On Spreading Recommendations via Social Gossip

Yaacov Fernandess
School of Engineering and Computer Science
The Hebrew University of Jerusalem
Israel
fery@cs.huji.ac.il

Dahlia Malkhi
Microsoft Research
Silicon Valley, California
U.S.A
dalia@microsoft.com

ABSTRACT

This paper introduces and analyzes a variant of distributed gossip which is motivated by the sharing of recommendations in a social network. The social settings bear two implications on gossip. First, rumors fade after a few hops, and so does our gossip mechanism. Second, users require a rumor to be substantiated by multiple, independent sources in order to adopt it. Consequently, in our social gossip a message is adopted only when it is received over a threshold of independent paths. Social gossip is a new, highly relevant and practically motivated variant of distributed gossip, whose analysis contributes to the fundamental theory of distributed algorithms.

Categories and Subject Descriptors

C.2.2 [**Computer Systems Organization**]: Computer Communication Networks— *Network Protocols*

General Terms

Algorithms, Theory

Keywords

randomized algorithms, message dissemination, gossip algorithms, epidemic algorithms

1. INTRODUCTION

Whether you are searching the web, browsing for books, or looking for recommendations on child-care, your friends' preferences will likely get you better search results. This vision underlies the growing success of numerous services that are based on social networks such as MySpace, FaceBook, Linkedin, and Yahoo's 360.

Our work takes a formal look at the problem of reliably sharing information in a social network. It arises within the context of the Nocturnal project [1]. The project harnesses the power of social networking to automatically share information with a user's friends, family and business contacts to deliver collaborative recommendations.

Nocturnal provides automated information sharing among users in an instant messaging (IM) network. An agent running on a user's workstation automatically exchanges the user's stored recommendations with immediate IM contacts. The information is relayed to the contacts' contacts and their contacts, and so on. No central server is involved. The information arriving at any user is stored locally, along with tags indicating the simple path it came from. That is, the simple path contains the users through which the recommendation has traveled. The recommendation-sharing infrastructure is generic and is useful to various network applications.

Currently, Nocturnal shares recommendations about webpages in order to provide better web search results. A user's browser enhanced with the tool uses the information gathered from the user's social network to further filter and resort search engine results. This brings recommended URLs to higher ranking, whereas pages reported as low quality are demoted. In addition, the tool enhances the user browsing experience: The tool's graphical user interface renders online feedback, listing page quality and the social path by which the information was passed through the network.

Privacy via gossip.

The core communication mechanism in the Nocturnal network is gossip. Users' recommendations are neither monitored by, nor stored at, a centralized location. Utilizing the existing end-to-end communication IM infrastructure, our approach guarantees that the user information is shared and stored only within his community. Randomized gossip techniques for information dissemination are key in numerous distributed systems. A randomized gossip algorithm operates in (semi-) synchronous rounds, each node selects a partner uniformly at random from all its direct neighbors, and exchanges information with it. In information dissemination gossip algorithms each node starts with a message, and the nodes must spread the messages throughout the network using local communication so that every node eventually has every message. The goal is to minimize the amount of rounds required for all nodes to eventually has every message. However, our social gossip differs from standard gossip in two substantial ways.

First, each user in the Nocturnal network sets a *hop-count bound* that limits the distance from which information is deemed relevant. That is, the user's agent only stores recommendations, later to be verified, that traversed through

SPAA'08, June 14–16, 2008, Munich, Germany.
Copyright 2008 ACM 978-1-59593-973-9/08/06 ...$5.00.

paths shorter than a designated hop-count bound, and discards the information otherwise.

Second, the user's agent accepts the information as trustworthy only if it arrives from sufficiently many independent sources, rather than just one. Having such *adoption threshold* is common in social network theory (see, e.g., [20]). In order to implement the adoption threshold, the gossip protocol in Nocturnal relays not only the gossiped information itself, but also meta-information about the *path* through which it traversed. Each user then accepts a gossip message only if it has been received over $f + 1$ disjoint gossip paths, for some parameter threshold f.

The adoption criterion in Nocturnal is based on *path–verification*, a gossip method that was independently introduced in [27] and in [24] for different settings. Implementing such *path verification* gossip has challenged researchers in a number of previous works, but no proven, practical solution has yet emerged. Exponential protocols appear in [27, 24, 25], requiring a node to incur an exponential computational step in order to find a satisfying vertex–disjoint path. The exponential computation proves impractical, even for a moderate network size. A proposal for a practical heuristic was given in [27], but thus far, has been validated by simulation only.

Our contribution is a practical path verification protocol with computation and storage complexity, which is polynomial in n. In short, it works in two logical phases. The first 'Aggregate' phase takes $O(\log(n))$ rounds. The length of the paths is bounded by $O(\frac{(\log(n\log(n))}{\log\log(n)})$. In the second 'Collect' phase, nodes exchange simple, non-intersecting paths with one another using a greedy linear-time path selection rule. The main body of this paper is devoted to showing that the greedy selection rule suffices for nodes to collect the required $f+1$ disjoint paths efficiently, and that the total completion time of the gossip is asymptotically optimal.

Analysis.

More concretely, we first look at the effect of bounded hop limit on (standard) information dissemination gossip algorithm. Let $A(\ell)$ be a gossip algorithm that spreads messages from origin(s) to within distance ℓ. Since standard gossip (without a hop bound) reaches all nodes within $O(\log(n))$ protocol rounds, having $\ell = O(\log(n))$ essentially poses no hop count constraint, and it suffices to achieve optimal dissemination time. The natural question to ask is what happens with smaller ℓ's. Denote by $x_d(r)$ the number of nodes that have received m over a shortest path of length d before the beginning of round r. At the beginning of round $r + 1$, $x_d(r+1)$ will count those nodes that remain in possession of shortest path of length d from the previous round, as well as new nodes that have obtained m through nodes at distance $d-1$. Clearly then, $x_d(r+1) \leq x_d(r) + x_{d-1}(r)$. From this, it easily follows that $\sum_{i=1}^{\ell} x_i(r) \leq 2r^{\ell}$ (see Lemma 4.1). One immediate result of this upper bound is that for $A(\ell)$ to complete in logarithmic time, $\log^{\ell}(n)$ needs to be linear in n, which implies $\ell \geq \log(n)/\log\log(n)$. More generally, the following theorem states the number of gossip rounds needed for termination for any $\ell \leq \frac{2(\log(n\log(n))}{\log\log(n)}$:

Theorem 1.1. *The number of rounds for termination of $A(\ell)$, where $\ell \leq \frac{2(\log(n\log(n))}{\log\log(n)}$, is $O((n\log(n))^{(2/\ell)})$ in expectation.*

In particular, we obtain that for $\ell = \frac{2(\log(n\log(n))}{\log\log(n)}$, the gossip time is asymptotically optimal in expectancy.

We further incorporate in the analysis our path verification adoption rule. That is, we analyze the time for nodes to obtain information on m over $f + 1$ independent paths. Let $B(\ell, f)$ be our path verification gossip protocol, where ℓ is the path length bound, and by f, the adoption threshold. Our main complexity result is captured in the following theorem:

Theorem 1.2. *The number of rounds for termination of $B(\ell, f)$ in a network where $f(1+\ell) < \frac{n}{2}$ and $\ell = \Theta(\frac{\log(n)}{\log\log(n)})$ is $O(\log n + f)$ with high probability.*

We note that this time complexity is asymptotically optimal: The dissemination of a single message using gossip requires at least $\Omega(\log n)$ rounds. On the other hand, in order for a node to receive the message through $f + 1$ disjoint paths, it requires at least $f + 1$ rounds. Hence the running time is lower bounded by $max\{\log n, f + 1\} = \Omega(\log n + f)$.

Summary.

This paper introduces a new social gossip protocol and its analysis. Our gossip model has a natural, appealing intuition. As a recommendation travels from one user to the next, its relevance decreases. In other words, in terms of trust, the trust put in a recommendation is highest when it comes from a direct contact, and gradually diminishes as it is conveyed from one user to the next. Therefore, when a certain hop-count limit is reached, the trust goes to zero and the message dissemination stops.

The adoption criterion protects the network from spam recommendations: It might be possible for intruders to penetrate the network at a single place, or even manage to hook up with other nodes. However, given the security threshold f, a message needs to be carried over $f + 1$ non-overlapping simple paths to be validated. Thus, recommendations from bad sources are presumed impossible to reinforce.

Moreover, good recommendations automatically reinforce themselves. Each node becomes an active origin itself once the adoption criterion has been met. This, in turn, enhances the recommendation's influence on the network.

Our path-verification protocol and its analysis contribute to the theory of gossip protocols and demonstrate that our social gossip mechanism has optimal convergence time in terms of protocol rounds, while incurring polynomial communication and linear computational complexity.

Organization.

The rest of the paper is organized as follows: Section 2 provides formal definitions. Section 3 describes our gossip based path–verification protocol followed by a performance analysis on fully connected communication networks in Section 4, and on social networks in Section 5. Last, in Section 6 we review related work.

2. SYSTEM MODEL AND PRELIMINARIES

In our analysis of the suggested gossip mechanism we view a social network as a set V of n nodes that interact with random uniformly chosen contacts. In Section 5, we expand our discussion to partial graphs in which users are limited to interact with pre-designated sets of neighbors.

We envision a system in which different recommendations are introduced at different times and locations continuously, and focus our discussion on a single recommendation, denoted throughout the paper by m. Spammers are treated as *corrupt* nodes, and we assume that the set F of spammers of any specific message is bounded by size f. Initially, we assume that a random set of non–corrupt nodes originate the message m and thus are considered *active* relative to m. We denote by $I \subseteq V \setminus F$ the initial subset of active nodes that originate the message, and stipulate that $|I| > f$.

The active nodes are unknown *a priori*, nor are they known to the nodes themselves at the outset of the protocol, or even during the protocol's execution. The remaining non–corrupt nodes are considered *passive*. Passive nodes store and relay messages. Each time the message is relayed, its relevance diminishes. Consequently, we set a bound ℓ on the number of hops that a gossip message may traverse from any origin.

The goal of the algorithm is to turn passive nodes into active ones. A passive node becomes active for message m once it received m through $f+1$ vertex–disjoint paths of length shorter than ℓ. In order to implement our adoption criterion, a path-verification mechanism needs to be incorporated. A passive node forwards the message even before adopting it. Our gossip protocol communicates not only the messages, but also the paths by which the message has traversed through the network. Each node needs to collect sets of disjoint paths of bounded lengths by which the message traversed through the network. When the set contains $f+1$ disjoint paths, the message is adopted by the node.

Transmission Model. We model the communication as a uniform random epidemic process, similar to Demers *et al.* [7]. These protocols operate in rounds, denoted $r = 1, 2, \ldots$. In each round r, each node chooses uniformly at random another node to be a communication partner, and exchanges information with it. With reference to the flow of information, [7] distinguishes between push and pull transmission models. Assume node v calls node u.

- The message is *pushed* if v transfers u a message.
- The message is *pulled* if u transfers v a message.

In the protocols suggested by [27], nodes pull messages, rather than push messages to the system, in order to limit the ability of corrupt nodes to inject false information into the system. Our work adopts this approach by choosing the pull transmission model for the protocols discussed in this paper. Last, we assume that the communication channels among the nodes are reliable and authenticated, in the sense that a non-corrupt node u receives a message over a communication channel from another non-corrupt node v if and only if v sent that message to u. This is a realistic constraint in a large and wide-spread social network such as our example of an instant-messaging network.

3. PROTOCOL

In this section, we present a practical, bounded-hop path–verification protocol based on a gossip pull transmission model, whose time complexity is asymptotically optimal.

Our protocol accumulates at each node v a set of simple paths. Let $P_v(r)$ denote the set of simple paths through which a passive node v receives the message at the beginning of round r. The protocol also grows at each node a separate set of **disjoint** simple paths $\overline{P}_v(r)$. Initially, both sets are empty $P_v(0) = \overline{P}_v(0) = \emptyset$ for every passive node.

Aggregate Phase. In the first $O(\log n)$ rounds, each passive node u chooses a communication partner v uniformly at random and sends a pull request. If v obtained m through path(s) of length smaller than ℓ, v sends m to u along with the set of paths it stores for m, $P_v(r)$. When node u receives a response, it verifies that $P_v(r)$ includes paths no longer than $\ell - 1$ and that $|P_v(r)| \leq 2^r$. If so, let $P_v^+(r)$ be the result of appending v to each path in set $P_v(r)$, u sets $P_u(r+1) = P_v^+(r) \cup P_u(r)$. Otherwise u discards the response.

Collect Phase. In each of the following rounds, each passive node u chooses a communication partner v uniformly at random and sends a pull request. Along with each pull request, u includes the set $\overline{P}_u(r)$. When v receives the request it acts as follows: If a simple path P exists in $P_v(r)$ of length smaller than ℓ such that P does not contain any node that already appears in $\overline{P}_u(r)$, v sends P to u. Otherwise, v sends the shortest path from the set $\overline{P}_v(r)$. When node u receives the response from v denoted by P, u first verifies that its length is smaller than ℓ. Next, u appends v to the path P and then verifies that P is indeed vertex–disjoint to $\overline{P}_u(r)$. If so, u updates $\overline{P}_u(r+1) = \overline{P}_u(r) \cup P$. Otherwise, u replaces P with an existing path Q if and only if P's length is shorter than Q *i.e.* $\overline{P}_u(r+1) = (\overline{P}_u(r) \setminus Q) \cup P$.

Termination. A passive node v that obtained m becomes active for a message m only if it received m through $f+1$ vertex–disjoint paths, namely $|\overline{P}_v(r)| > f$.

4. ANALYSIS

As has been observed in previous works that employ path–verification [27, 24], a message m that is introduced by f or fewer nodes will not be adopted by any non-corrupt node. The reason for this is that a node that pulls m from a sender appends the sender's identity to the path(s). Hence, corrupt nodes cannot remove their own identities from the gossip paths of a message, but only the identity of others. Consequently, if there are f or less origins for the message, there may not be $f+1$ disjoint paths for m at any point. Consequently, any rumor injected by the corrupt nodes in F will not be adopted by any non-corrupt node.

The main focus of our analysis in the rest of the paper is to determine the number of rounds needed to spread a good recommendation m, which is originated at I, to all non-corrupt nodes. Additionally, we comment on the storage and computation load on passive nodes.

Generally, the number of rounds needed to turn all passive nodes active depends on the network topology G. For the moment, we consider a fully–connected communication network (*i.e.* a complete graph). We extend our analysis to a social network topology in Section5.

Aggregate Phase. During the aggregate phase, the dissemination of the same message m from different origins occurs independently of one another, since the entire paths information is accumulated and forwarded in every gossip interaction. Therefore, we can track the dissemination of m from each individual origin separately. We do this by analyzing a randomized gossip protocol, denoted $A(\ell)$, of a single message. In order to avoid confusion, we denote the individual copy of m that $A(\ell)$ spreads by $\hat{m}$. The only distinction of $A(\ell)$ of $\hat{m}$ from standard information spread randomized gossip is that the message $\hat{m}$ propagates from its origins up to a bounded hop-distance ℓ.

More specifically, in each round, each node pulls information from a communication partner chosen uniformly at random from its neighbors along with paths by which the message traversed through the network. Upon request, a node forwards the message $\hat{m}$ only if it received $\hat{m}$ through a path of length smaller than ℓ, otherwise, it ignores the request.

Let $X(r)$ denote the set of nodes that obtained the message $\hat{m}$ before the beginning of round $r = 0, 1, 2, \ldots$. Initially, $X(0) = \{v\}$, where v is the origin node. Let ε denote the desired bounded error probability of $A(\ell)$ (e.g. w.h.p. means that $\varepsilon \leq n^{-c}$ for an arbitrary constant $c > 0$), hence, the time complexity of $A(\ell)$ is as follows:

$$T_{A(\ell)}(\varepsilon) = \inf_{r=0,1,2,\ldots} \{r : Pr(X(r) \neq V) \leq \varepsilon\}$$

Let $X_d(r) \subseteq X(r)$ denote the subset of nodes that obtained $\hat{m}$ through a shortest path of length $0 \leq d \leq \ell$, and denote by $x_d(r) = |X_d(r)|$ its size. The algorithm terminates when all the nodes received the message $\hat{m}$. Our first lemma shows that the hop bound ℓ substantially impacts the speed of message spreading.

Lemma 4.1. *In any execution of algorithm $A(\ell)$, and for all rounds r, $|X(r)| \leq 2r^{\ell}$.*

Proof. At the beginning of round $r+1$, $x_d(r+1)$ counts those nodes that remain in possession of shortest path of length d from the previous round, as well as new nodes that have obtained $\hat{m}$ through nodes at distance $d-1$. Clearly then, $x_d(r+1) \leq x_d(r) + x_{d-1}(r)$. From this, $x_d(r+1) \leq \sum_{i=0..r} x_{d-1}(i) \leq (r+1)x_{d-1}(r)$. Since $x_0(\cdot) = 1$, obtain inductively $x_d(r+1) \leq (r+1)^d$. It follows that $|X(r)| = \sum_{d=0..\ell} x_d(r) \leq \sum_{d=0..\ell} r^d \leq 2r^{\ell}$ as required. □

In particular, logarithmic termination time requires that for round $r = c\log(n)$, where c is a constant, $\hat{m}$ has reached all nodes at round r. This implies $c\log^{\ell}(n) = n$, which requires $\ell \geq \log(n)/\log\log(n)$.

An upper bound on the number of rounds required for termination with any choice of $\ell = O(\log(n)/\log\log(n))$ is given in the following theorem.

Theorem 1.1. *The number of rounds for termination of $A(\ell)$, where $\ell \leq \frac{2(\log(n\log(n))}{\log\log(n)}$, is $O((n\log(n))^{(2/\ell)})$ in expectation.*

Proof. For every node $v \in I$ and for every $d = 0, 1, \ldots, \ell$ and round r let $B_d(r)$ denote $V \setminus \cup_{i=0}^{d} X_i(r)$, and by $b_d(r)$ the ratio $\frac{|B_d(r)|}{n}$. In particular, for every round $r = 0, 1, \ldots$ $b_0(r) = 1 - \frac{1}{n}$, hence, the algorithm $A(\ell)$ terminates once $b_\ell(r) \leq \frac{1}{n}$. Assume that $r \geq \ell$, otherwise, for every round in which $\ell > r$ satisfies $b_\ell(r) = b_{\ell-1}(r)$.

$$\begin{aligned}
E[b_d(r+1)|x_0(r), x_1(r), \ldots, x_\ell(r)] &= \\
\frac{n - \sum_{i=0}^{d} x_i(r)}{n} \cdot \frac{n - \sum_{i=0}^{d-1} x_i(r)}{n} &= \\
b_d(r) \cdot b_{d-1}(r) &= \\
\Pi_{i=d-1}^{r} b_{d-1}(i) &= \\
\Pi_{i_{d-1}=d-1}^{r} \Pi_{i_{d-2}=d-2}^{i_{d-1}-1} \cdots \Pi_{i_1=1}^{i_2-1} (1 - \frac{1}{n})^{i_1} &= \\
(1 - \frac{1}{n})^{\sum_{i_{d-1}=d-1}^{r} \sum_{i_{d-2}=d-2}^{i_{d-1}-1} \cdots \sum_{i_1=1}^{i_2-1} i_1} &= \\
e^{-\frac{1}{n} \sum_{i_{d-1}=d-1}^{r} \sum_{i_{d-2}=d-2}^{i_{d-1}-1} \cdots \sum_{i_1=1}^{i_2-1} i_1} &\leq \\
e^{-\frac{r^{\frac{d}{2}}}{n}} \ \forall r = \omega(d)
\end{aligned}$$

For every $d = 0, 1, 2, \ldots$ and for every round $r \geq d$ satisfies $E[b_{d+1}(r)] \leq E[b_d(r)]$. Setting $\ell = r = O(\log(n))$ essentially poses no hop count constraint, which is equivalent to a pull based information spread algorithm, thus $e^{-r} \leq E[b_d(r)] \leq e^{-\frac{r^{\frac{d}{2}}}{n}}$ Applying the above bound, we obtain that when $\ell \leq \frac{2(\log(n\log(n))}{\log\log(n)}$ and round $r = (n\log(n))^{(2/\ell)}$ is reached, the expected number of uninfected nodes drops below 1 and the protocol terminates. □

Collect Phase. The key ingredient to our analysis of the collection phase is captured in the following intuitive lemma. It tracks the progress of the gossip algorithm $A(\ell)$ within two separate sets of nodes, 'red' and 'blue', denoted by $\mathcal{R}$ and $V \setminus \mathcal{R}$ respectively. The colors are unknown a priori, nor are they known to the nodes themselves during the protocol's execution. Let $C(\ell, \mathcal{R})$ denote an execution of the bounded hop gossip algorithm $A(\ell)$, during which any attempted interaction between the nodes in $V \setminus \mathcal{R}$ and in $\mathcal{R}$ is silently dropped. That is, $A(\ell)$ diffuses a message initiated at a blue node only through 'blue' nodes to all 'blue' nodes. Let us denote by q the ratio $\frac{|\mathcal{R}|}{n}$. The lemma shows that there is constant slow down factor as long as $q < \frac{1}{2}$.

Lemma 4.2. *The number of rounds for termination of $C(\ell, \mathcal{R})$, where $\ell = \Theta(\frac{\log(n)}{\log\log(n)})$ and $q < \frac{1}{2}$, is $O(\log(n))$ with high probability.*

Proof. Let $\hat{X}_d(r) \subseteq V \setminus \mathcal{R}$ denote the subset of blue nodes that obtained $\hat{m}$ through a shortest path of length $0 \leq d \leq \ell$, and denote by $\hat{x}_d(r) = |\hat{X}_d(r)|$ its size. For every $d = 0, 1, 2, \ldots, \ell$ and round r let $\hat{B}_d(r) = V \setminus (\mathcal{R} \bigcup \cup_{i=0}^{d} \hat{X}_i(r))$, and the ratio $\hat{b}_d(r)$ by $\frac{|\hat{B}_d(r)|}{n-|\mathcal{R}|}$. Hence, the execution $C(\ell, \mathcal{R})$ terminates once $\hat{b}_\ell(r) \leq \frac{1}{n-|\mathcal{R}|}$.

$$E[\hat{b}_d(r+1)|\hat{b}_d(r), \hat{b}_{d-1}(r)] \quad = \quad \hat{b}_d(r)(q + (1-q)\hat{b}_{d-1}(r))$$

Given $\hat{b}_i(r)$, $i = 0, 1, \ldots, \ell$, the expected reduction in size for $\hat{b}_d(r+\Delta)$ is at least by factor

$$\rho(\Delta, \hat{b}_{d-1}(r)) = [q + (1-q)\hat{b}_{d-1}(r)]^{\Delta}$$

By studying the function $\rho(\Delta, (1-\epsilon))$, when $\epsilon < \frac{1}{2}$ and likewise, $q < \frac{1}{2}$, it is easy to see that for all $\Delta \geq 3$, $\rho(\Delta, (1-$

$\epsilon)) < (1-\epsilon)$ Therefore, for all rounds r s.t. $\hat{b}_{\ell-1}(r) > \frac{1}{2}$, we obtain the following recursion formula:

$$E[\hat{b}_d(r+3)|\hat{b}_d(r), \hat{b}_{d-1}(r)] \leq \hat{b}_d(r) \cdot \hat{b}_{d-1}(r)$$

Applying a Chernoff bound yields

$$n \cdot \hat{b}_d(r+3) \leq (1 + \frac{1}{\log(n)})(n \cdot \hat{b}_d(r))\hat{b}_{d-1}(r)$$

w.h.p., provided $\hat{b}_{\ell-1}(r) > \frac{1}{2}$. The recursion formula is similar to theorem1.1 with 3-round "steps" instead of single round step. The analysis in theorem1.1 solves the recursion. In the final phase, where $\hat{b}_{\ell-1}(r) < \frac{1}{2}$, the expected reduction in size for $\hat{b}_\ell(r)$ is bounded as follows.

$$E[\hat{b}_\ell(r+1)|\hat{b}_\ell(r)] \quad \leq \quad \frac{3}{4}\hat{b}_\ell(r)$$

Thus, within additional $O(\log_{\frac{4}{3}}(n))$ rounds the execution terminates w.h.p. □

As before, let $B(\ell, f)$ be our path verification gossip protocol, where ℓ is the path length bound, and by f, the adoption threshold. Our main complexity result is captured in the following theorem:

Theorem 1.2. *The number of rounds for termination of $B(\ell, f)$ in a network where $f(1+\ell) < \frac{n}{2}$ and $\ell = \Theta(\frac{\log(n)}{\log\log(n)})$ is $O(\log n + f)$ with high probability.*

Proof. Consider a passive node u after $\Theta(\log(n))$ rounds. Suppose that u obtained the message m through a set of vertex–disjoint paths, $\overline{P}_u(r)$. We show that any node $v \notin (\overline{P}_u(r) \cup F)$ obtained m through a path P such that $(\overline{P}_u(r) \cup F) \cap P = \emptyset$ w.h.p. Consider the nodes in the set $(\overline{P}_u(r) \cup F)$ 'red' nodes, and note that $|(\overline{P}_u(r) \cup F)| \leq f(\ell+1)$. That is, the number of 'blue' nodes is at least half of the network, namely $f(1+\ell) < \frac{n}{2}$. For each origin 'blue' node $w \in I$ of the message m, we may apply Theorem 1.1 and conclude that v has obtained a path P from w excluding any 'red' nodes within $O(\log(n))$ rounds w.h.p. Accordingly, within additional $O(f+\log(n))$ rounds, node u collects $f+1$ vertex–disjoint paths for m w.h.p. and becomes an active node for m. □

Storage Complexity. Next, we address the storage overhead incurred by each of the passive nodes in the system. Each passive node aggregates path information for a message m during the first $\Theta(\log n)$ rounds, where maximum-path-allowed is at most $O(\frac{\log(n)}{\log\log(n)})$. Moreover, a passive node stores at most 2^r paths during round r. Consequently, the number of paths stored by each passive node is polynomial in n.

Computational Complexity. In previous work [27, 24, 25] each node had a significant computational step in order to find a satisfying vertex–disjoint path set among the paths held by a passive node. As shown in [27], this problem is NP-complete. Fortunately, using randomization, our protocol's computational complexity is polynomial in n. More specifically, in each round $r = \Theta(\log n)$, each passive node u pulls information from a communication partner v at random. Along with each pull, u includes the set $\overline{P}_u(r)$. When v receives the request it needs to look for a path $P \in P_v(r)$ such that $\overline{P}_u(r) \cap P = \emptyset$. Obviously, such a search is linear in the number of nodes in $P_v(r)$, which is polynomial in n.

For completeness, we show that the protocol terminates with optimal asymptotic running time for the case where $f = O(n)$ (*i.e.* since $|I| > f$ requires that $f < \frac{n}{2}$) w.h.p.

Claim 4.3. *The protocol terminates within $O(n)$ rounds, where $f = O(n)$ (*i.e. $f < \frac{n}{2}$) w.h.p.*

Proof. In this special case we consider a path–verification where $\ell = 1$. Hence, since each active node, $v \in I$, keeps sending the message for n rounds the probability that a passive node obtained the message originated at any of the $|I|$ originators is exactly $1 - \left(1 - \frac{1}{n}\right)^{\Omega(n)} \geq \Omega(1 - \frac{1}{e})$. Henceforth, within additional $O(f)$ rounds all the passive nodes become active w.h.p. with minimal message payload. □

5. SOCIAL NETWORKS

A real social network differs from the setting of the analysis above in two important ways. First, nodes communicate directly only with their social partners, and not with a partner selected uniformly at random, as in randomized gossip. Second, the acceptance criterion employed by users' agents may be more complicated than our fixed, global f threshold. In this section, we refer to a reduction from the analysis above to a more realistic setting.

We address the acceptance criteria first. Information cascades in social settings have been studied extensively in sociology. Information cascades are phenomena in which individuals are influenced by others to adopt a new action or idea. In information cascade the goal is usually to determine which initial set will influence the network the most. This is of interest to marketing, epidemiology, and computer networking. For example, the general threshold model used in sociology and economics, whose generalization was introduced in [20], can be used for quantifying the "cumulative influence" of a node set S over a node v. In the model, each node v has a monotonous activation function $f_v : 2^V \mapsto [0,1]$, and a threshold θ_v from the interval $(0,1]$. A node v becomes active at the beginning of round $r+1$ if $f_v(S) \geq \theta_v$, where S is the set of nodes active at the beginning of round r.

In our approach, f_v is mapped to the following criterion. First, we assign a step-function $W : V^* \mapsto [0,1]$ that gives relevance value to gossip paths. A path P has relevance 1 if its length is up to our hop-count limit ℓ, and 0 otherwise. Second, we evaluate the function f_v at S as follows. A node v becomes active at round $r+1$ if $f_v(S) \geq 1$, where $S \supseteq I$ is the set of nodes active at round r and v has obtained the message from the nodes in S via a set of $f+1$ disjoint paths whose relevance is 1. It would be interesting to study a more general application of the threshold model. In particular, the relevance value is application specific data that indicates the importance of the data from the local user's perspective and may change each time the data is shared. For example, as a recommendation propagates away from its creator, the data relevance value may decrease. These extensions are left for future studies.

Second, we address the issue of connectivity. Formally, we investigate the time complexity of an information spread gossip algorithm A in which a node v initiates a communication with one of its immediate neighbors $u : v \neq u$ with probability P_{vu}. With probability P_{vv}, it does not contact another node. The $n \times n$ matrix $P = [P_{uv}]$ characterizes the information spread gossip algorithm A; each matrix P gives rise to a different algorithm A_P. We assume that P is stochastic, and that $P_{vu} = 0$ if $(v, u) \notin E$, as nonadjacent nodes cannot communicate with each other.

For a lack of a better model, we assume that a social network is a small world topology [36, 21]. In [13] Flaxman studies the expansion properties of perturbed random graphs and conjectures that small world topologies tend to be fast–mixing time. More formally, the mixing time T of a graph $G = (V, E)$ of n nodes quantifies how fast the ending point of a random walk approaches the stationary distribution of the graph G. If $T = O(\log(n))$, the graph is called fast–mixing. Furthermore, Boyd *et al.* show in [5] that the time complexity of an information spread gossip algorithm in an arbitrary graph is closely related to the mixing time of the random walk defined by the stochastic matrix that characterizes the algorithm. This connection was later confirmed by [28] Let $T_{mix}(P)$ be the mixing time of a simple random walk defined by the stochastic matrix P, which characterizes a gossip algorithm denoted by A_P. The termination time of the algorithm A_P is bounded by $O(\log(n) + T_{mix}(P))$.

Our analysis of the protocol presented in Section3 derives the time complexity from the diffusion time of an information spread gossip protocol. As before, the protocol activates only passive nodes that received the message through $f + 1$ disjoint paths. As a result, we require that the network be at least $2f + 1$-connected, since there are at most f corrupt nodes we require that the induced graph on the non-corrupt nodes $G[V \setminus F]$ be $f + 1$-connected. Furthermore, by setting $\ell = \Theta(\frac{\log(n)}{\log\log(n)})$, the time complexity of the protocol running over a small world network topology is $O(\log n + f)$. This is true since the protocol's time complexity was derived from an information spread gossip algorithm $A(\ell)_P$, which, as shown above, exhibits an asymptotically optimal time complexity $O(\log n)$.

6. RELATED WORK

Gossip-style techniques are found in numerous robust distributed systems for information dissemination, including Usenet news [22], the Grapevine distributed system [4], Ad Hoc routing [17], distributed failure detectors [32], the Astrolabe network management system [31], lightweight broadcast [9], membership maintenance [15], the CYCLON system [33], GosSkip [16], collaborative content distribution [12], and others. The performance of randomized gossip has been addressed in several seminal papers, including[7, 19].

Our protocol borrows the path–verification technique from the realm of Byzantine gossip. The problem of secure information dissemination in a fully Byzantine environment without the use of conditional cryptography (Cryptographic schemes, for which we do not have a mathematical proof that they are totaly secure) was introduced in [23] and further explored in [25, 24, 27]. The works closest in spirit to ours in the use of epidemic-style propagation are [25, 24]. By bounding the size of paths to $O(\frac{\log n}{\log\log n})$ we are able to completely circumvent the lower bound of [23] and obtain asymptotically optimal dissemination time, albeit at the cost of increased storage size. With reference to computational complexity, in previous work [25, 24, 27], each node had a significant computational step in order to find a satisfying vertex–disjoint path set among the paths held by a passive node. As shown in [27], this problem is NP-complete. Our work adds on to previous studies with the introduction of a new gossip-based algorithm for path–verification that spreads a message among non–corrupt nodes in $O(\log n + f)$ rounds while incurring polynomial computational complexity.

On a somewhat higher level, the vision underlying social gossip employs the concepts of trust and reputation, in that it leverages on existing trust relationships for high credential recommendations. Trust and reputation systems initially attracted attention as traditional quality assurance mechanisms were not as applicable for fostering cooperation in online trading communities such as `www.ebay.com`, `www.amazone.com`. Online reputation values are computed by systems that aggregate feedback provided by online community members. Most existing examples of reputation systems are centralized in nature. However, in many cases [10, 6, 11, 2, 18, 29, 30] it is desirable to perform calculations in a distributed manner, especially in the presence of some potentially misbehaving system elements that cannot be trusted to perform computations as expected.

The scale and decentralization envisioned in peer-to-peer setting call into question the credibility of nodes, and raise a concern that these schemes could be misused to spread corrupt information. For example, the Sybil attack is an attack in which a single peer controls multiple identities to other legitimate peers in the networks, which threatens decentralized distributed systems that have no central, trusted authority to vouch for a peer-to-peer correspondence between users and identities. By controlling a large fraction of the nodes in the system, the malicious user is able to ŞoutvoteŤ the honest users in such collaborative tasks as Byzantine failure defenses. The problem of Sybil attack was initiated in [8] and further explored in [3, 14]. The works closest in spirit to ours with regards to social network are SybilGuard [37] and LOCKSS [26]. These protocols are based on the Şsocial networkŤ connecting user identities, where an edge between two identities indicates a human-established trust relationship. Malicious users can create many identities but few trust relationships. Thus, there is a disproportionately-small ŞcutŤ in the graph between the Sybil nodes and the honest nodes, which obviously bounds the number of vertex-disjoint paths. In essence, the trust relation among social network users helps achieve much stronger properties compared to networks with no trust relation between their users.

Recent studies indicate that many peer-to-peer file-sharing activities involve corrupt and polluted files. Credence [35, 34] addresses content pollution by providing methods to evaluate file authenticity. In order to validate a file, a client issues a vote query to collect votes on the file; peers respond with matching cryptographically signed votes they possess, if any. Next, votes are collected and authenticated, where the client uses statistical correlation to weight the relevance of each vote it obtained. In order to compute the statistical correlations with its peers, each client utilizes either shared voting history or correlation values (*i.e.* transitive correlation) or both. Conceptually, each client keeps track of both signed votes and correlations it has encountered using two separate databases.

Compared with our approach, Credence protocol is also based on gossip communication mechanism, where network nodes store and relay recommendations/votes regardless of their content. However, in Credence, recommendations are flooded through the entire network, whereas in our approach recommendations traverse only through a limited-hop bound from their sources. By employing our prudent reinforcement mechanism, we in fact significantly reduce the impact of malicious nodes on the network traffic. Our protocol's local approach allows for an efficient information storage each node stores message's paths of bounded length. Finally, while Credence makes use of central certificate authority, our reinforcement mechanism does not require any conditional cryptography and as such is entirely decentralized.

7. REFERENCES

[1] *Nocturnal.* http://research.microsoft.com/research/sv/Nocturnal/.

[2] Z.ABRAMS and R.MCGREW. Keeping peers honest in eigentrust. In *P2PECON '04: Proceeding of the 2004 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 102–111, New York, NY, USA, 2004. ACM Press.

[3] R.A. Bazzi and G.Konjevod. On the establishment of distinct identities in overlay networks. In *PODC '05: Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, pages 312–320, New York, NY, USA, 2005. ACM Press.

[4] A.D. Birrell, R.Levin, R.M. Needham, and M.D. Schroeder. Grapevine, an exercise in distributed computing. *Communications of the ACM*, 25(4):260–274, 1982.

[5] S.Boyd, A.Ghosh, B.Prabhakar, and D.Shah. Gossip algorithms: Design, analysis and applications. In *Proceedings of IEEE INFOCOM*, 2005.

[6] Alice Cheng and Eric Friedman. Sybilproof reputation mechanisms. In *P2PECON '05: Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 128–132, New York, NY, USA, 2005. ACM Press.

[7] A.Demers, D.Greene, C.Hauser, W.Irish, J.Larson, S.Shenkcr, H.Sturgis, D.Swinehart, and D.Terry. Epidemic algorithms for replicated database maintenance. In *PODC '87: Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, pages 1–12, New York, NY, USA, 1987. ACM Press.

[8] J.R. Douceur. The sybil attack. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 251–260, London, UK, 2002. Springer-Verlag.

[9] P.Th. Eugster, R.Guerraoui, S.B. Handurukande, P.Kouznetsov, and A.-M. Kermarrec. Lightweight probabilistic broadcast. *ACM Trans. Comput. Syst.*, 21(4):341–374, 2003.

[10] Michal Feldman and John Chuang. Overcoming free-riding behavior in peer-to-peer systems. *SIGecom Exch.*, 5(4):41–50, 2005.

[11] Michal Feldman, Kevin Lai, Ion Stoica, and John Chuang. Robust incentive techniques for peer-to-peer networks. In *P2PECON '04: Proceeding of the 2004 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 102–111, New York, NY, USA, 2004. ACM Press.

[12] Y.Fernandess and D.Malkhi. On collaborative content distribution using multi-message gossip. In *IPDPS: IEEE International Parallel and Distributed Processing Symposium.* IEEE, April 2006.

[13] A.D. Flaxman. Expansion and lack thereof in randomly perturbed graphs. Technical report, 2006. Manuscript under submission.

[14] M.F.Kaashoek G.Danezis, C. Lesniewski-Laas and R.Anderson. Sybil-resistant dht routing. In *In European Symposium On Research In Computer Security*, pages 305–318, September 2005.

[15] A.J. Ganesh, A.Kermarrec, and L.Massoulié. Peer-to-peer membership management for gossip-based protocols. *IEEE Transactions on Computers*, 52(2), February 2003.

[16] R.Guerraoui, S.B. Handurukande, and A.-M. Kermarrec. GosSkip: a Gossip-based Structured Overlay Network for Efficient Content-based Filtering. Technical report, 2004.

[17] Z.Haas, J.Y. Halpern, and L.Li. Gossip-based ad hoc routing. In *Proceedings of IEEE INFOCOM*, June 2002.

[18] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 640–651, New York, NY, USA, 2003. ACM Press.

[19] R. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking. Randomized rumor spreading. In *FOCS '00: Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, page 565, Washington, DC, USA, 2000.

[20] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, New York, NY, USA, 2003. ACM Press.

[21] J. Kleinberg. The Small-World Phenomenon: An Algorithmic Perspective. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, 2000.

[22] K. Lidl, J. Osborne, and J. Malcome. Drinking from the firehose: Multicast usenet news. In *Proceedings of the Usenix Winter Conference*, pages 33–45, January 1994.

[23] D. Malkhi, Y. Mansour, and M. K. Reiter. On diffusing updates in a byzantine environment. In *Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems*, page 134, Washington, DC, USA, 1999. IEEE Computer Society.

[24] D. Malkhi, E. Pavlov, and Y. Sella. Optimal unconditional information diffusion. In *Proceedings of the 15th International Symposium on DIStributed Computing*, 2001.

[25] D. Malkhi, M. K. Reiter, O. Rodeh, and Y. Sella. Efficient update diffusion in byzantine environments. In *Proceedings of the 20th IEEE Symposium on Reliable Distributed Systems*, Washington, DC, USA, 2001. IEEE Computer Society.

[26] Petros Maniatis, Mema Roussopoulos, T.J. Giuli, David S. H. Rosenthal, and Mary Baker. The lockss peer-to-peer digital preservation system. *ACM Trans.*

Comput. Syst., 23(1):2–50, 2005.

[27] Y. M. Minsky and F. B. Schneider. Tolerating malicious gossip. *Distributed Computing*, 16(1):49–68, 2003.

[28] D.Mosk-Aoyama and D. Shah. Computing separable functions via gossip. In *PODC '06: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 113–122, New York, NY, USA, 2006. ACM Press.

[29] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation for e-businesses. In *HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 7*, page 188, Washington, DC, USA, 2002. IEEE Computer Society.

[30] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Commun. ACM*, 43(12):45–48, 2000.

[31] R.van Renesse. Scalable and secure resource location. In *Proceedings of IEEE Hawaii International Conference on System Sciences*, January 2000.

[32] R. van Renesse, Y. Minsky, and M. Hayden. A gossip-style failure detection service. In *Proceedings of Middleware*, 1998.

[33] S. Voulgaris, D. Gavidia, and M. van Steen. Cyclon: Inexpensive membership management for unstructured p2p overlays. *J. Network Syst. Manage.*, 13(2), 2005.

[34] K. Walsh and E. Sirer. Fighting peer-to-peer spam and decoys with object reputation. In *P2PECON '05: Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 138–143, New York, NY, USA, 2005. ACM.

[35] K. Walsh and E. Sirer. Experience with an object reputation system for peer-to-peer filesharing. In *NSDI'06: Proceedings of the 3rd conference on 3rd Symposium on Networked Systems Design & Implementation*, pages 1–1, Berkeley, CA, USA, 2006. USENIX Association.

[36] D. J. Watts and S. H. Strogatz. Collective dynamics of Şsmall-worldŤ networks. *Nature*, pages 440Ű–442, 1998.

[37] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. In *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 267–278, New York, NY, USA, 2006. ACM Press.