frac{i+k-1}{N-1}\right)^l\left(\frac{j}{N-1}\right)^{i-l}\frac{\Omega(j, i-l, m)}{j^{i-l}}.$$

Proof. The event that l nodes stop spreading the gossip is when l of the i nodes that are actively spreading the gossip end up spreading to nodes that already know the gossip. In other words, out of the i nodes, l of them choose to tell nodes who already know the gossip and $i-l$ of them choose to tell nodes that do not have the gossip. That happens with probability

$$\binom{i}{l}\left(\frac{i+k-1}{N-1}\right)^l\left(\frac{j}{N-1}\right)^{i-l}.$$

Now, of the j nodes that have not received the gossip, m of them get it from the $i-l$ nodes mentioned above. Clearly some (or all) of the m nodes may have been contacted by more than one of the $i-l$ nodes. The total number of different ways j nodes can get the gossip from $i-l$ nodes is j^{i-l} ways. Of these j^{i-l} ways, based on Lemma 1, exactly $\Omega(j, i-l, m)$ would result in m different new nodes hearing the gossip, where

$$\Omega(j, i-l, m) = \binom{j}{m} \left[\sum_{x=1}^{m} (-1)^{m-x} \binom{m}{x} x^{i-l} \right].$$

Hence the expression for the transition probability in Theorem 1. ∎

Care must be taken while constructing the DTMC to ensure that impossible states are removed from the DTMC. For example $(1, N-1)$ is the only state for which $X_n = 1$. Also there cannot be states such as $(0, N)$ or $(0, N-1)$. In essence, all states that are unreachable from $(1, N-1)$ in one or more steps should be removed. Let P be the transition probability matrix of the remaining states of the DTMC such that $P = [p_{i,j}(i-l+m, j-m)]$, i.e. a matrix of $p_{i,j}(i-l+m, j-m)$ values. Now that the state of the system is modeled as a DTMC with transition probabilities obtained from Theorem 1, the next step is to analyze the DTMC to obtain performance measures such as time for gossip spreading to end and fraction of nodes receiving the gossip.

4 Performance Analysis

The DTMC modeled in Section 3 is reducible and transient, since states such as $(0, j)$ are absorbing (i.e. $p_{0,j}(0, j) = 1$) and state $(1, N-1)$ cannot be reached from any other state. Although it is possible to analyze reducible DTMCs, for the purposes of this paper, it is more convenient to transform the DTMC into an irreducible and positive recurrent one (such DTMCs are sometimes called ergodic, for definition and properties refer to Kulkarni [9]).

4.1 Modified DTMC

Consider a modification to the transition probability matrix P such that for all $j \leq N-2$, $p_{0,j}(0, j) = 0$ and $p_{0,j}(1, N-1) = 1$. Let $\hat{P}$ be the new transition probability matrix, but with the same states as the original DTMC. The modification implies that as soon as a gossip spreading ends, a new gossip begins. Therefore every time state $(1, N-1)$ is reached, it is like starting a new replication in a simulation. This modified $\hat{P}$ matrix is such that the DTMC is irreducible and aperiodic. Note that everytime the system reaches $(1, N-1)$, it denotes the gossip spreading ended in the previous transisition. Also if $(1, N-1)$ was reached from $(0, j)$ then $N-j$ nodes ended up receiving the gossip. In that light the two performance measures we are interested are: how long does it take to revisit $(1, N-1)$ (i.e. how many transmission slots does the gossip algorithm last which is a function of how often state $(1, N-1)$ occurs); and how many nodes end up receiving the gossip.

For that, we compute the steady-state distribution of the modified DTMC. Let $\pi_{i,j}$ be the steady-state probabilities of the DTMC with transition probability matrix $\hat{P}$. Therefore,

$$\pi_{i,j} = \lim_{n \to \infty} P\{X_n = i, Y_n = j\}.$$

Clearly the row vector π of $\pi_{i,j}$ values ($\pi = [\pi_{i,j}]$) can be obtained as the left eigen vector of $\hat{P}$ corresponding to eigenvalue of 1 and normalized so that $\sum_{i,j} \pi_{i,j} = 1$. In other words, π is the solution to

$$\begin{aligned} \pi &= \pi \hat{P} \\ \pi e &= 1 \end{aligned}$$

where e is a column vector of 1's. Now, using the $\pi_{i,j}$ values, we develop the performance measures of interest in the next section.

4.2 Performance Measures

For all states (i, j) in the DTMC, $\pi_{i,j}$ can be obtained numerically. Using that, we derive some performance measures in terms of $\pi_{i,j}$. The first measure of performance of interest is the time to complete spreading the gossip. In other words, this is the time for the gossip algorithm to end and the original DTMC to end up in state $(0, j)$. Define τ as the average number of time slots for completion (i.e. stop spreading the gossip).

Theorem 2 $\tau = \dfrac{1}{\pi_{1,N-1}} - 1.$

Proof. For the modified DTMC, since state $(1, N-1)$ is reached every time the gossip spreading is completed, $1/\pi_{1,N-1}$ denotes the average amount of time to leave state $(1, N-1)$ and return back for the first time. This is one extra transmission phase over when the gossip ended. Hence the average time for gossip to end in terms of number of transmission slots is given in Theorem 2. ∎

The second performance measure of interest is the number of nodes that end up getting the gossip when the algorithm completes. This is called *reach.* Let μ and σ be the average and standard deviation of reach, i.e. the mean and standard deviation of the number of nodes that receive the gossip. Clearly,

$$\mu = \sum_{r=2}^{N} r\rho_{N-r}$$
$$\sigma^2 = \sum_{r=2}^{N} r^2\rho_{N-r} - \mu^2$$

where ρ_j is the probability that j nodes do not receive the gossip when the algorithm ends. In order to compute ρ_j, we state the following theorem.

Theorem 3 $\rho_j = \dfrac{\pi_{0,j}}{\sum_{i=0}^{N-2} \pi_{0,i}}$.

Proof. Straightforward conditional probability that the algorithm ends in node j given that the DTMC is in one of the algorithm-ending nodes, i.e. of the form $(0, i)$. ■

Having derived the performance measures τ, μ and σ, the next step is to obtain them numerically. This is done in the next section.

5 Results

Notice that N is the only parameter in the DTMC modeled in Section 3. Therefore for various values of N, we obtain the performance measures τ, μ and σ numerically. For each N, we obtain P first, then convert to $\hat{P}$ and finally the steady-state probabilities $\pi_{i,j}$. Using the steady-state probabilities, we obtain performance measures τ, μ and σ. In this paper we used MATLAB to compute the numerical values.

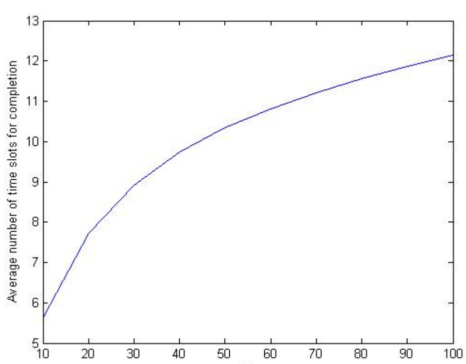

Figure 1. τ **versus** N

Figure 1 is an illustration of how τ, the average number of transmission phases for the gossip algorithm to complete, varies with N. From the figure, it is evident that as the number of nodes N increases, τ increases at a much slower rate and the rate of increase decreases with N. It appears as though τ would eventually flatten out. This implies that the proposed gossip algorithm is extremely scalable.

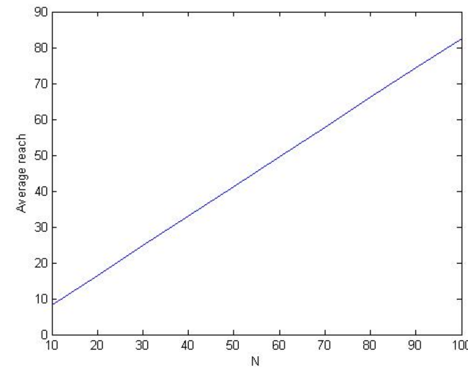

Figure 2. μ **versus** N

The most remarkable finding of this paper is that in Figure 2. Notice that the average number of nodes reached (μ) is linear with respect to N. In fact, numerically it can be shown that $\mu \approx 0.826N$ for $N \geq 5$. What this implies is that for any N, μ can be immediately predicted without even running the algorithm. In other words, on an average 82.6% of the nodes can be reached via the gossip algorithm presented in this paper. Notice that 0.826 is a parameter-free constant. Although this is an asymptotic result, it works for even small values of N.

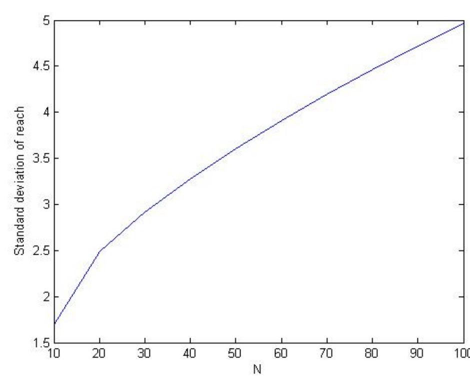

Figure 3. σ **versus** N

Since the number of nodes the gossip reaches is a random variable, in order to study the variability of this random variable, in Figure 3, we plot σ (the standard deviation of the number of nodes that receive the gossip) as a function of N. Although σ does increase with N, the rate of increase reduces with N. From the figure it appears as though σ has an upper bound of $2.5 + 0.025N$. Therefore for larger values of N, σ can be approximated as $2.5 + 0.025N$.

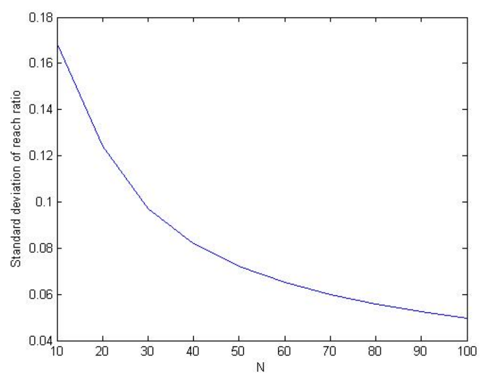

Figure 4. σ/N **versus** N

In Figure 4, we study the standard deviation of the fraction of nodes the gossip reaches. Clearly that quantity, i.e. σ/N is a decreasing function of N. An interesting observation one can make is that with 99% confidence one can state that at least two-thirds of the nodes would receive the gossip, especially for large N. The claim is made using central limit theorem and strong law of large numbers.

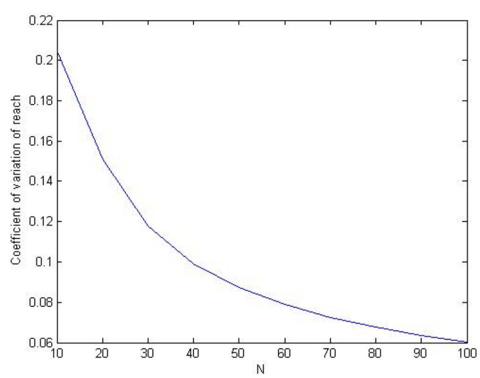

Figure 5. σ/μ **versus** N

Finally, in Figure 5, we study the coefficient of variation of the number of nodes that receive the gossip (also equal to the coefficient of variation of the fraction of nodes that receive the gossip). First of all notice that the coefficient of variation is extremely small. Secondly it is reducing with N which means that the random variable is less varying as N increases.

6 Comparisons

The next question to ask is how our proposed gossip algorithm compares against other algorithms. In order to make our comparison meaningful, we select algorithms that have explicit stopping criteria. There are two such algorithms in the literature. First is the *random walk* algorithm that several researchers have used for comparison. The second is an adaptation of the *multicast gossip* in Ganesh et al [2] modified to our scenario. We now briefly explain the algorithms and compare the performance measures τ and μ against the proposed gossip algorithm in this paper.

6.1 Random Walk

The random walk algorithm works in the following manner: the initial node with a gossip selects one of its neighbors at random and transfers the gossip; then the initial node stops speading the gossip. Then the node with the gossip selects a neighbor at random to spread the gossip: if the neighbor already has the gossip, the node with the gossip picks another node, otherwise the node stops spreading the gossip and gives it to the neighbor. The algorithm continues until all nodes have the gossip. Notice that at any given time only 1 node has the gossip. Clearly $\tau > N - 1$ because it the best case it would take $N - 1$ transmission phases to spread the gossip to all the nodes. This is much higher than our proposed algorithm. However $\mu = N$ because all nodes eventually receive the gossip. Applied to our scenario, this random walk algorithm would be slow and reliable.

6.2 Multicast Gossip

The multicast gossip algorithm works in the following manner: the initial node with a gossip selects few of its neighbors at random and transfers the gossip using a multicast scheme; then the initial node stops speading the gossip. Then the nodes with the gossip select few neighbors at random to spread the gossip: whether the neighbors already have the gossip or not, the node stops spreading the gossip and gives it to the neighbors. If a node that already has the gossip receives it from a neighbor, it does not spread it but remains passive. The algorithm continues until all nodes are passive (some with the gossip and others without). Usually τ is better than our proposed algorithm in this paper. However μ is worse than our proposed algorithm Applied to our scenario, this multicast gossip algorithm would be fast but not-so-reliable.

7 Concluding Remarks

In this paper we proposed a gossip algorithm (with explicit stopping criterion) to route information to as many nodes as possible. In the literature typically gossip algorithms are analyzed using simulations, however we use an analytical model based on Markov chains to obtain performance measures. Also, most gossip algorithms in the literature do not consider stopping criteria explicitly. Besides the gossip algorithm and method of analysis, another contribution of this paper is the curious finding that the average fraction of nodes that eventually receive the routed information is a parameter-free constant. We compare the performance of our algorithm against others in the literature and claim that our algorithm performs very well with a dual objective of being both reliable and fast.

There are a few limitations for this paper. First of all, the algorithm works well only under the framework of the setting mentioned in the paper: nodes move rapidly and only periodically they are in a transmission phase; in addition, it is enough if many (but not all) nodes receive the information. Secondly, numerical studies for only upto $N = 100$ were made. The reason is that the factorial of a large number ends up higher than the largest number in MATLAB. This can be circumvented in the future by either approximations or usage of logarithms cleverly for the factorials. The gossip algorithm itself can certainly be improved such as: continue gossiping until 2-3 attempts result in nodes already knowing the gossip; contact more than 1 neighbor during a transmission phase. These upgrades will be considered in future, however obtaining an analytical model may be difficult in those cases.

References

[1] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Gossip and mixing times of random walks on random graphs. *Proc. IEEE INFOCOM*, 3:1653–1664, 2005.

[2] A. Ganesh, A. Kermarrec, and L. Massoulie. Peer-to-peer membership management for gossip-based protocols. *IEEE Trans. Comp.*, 52(2):139–149, 2003.

[3] D. Gusfield and R. W. Irving. *The Stable Marriage Problem, Structure and Algorithms.* MIT Press, Cambridge, MA, 1989.

[4] Z. Haas, J. Halpern, and L. Li. Gossip-based ad hoc routing. *Proc. IEEE INFOCOM*, 3:1707–1716, 2002.

[5] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. In *Proc. of 41st IEEE Symp. on Foundations of Comp. Science*, 2000.

[6] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proc. of 44th IEEE Symp. in Foundations of Comp. Science*, 2003.

[7] B. Koldehofe. Simple gossiping with balls and bins. *Studia Informatica Universalis*, 3(1), 2004.

[8] W. Kowalczyk and N. Vlassis. *Newscast EM: Advances in Neural Information Processing Systems.* MIT Press, Cambridge, MA, 17 edition, 2005.

[9] V. G. Kulkarni. *Modeling and Analysis of Stochastic Systems.* Chapman & Hall, London, UK, 1995.

[10] S. D. Servetto and G. Barrenechea. Constrained random walks on random graphs: routing algorithms for large scale wireless sensor networks. *WSNA*, 2002.

[11] W. Vogels, R. van Renesse, and K. P. Birman. The power of epidemics: robust communication for large-scale distributed systems. *Computer Communication Review*, 33(1):131–135, 2003.