Random Gossip Processes in Smartphone Peer-to-Peer Networks

Newport, Calvin
Georgetown University
cnewport@cs.georgetown.edu

Weaver, Alex
Georgetown University
aweaver@cs.georgetown.edu

***Abstract*—In this paper, we study random gossip processes in communication models that describe the peer-to-peer networking functionality included in standard smartphone operating systems. These processes are well-understood in standard peer-to-peer network models, but little is known about their behavior in models that abstract the smartphone peer-to-peer setting. With this in mind, we begin by studying a simple random gossip process in the synchronous mobile telephone model (the most common abstraction used to study smartphone peer-to-peer systems). We prove that the simple process is actually more efficient than the best-known gossip algorithm in the mobile telephone model, which required complicated coordination among the nodes in the network. We then introduce a novel variation of the mobile telephone model that removes the synchronized round assumption, shrinking the gap between theory and practice. We prove that simple random gossip processes still converge in this setting and that information spreading still improves along with graph connectivity.**

***Index Terms*—gossip, distributed algorithms, peer-to-peer networks**

I. Introduction

In this paper, we study random gossip processes in smartphone peer-to-peer networks. We prove the best-known gossip bound in the standard synchronous model used to describe this setting, and then establish new results in a novel asynchronous variation of this model that more directly matches the real world behavior of smartphone networks. Our results imply that simple information spreading strategies work surprisingly well in this complicated but increasingly relevant environment.

In more detail, a random gossip process is a classical strategy for spreading messages through a peer-to-peer network. It has the communicating nodes randomly select connection partners from their eligible neighbors, and then once connected exchange useful information.[1] As elaborated in Section II, these random processes are well-studied in standard peer-to-peer models where they have been shown to spread information efficiently despite their simplicity.

To date, however, little is known about these processes in the emerging setting of *smartphone* peer-to-peer networks, in which nearby smartphone devices connect with direct radio links that do not require WiFi or cellular infrastructure. As also elaborated in Section II, both Android and iOS now provide support for these direct peer-to-peer connections, enabling the possibility of smartphone apps that generate large peer-to-peer networks that can be deployed, for example, when infrastructure is unavailable (i.e., due to a disaster) or censored (i.e., due to government repression). This paper investigates whether the random gossip processes that have been shown to spread information well in other peer-to-peer settings will prove similarly useful in this intriguing new context.

A. The Mobile Telephone Model (MTM)

The *mobile telephone model* (MTM), introduced by Ghaffari and Newport [1], extends the well-studied *telephone model* of wired peer-to-peer networks (e.g., [2]–[10]) to better capture the dynamics of standard smartphone peer-to-peer libraries. In recent years, several important peer-to-peer problems have been studied in the MTM, including rumor spreading [1], load balancing [11], leader election [12], and gossip [13].

As we elaborate in Section III-A, the mobile telephone model describes a peer-to-peer network topology with an undirected graph, where the nodes correspond to the wireless devices, and an edge between two nodes indicates the corresponding devices are close enough to enable a direct device-to-device link. Time proceeds in synchronous rounds. At the beginning of each round, each node can *advertise* a bounded amount of information to its neighbors in the topology. At this point, each node can then decide to either send a connection *invitation* to a neighbor, or instead receive these invitations, choosing at most one incoming invitation to accept, forming a connection. Once connected, a pair of node can perform a bounded amount of communication before the round ends.

B. Gossip in the MTM

The gossip problem assumes that k out of the $n \geq k$ nodes start with a gossip message. The problem is solved once all nodes have learned all k messages. In the context of the MTM, we typically assume that at most $O(1)$ gossip messages can be transferred over a connection in a single round, and that advertisements are bounded to $O(\log n)$ bits.

A natural random gossip process in this setting is to have nodes advertise a hash of their current token set and only attempt random connections with nodes advertising different hashes. It is straightforward to establish that this process solves gossip in $O(nk)$ rounds, with high probability in n (see [13] for the details of this analysis).

In our previous work [13], we improved on this bound by introducing a more complicated gossip algorithm called

[1]The main place where different random gossip processes vary is in their definition of "eligible." What unites them is the same underlying approach of random connections to nearby nodes.

crowded bin, which requires, among other feats of distributed coordination, for nodes to run a subroutine that estimates k, and then uses this value to build a TDMA schedule that enables k independent spreading processes to run in parallel. We prove that crowded bin solves gossip in $O((k/\alpha)\log^5 n)$ rounds, with high probability in n, when run in a network topology with vertex expansion α (see Section III-C). For all but the smallest α values (i.e., least amounts of connectivity) this result improves the $O(nk)$ bound for the simple process.

A key open question from [13] is whether or not it is possible to close the time complexity gap between appealingly simple random gossip processes and the more complicated machinations of crowded bin. As we detail next, this is a core question tackled in this paper.

C. New Result #1: Improved Analysis for Gossip in the MTM

In Section III-A, we consider a simple random gossip process that we call *random spread*. By introducing a new analysis technique, we significantly improve on the straightforward $O(nk)$ bound from [13] for processes of this type. Indeed, we prove random spread is actually slightly *more* efficient than the much more complicated crowded bin algorithm from [13], showing that with high probability in n, random spread requires only $O((k/\alpha)\log^4 n)$ rounds to solve gossip.

The primary advantage of random spread gossip is its simplicity. A secondary advantage is that this algorithm works in the *ongoing* communication scenario in which new rumors keep arriving in the system. Starting from any point in an execution, if there are k rumors that are not yet fully disseminated, they will reach all nodes in at most an additional $O((k/\alpha)\log^4 n)$ rounds, regardless of how many rumors have been previously spread. The solution in [13], by contrast, must be restarted for each collection of k rumors, and includes no mechanism for devices to discover that gossip has completed for the current collection. Accordingly, this new result fully supersedes the best-known existing results for gossip in the MTM under similar assumptions.[2]

At the core of our analysis is a new data structure we call a *size band table* that tracks the progress of the spreading rumors. We use this table to support an amortized analysis of spreading that proves that the stages in which rumors spread slowly are balanced out sufficiently by stages in which they spread quickly, providing a reasonable average rate.

D. New Result #2: Gossiping in the Asynchronous MTM

The mobile telephone model is a high-level abstraction that captures the core dynamics of smartphone peer-to-peer communication, but it does not exactly match the behavior of real smartphone networking libraries. The core difference between theory and practice in this context is synchronization. To support deep analysis, this abstract model (like many models used to study distributed graph algorithms) synchronizes devices into well-defined rounds. Real smartphones, by contrast, do not offer this synchronization. It follows that algorithms developed in the mobile telephone model cannot be directly implemented on real hardware.

With the goal of closing this gap, in Section IV we introduce the *asynchronous mobile telephone model* (aMTM), a variation of the MTM that removes the synchronous round assumption, allowing nodes and communication to operate at different speeds. The main advantage of the aMTM is that algorithms specified and analyzed in the aMTM can be directly implemented using existing smartphone peer-to-peer libraries. The main disadvantage is that the introduction of asynchrony complicates analysis.

In Section IV, we first study the question of whether simple random gossip processes even still converge in an asynchronous setting. The answer is not *a priori* obvious as it might be possible that the adversarial scheduler could starve certain messages. We answer this question positively by proving that an asynchronous version of random spread solves gossip in $O(nk\delta_{max})$ time, where δ_{max} is an upper bound on the maximum time for certain key steps (as is standard, we assume δ_{max} is determined by an adversary, can change between executions, and is unknown to the algorithm).

We then tackle the challenge of adapting the analysis techniques from the synchronous MTM, which heavily leverage the model synchronization, to show that information spreading still speeds up with respect to vertex expansion in the asynchronous setting. By introducing a novel analysis technique, in which we show that the probabilistic connection behavior in the aMTM over time sufficiently approximates synchronized behavior, to allow our more abstract graph theory results to apply. we prove that for $k = 1$, the single message spreads in at most $O(\sqrt{(n/\alpha)} \cdot \log^2 n\alpha \cdot \delta_{max})$ time. This result falls somewhere between our previous $O(n\delta_{max})$ result for gossip with $k = 1$ in the aMTM, and the bound of $O(\text{polylog}(n)/\alpha)$ rounds possible in the synchronous MTM for $k = 1$.

We argue that our introduction of the aMTM, as well as a powerful set of tools for analyzing information spreading in this setting, provides an important foundation for the future study of communication processes in realistic smartphone peer-to-peer models.

II. Related Work

In recent years, there has been a growing amount of research on smartphone peer-to-peer networking [14]–[20] (see [21] for a survey). In this paper, we both study and extend the mobile telephone model introduced in 2016 by Ghaffari and Newport [1]. Among other results, we prove that a simple random gossip process solves the problem in $O((k/\alpha)\log^4 n)$ rounds in the mobile telephone model, improving over the $O((k/\alpha)\log^5 n)$ result proved for a more complicated algorithm in [13]. To put these bounds into context, note that previous work in the mobile telephone model solved rumor spreading [1] and leader election [12] in $O(\text{polylog}(n)/\alpha)$ rounds. In the classical telephone model, a series of papers [7]–[10] (each optimizing the previous) established that simple random rumor spreading requires $O(\log^2 n/\alpha)$ rounds [10], which is optimal in the sense that for many α values, there

[2] In [13], we also study gossip under other assumptions, like changing communication graphs and the lack of good hash functions.

exists networks with a diameter in $\Omega(\log^2 n/\alpha)$. The fact that our gossip solution increases these bounds by a factor of k (ignoring log factors) is natural given that we allow only a constant number of tokens to be transferred per round.

As mentioned, random gossip processes more generally have been studied in other network models. These abstractions generally model time as synchronized rounds and by definition require nodes to select a neighbor uniformly at random in each round [25] [26]. More recent work has demonstrated that these protocols take advantage of key graph properties such as vertex expansion and graph conductance [27]. Asynchronous variants of these protocols have also been explored, where asynchrony is captured by assigning each node a clock following an unknown but well-defined probability distribution [25] [28]. The asynchronous MTM model introduced in our paper, by contrast, deploys a more general and classical approach to asynchrony in which an adversarial scheduler controls the time required for key events in a worst-case fashion.

III. Random Gossip in the Mobile Telephone Model

Here we study a simple random gossip process in the mobile telephone model. We begin by formalizing the model, the problem, and some graph theory preliminaries, before continuing with the algorithm description and analysis.

A. The Mobile Telephone Model

The mobile telelphone model describes a smartphone peer-to-peer network topology as an undirected connected graph $G = (V, E)$. A computational process (called a *node* in the following) is assigned to each vertex in V. The edges in E describe which node pairs are within communication range. In the following, we use $u \in V$ to indicate both the vertex in the topology graph as well as the computational process (node) assigned to that vertex. We use $n = |V|$ to indicate the network size.

Executions proceed in synchronous rounds labeled $1, 2, ...$, and we assume all nodes start during round 1. At the beginning of each round, each node $u \in V$ selects an *advertisement* to broadcast to its neighbors $N(u)$ in G. This advertisement is a bit string containing no more than $O(\log n + \ell_h)$ bits, where ℓ_h is the digest length of a standard hash function parameterized to obtain the desired collision resistance guarantees. After broadcasting its advertisement, node u then receives the advertisements broadcast by its neighbors in G for this round.

At this point, u decides to either send a *connection invitation* to a neighbor, or passively receive these invitations. If u decides to receive, and at least one connection invitation arrives at u, then node u can select at most one such incoming invitation to *accept*, forming a connection between u and the node v that sent the accepted invitation. Once u and v are connected, they can perform a bounded amount of reliable interactive communication before the round ends, where the magnitude of this bound is specified as a parameter of the problem studied. Notice that the model does not guarantee to deliver u all invitations sent to u by its neighbors. It instead only guarantees that if at least one neighbor of u sends an invitation, then u will receive a non-empty subset (selected arbitrarily) of these invitations before it must make its choice about acceptance.

If u instead chooses to send a connection invitation to a neighbor v, there are two outcomes. If v accepts u's invitation, a connection is formed as described above. Otherwise, u's invitation is implicitly rejected.

B. The Gossip Problem

The gossip problem is parameterized with a token count $k > 0$. It assumes k unique *tokens* are distributed to nodes at the beginning of the execution. The problem is solved once all nodes have received all k tokens. We treat the tokens as black boxes objects that are large compared to the advertisements. With this in mind, we assume the only ways for a node u to learn token t are: (1) u starts with token t; or (2) a node v that previously learned t sends the token to u during a round in which v and u are connected.

We assume that at most a constant number of tokens can be sent over a given connection. Notice that this restriction enforces a trivial $\Omega(k)$ round lower bound for the problem.

C. Vertex Expansion

Some network topologies are more suitable for information dissemination than others. In a clique, for example, a message can spread quickly through epidemic replication, while spreading a message from one endpoint of a line to another is necessarily slow. With this in mind, the time complexity of information dissemination algorithms are often expressed with respect to graph connectivity metrics such as *vertex expansion* or *graph conductance*. In this way, an algorithm's performance can be proved to improve along with available connectivity.

In this paper, as in previous studies of algorithms in the mobile telephone model [1], [11]–[13], we express our results with respect to vertex expansion (see [1] for an extended discussion of why this metric is more appropriate than conductance in our setting). Here we define this metric and establish a useful related property.

For fixed undirected connected graph $G = (V, E)$, and a given $S \subseteq V$, we define the *boundary* of S, indicated ∂S, as follows: $\partial S = \{v \in V \setminus S : N(v) \cap S \neq \emptyset\}$: that is, ∂S is the set of nodes not in S that are directly connected to S by an edge in E. We define $\alpha(S) = |\partial S|/|S|$. We define the *vertex expansion* α of a given graph $G = (V, E)$ as follows:

$$\alpha = \min_{S \subset V, 0 < |S| \leq n/2} \alpha(S).$$

Notice that despite the possibility of $\alpha(S) > 1$ for some S, we always have $\alpha \leq 1$. In more detail, this parameter ranges from $2/n$ for poorly connected graphs (e.g., a line) to values as large as 1 for well-connected graphs (e.g., a clique). Larger values indicate more potential for fast information dissemination.

The mobile telephone model requires the set of pairwise connections in a given round to form a matching in the topology graph $G = (V, E)$. The induces a connection between maximum matchings and the maximum amount of potential

communication in a given round. Here we adapt a useful result from [1] that formalizes the relationship between vertex expansion and these matchings as defined with respect to given partition.

In more detail, for a given graph $G = (V, E)$ and node subset $S \subset V$, we define $B(S)$ to be the bipartite graph with bipartitions $(S, V \setminus S)$, and the edge set $E_S = \{(u, v) : (u, v) \in E,\ u \in S,$ and $v \in V \setminus S\}$. Recall that the *edge independence number* of a graph H, denoted $\nu(H)$, describes the size of a maximum matching on H. For a given S, therefore, $\nu(B(S))$ describes the maximum number of concurrent connections that a network can support in the mobile telephone model between nodes in S and nodes outside of S. This property follows from the restriction in this model that each node can participate in at most one connection per round.

The following result notes that the vertex expansion does a good job of approximating the size of the maximum matching across any partition:

Lemma III.1 (from [1])**.** *Fix a graph $G = (V, E)$ with $|V| = n$ with vertex expansion α. Let $\gamma = \min_{S \subset V, |S| \leq n/2}\{\nu(B(S))/|S|\}$. It follows that $\gamma \geq \alpha/4$.*

D. The Random Spread Gossip Algorithm

We formalize our random spread gossip algorithm with the pseudocode labeled Algorithm 1. Here we summarize its behavior.

The basic idea of the algorithm is that in each round, each node advertises a hash of their token set. Nodes then attempt to connect only to neighbors that advertised a different hash, indicating their token sets are different. When two nodes connect, they can transfer a constant number of tokens in the non-empty set difference of their respective token sets.

As detailed in the pseudocode, the random spread algorithm implements the above strategy combined with some minor additional structure that supports the analysis. In particular, nodes partition rounds into *phases* of length $\lceil \log N \rceil$, where $N > 1$ is an upper bound on the maximum degree Δ in the network topology. Instead of each node deciding whether to send or receive connection invitations at the beginning of *each round*, they make this decision at the beginning of *each phase*, and then preserve this decision throughout the phase (this is captured in the pseudocode with the $status$ flag that is randomly set every $\lceil \log N \rceil$ rounds). Each receiver node also advertises whether or not it has been involved in a connection already during the current phase (as captured with the $done$ flag). A sender node will only consider neighbors that advertise a different hash, are receivers in the current phase, *and* have not yet been involved in a connection during the phase.

E. Analysis of Random Spread Gossip

Our goal is to prove the following result about the performance of random spread gossip:

Theorem III.2. *With high probability, the random spread gossip algorithm solves the gossip problem in $O((k/\alpha) \log^2 n \log N \log \Delta)$ rounds, when executed with $k > 0$ initial tokens and degree bound $N \geq \Delta$, in a network topology graph of size n, maximum degree Δ, and vertex expansion α.*

Algorithm 1 Random spread gossip (for node u).

Initialization:
$N \leftarrow$ upper bound on maximum degree in topology
$T \leftarrow$ initial tokens (if any) known by u
H is a hash function

For each round r:

if r mod $\lceil \log N \rceil = 1$ **then**
  $status \leftarrow$ random bit (1=sender; 0=receiver)
  $done \leftarrow 0$

`Advertise`$(\langle status, done, H(T, r), u \rangle)$
$A \leftarrow$ `RecvAdvertisements()`

$A' \leftarrow \{v \mid \langle 0, 0, h, v \rangle \in A, h \neq H(T, r)\}$
if $status = 1$ **and** $|A'| > 0$ **then**
  $v \leftarrow$ node selected with uniform randomness from A'
  (Attempt to connect with v. If successful, exchange a token in set difference.)
else if $status = 0$ **then**
  (If receive connection proposal(s): accept one, exchange token in the set difference, set $done \leftarrow 1$.)

We begin by establishing some preliminary notations and assumptions before continuing to the main proof argument.

a) Notation: For a fixed execution, let Q be the non-empty set of k tokens that the algorithm must spread. For each round $r > 0$ and node $u \in V$, let $T_u(r)$ be the tokens (if any) "known" by u at the start of round r (that is, the tokens that u starts with as well as every token it received through a connection in rounds 1 to $r - 1$).

For each $t \in Q$, and round $r > 0$, let $S_t(r) = \{v : t \in T_v(r)\}$ be the nodes that know token t at the start of round r. Let $n_t(r) = |S_t(r)|$ be the number of nodes that know token t iat the beginning of this round, and let $n_t^*(r) = \min\{n_t(r), n - n_t(r)\}$.

Finally, let $t^*(r) = \text{argmax}_{t \in Q}\{n_t^*(r)\}$ be a token t with the maximum $n_t^*(r)$ value in this round (breaking ties arbitrarily). According to Lemma III.1, which connects vertex expansion to matchings, there is a matching between nodes in $S_{t^*(r)}(r)$ and $V \setminus S_{t^*(r)}(r)$ of size at least $(\alpha/4) \cdot n^*_{t^*(r)}(r)$. Token $t^*(r)$, in other words, has the largest guaranteed potential to spread in round r among all tokens.[3] Accordingly, in the analysis that follows, we will focus on this token in each phase to help lower bound the amount of spreading we hope to achieve.

[3]To be slightly more precise, $(\alpha/4) \cdot n^*_{t^*(r)}(r)$ is a lower bound on the size of the matching across the cut defined by $t^*(r)$, so $t^*(r)$ is the token with the largest lower bound guarantee on the size of its matching.

b) Productive Connections and Hash Collisions: In the following, we say a given pairwise connection between nodes u and v in some round r is *productive* if $T_u(r) \neq T_v(r)$. That is, at least one of these two nodes learns a new token during the connection. By the definition of our algorithm, if u and v connect in round r, then it must be the case that $H(T_u(r), r) \neq H(T_v(r), r)$, where H is the hash function used by the random spread gossip algorithm. This implies $T_u(r) \neq T_v(r)$—indicating that every connection created by our algorithm is productive.

On the other hand, it is possible for some u, v, and r that even though $T_u(r) \neq T_v(r)$, $H(T_u(r), r) = H(T_v(r), r)$ due to a hash collision. For the sake of clarity, in the analysis that follows we assume that no hash collisions occur in the analyzed execution. Given the execution length is polynomial in the network size n, and there are at most n different token sets hashed in each round, for standard parameters the probability of a collision among this set would be extremely small, supporting our assumption. Even if such collisions do occasionally occur, their impact on our algorithm is negligible.

c) Matching Phases: Recall that our algorithm partitions rounds into *phases* of length $\lceil \log N \rceil$. For each phase $i > 0$, let $r_i = \lceil \log N \rceil \cdot (i-1)+1$ be the first round of that phase. Fix some arbitrary phase i and consider token $t = t^*(r_i)$, which, as argued above, is the token with the largest guaranteed potential to spread in round r_i. Our goal in this part of the analysis is to prove that with constant probability, our algorithm will create enough productive connections during this phase to well-approximate this potential. This alone is not enough to prove our algorithm terminates efficiently, as in some phases, it might be the case that *no* token has a large potential to spread. The next part of our argument will tackle this challenge by proving that over a sufficient number of phases the aggregate amount of progress must be large.

We begin by establishing the notion of a *productive subgraph*:

Definition III.3. *At the beginning of any round $r > 0$, we define the* productive subgraph *of the network topology $G = (V, E)$ for r as: $G_r = (V, E_r)$, where $E_r = \{\{u, v\} \mid \{u, v\} \in E, T_u(r) \neq T_v(r), u.status(r) \neq v.status(r)\}$, and for each $w \in V$, $w.status(r)$ indicates the value of the node w's status bit for the phase containing round r.*

That is, the productive subgraph for round r is the subgraph of G that contains only edges where the endpoints: (1) have different token set; and (2) have different statuses (one is a sender during this phase and one is a receiver). This subgraph contains every possible connection for a given round of our gossip algorithm (we ignore $done$ flags because, as will soon be made clear, we consider these graphs defined only for the first round of phases, a point at which all $done$ flags are reset to 0). Accordingly, a maximum matching on this subgraph upper bounds the maximum number of concurrent connections possible in a round.

We begin by lower bounding the size of the maximum matching in a productive subgraph at the beginning of a given phase i using the token $t = t^*(r_i)$. Recall that $n_t^*(r_i)$ is the number of nodes that know token t at the beginning of r, if less than half know the token, and otherwise indicates the number of nodes that do not know t.

Lemma III.4. *Fix some phase i. Let $t = t^*(r_i)$. Let G_{r_i} be the productive subgraph for round r_i, M_i be a maximum matching on G_{r_i}, and $m_i = |M_i|$. With constant probability (defined over the status assignments): $m_i \geq (\alpha/16) n_t^*(r_i)$.*

We now turn our attention to our gossip algorithm's ability to take advantage of the potential productive connections captured by the productive subgraph defined at the beginning of the phase. To do so, we first adapt a useful result on rumor spreading from [1] to the behavior of our gossip algorithm. Notice that it is the proof of the below adapted lemma that requires the use of the $done$ flag in our algorithm.

Lemma III.5 (adapted from Theorem 7.2 in [1])**.** *Fix a phase i. Let G' be a subgraph of the productive subgraph G_{r_i} that satisfies the following:*

1) *there is a matching of size m in G';*
2) *the set L of nodes in G' with sender status is of size m; and*
3) *for each node $u \in L$, every neighbor of u in G_{r_i} is in G'.*

With constant probability (defined over the random neighbor choices), during the first $\log \Delta$ rounds of phase i, at least $\Omega\left(\frac{m}{\log n \log \Delta}\right)$ neighbors of nodes in L in G' participate in a productive connection.

We now combine Lemmas III.4 and III.5 to derive our main result for this part of the analysis.

Lemma III.6. *Fix some phase i. Let $t = t^*(r_i)$. With constant probability, the number of productive connections in this phase is in $\Omega\left(\frac{\alpha n_t^*(r_i)}{\log n \log \Delta}\right)$.*

d) The Size Band Table: In the previous part of this analysis, we proved that with constant probability the number of productive connections in phase i is bounded with respect to the number of nodes that know $t^*(r_i)$. In the worst case, however, $t^*(r_i)$ might be quite small (e.g., at the beginning of an execution where each token is known by only a constant number of nodes, this value is constant). We must, therefore, move beyond a worst-case application of Lemma III.6, and amortize the progress over time to something more substantial.

To accomplish this goal, we introduce a data structure—a tool used only in the context of our analysis—that we call a *size band table*, which we denote as $\mathcal{S}$. This table has one column for each token $t \in T$, and $2 \log (n/2) + 1$ rows which we number $1, 2, ..., 2 \log n/2 + 1$.

As we will elaborate below, each row is associated with a range of values that we call a *band*. We call rows 1 through $\log (n/2)$ *growth bands*, and rows $\log (n/2) + 1$ through $2 \log (n/2) + 1$ *shrink bands*. Each cell in $\mathcal{S}$ contains a single bit. We update these bit values after every round of our gossip

algorithm to reflect the extent to which each token has spread in the system.

In more detail, for each round $r \geq 1$, we use $\mathcal{S}_r$ to describe the size band table at the beginning of round r. For each token $t \in T$ and row $i, 1 \leq 1 \leq 2\log(n/2) + 1$, we use $\mathcal{S}_r[t, i]$ to refer to the bit value in row i of the column dedicated to token t in the table for round r.

Finally, we define each of these bit values as follows. For each round $r \geq 1$, token $t \in T$, and growth band i (i.e., for each $i, 1 \leq i \leq \log(n/2)$), we define:

$$\mathcal{S}_r[t,i] = \begin{cases} 1 & \text{if at least } 2^i \text{ nodes know} \\ & \text{token } t \text{ at the beginning of round } r, \\ 0 & \text{else.} \end{cases}$$

Symmetrically, for each round $r \geq 1$, token $t \in T$, and shrink band i (i.e., for each $i, \log(n/2)+1 \leq i \leq 2\log(n/2)+1$), we define:

$$\mathcal{S}_r[t,i] = \begin{cases} 1 & \text{if less than } \frac{n}{2^{i-\log(n/2)}} \text{ nodes do } \textit{not} \\ & \text{know token } t \text{ at the beginning of round } r, \\ 0 & \text{else.} \end{cases}$$

A key property of the side band table is that as a given token t spreads, the cells in its column with 1 bits grow from the smaller rows toward the larger rows. That is, if row i is 1 at the beginning of a given round, all smaller rows for that token are also 1 at the beginning of that round. Furthermore, because nodes never lose knowledge of a token, once a cell is set to 1, it remains 1.

row #		band size	
9	0 0 0 0 0	< 1	bound on # nodes that do not know rumor (if 1 in cell)
8	0 0 0 0 1	< 2	
7	0 0 0 0 1	< 4	
6	0 0 0 0 1	< 8	
5	0 0 1 0 1	< 16	
shrink / growth			
4	0 0 1 0 1	≥ 16	bound on # nodes that do know rumor (if 1 in cell)
3	0 0 1 0 1	≥ 8	
2	1 0 1 0 1	≥ 4	
1	1 0 1 1 1	≥ 2	
	t_1 t_2 t_3 t_4 t_5		

Fig. 1. An example size band table for token set $T = \{t_1, t_2, t_3, t_4, t_5\}$ and network size $n = 32$. There is one column for each token. The largest row containing a 1 for a given token bounds the token spread. Token t_1, for example, has spread to at least 4 out of the 32 nodes, while token t_5 is known to all but 1 node (indicating that it has spread to at least 31). In this example table, token t_3, which is spread to somewhere between 16 to 24 nodes, has the biggest potential to spread in the current round

When all rows for a given token t are set to 1, it follows that all nodes know t. This follows because the definition of shrink band $i = 2\log(n/2) + 1$ being set to 1 is that the number of nodes that do *not* know t is strictly *less* than:

$$\begin{aligned} \frac{n}{2^{i-\log(n/2)}} &= \frac{n}{2^{2\log(n/2)+1-\log(n/2)}} \\ &= \frac{n}{2^{\log(n/2)+1}} \\ &= \frac{n}{2^{\log(n/2)} \cdot 2^1} \\ &= 1. \end{aligned}$$

e) Amortized Analysis of Size Band Table Progress: As the size band table increases the number of 1 bits, we say it *progresses* toward a final state of all 1 bits. Here we perform an amortized analysis of size band table progress.

To do so, we introduce some notation. For each phase i, and token $t \in T$, let $b_t(i)$ be the largest row number that contains a 1 in t's column in $\mathcal{S}_{r_i}$. We call this the *current band* for token t in phase i.

Let $a(i) = |b_{t^*(r_i)}(i) - \log(n/2)|$ define the distance from the current band of token $t^*(r_i)$ to the center row number $\log(n/2)$. By the definition of $t^*(r_i)$, no token has a current band closer to $\log(n/2)$ than $t^*(r_i)$ at the start of phase i. We say that phase i is *associated* with the current band for $t^*(r_i)$.

Finally, for a given phase i, with $t = t^*(r_i)$, we say this phase is *successful* if the number of productive connections during the phase is at least as large as the lower bound specified by Lemma III.6; i.e., there are at least $\frac{\gamma\alpha n_t^*(r_i)}{\log n \log \Delta}$ productive connections. where $\gamma > 0$ is the constant hidden in the asymptotic bound in the lemma statement.

Our first goal in this part of the analysis, is to bound the number of successful phases that can be associated with each band. To do so, we differentiate between two different types of successful phases, and then bound each separately.

Definition III.7. *Fix some phase i that is associated with some band j at distance $a(i)$ from the center of the size band table. We say phase i is an* upgrade *phase if there exists a subset of the productive connections during phase i that push some token t's current band to a position j' with $|\log(n/2) - j'| < a(i)$. If a phase is not an upgrade phase, and at least one node is missing at least one token, we call it a* fill *phase.*

Stated less formally, we call a phase an upgrade phase if it pushes some token's count closer to the center of the size band table—row $\log(n/2)$—than the band associated with the phase. Our definition is somewhat subtle in that it must handle the case where during a phase a token count does grow to be closer to the center of the size band table, but then its count continues to grow until it pushes *more than* distance $a(i)$ above the center. We still want to count this as an upgrade phase (hence the terminology about there existing some *subset* of the connections that push the count closer).

Our goal is to bound the number of successful phases possible before all tokens are spread. We begin with bound on upgrade phases (which hold whether or not the phase is successful). Our subsequent bound on fill phases, however, considers only successful phases.

Lemma III.8. *There can be at most $k(2\log(n/2) + 1)$ upgrade phases.*

We now bound the number of successful fill phases. To do so, we note that the number of fill phases associated with a given band is bounded by the worst case number of connections needed before some token's count must advance past that band. For bands associated with large ranges this worst case number is large. As shown in the following lemma, however, the number of connections in phases associated with large bands grows proportionally large as well. This balancing of growth required and growth obtained is at the core of our amortized analysis.

Lemma III.9. *There can be at most $O((k/\alpha)\log^2 n \log \Delta)$ successful fill phases.*

Proof. Consider a group of successful fill phases associated with some band j at distance a_j from the center of the size band table. Because these are fill phases, the productive connections generated during these phases can never push some token's count (perhaps temporarily) closer than distance a_j from the center of the table (any phase in which this occurs becomes, by definition, an upgrade phase).

One way to analyze the distribution of the productive connections during these phases is to consider a generalization of the size band table in which we record in each cell $[t, i]$ the total number of productive connections that spread token t while its count falls into the band associated with row i. (Of course, many connections for a given token might occur in a given round, in which we case, we process them one by one in an arbitrary order while updating the cell counts.)

If we apply this analysis only for the fill phases fixed above, then we know that the counts in all cells of distance less than a_j from the center of the table remain at 0. By the definition of the size band table, for a given token t, the maximum number of connections we can add to cells of distance at least a_j from the center is loosely upper bounded by $2 \cdot 2^{\log(n/2)-a_j}$ (the extra factor of two captures both growth and shrink band cells at least distance a_j). Therefore, the total number of productive connections we can process into cells at distance at least a_j is at most $2k2^{\log(n/2)-a_j}$.

By the definition, each phase i that is a successful fill phase associated with j generates at least $\frac{\gamma\alpha n_t^*(r_i)}{\log n \log \Delta}$ productive connections, where $t = t^*(r_i)$. By the definition of $t^*(r_i)$, t's current band is distance a_j from the center. Therefore, $n_t^*(r_i)$ is within a factor of 2 of $2^{\log(n/2)-a_j}$. By absorbing that constant factor into the constant γ (to produce a new constant γ'), it follows that this phase generates at least

$$z = \frac{\gamma'\alpha 2^{\log(n/2)-a_j}}{\log n \log \Delta}$$

new productive connections. Combined with our above upper bound on the total possible productive connections for successful fill phases associated with j, it follows that the total number of successful fill phases associated with j as less than:

$$\begin{aligned} z^{-1}2k2^{\log(n/2)-a_j} &= \left(\frac{\log n \log \Delta}{\gamma'\alpha 2^{\log(n/2)-a_j}}\right) 2k2^{\log(n/2)-a_j} \\ &= \Theta((k/\alpha)\log n \log \Delta). \end{aligned}$$

We multiply this bound over $2\log(n/2)+1$ possible bands to derive $O((k/\alpha)\log^2 n \log \Delta)$ total possible successful fill phases, providing the bound claimed by the lemma statement. □

f) Pulling Together the Pieces: We are now ready to combine the above lemmas to prove our main theorem.

Proof (of Theorem III.2). Combining Lemmas III.8 and III.9, it follows that there can be at most $\ell = k(2\log(n/2)+1) + O((k/\alpha)\log^2 n \log \Delta) = O((k/\alpha)\log^2 n \log \Delta)$ successful upgrade and fill phases before all k tokens are spread.

By Lemma III.6, if the token spreading is not yet complete, then the probability that the current phase is successful is lower bounded by some constant probability $p > 0$. The actual probability might depend on the execution history up until the current phase, but the lower bound of p always holds, regardless of this history. We can tame these dependencies with a stochastic dominance argument.

In more detail, for each phase i before the tokens are spread, we define a trivial random variable $\hat{X}_i$ that is 1 with independent probability p, and otherwise 0. Let X_i, by contrast, be the random indicator variable that is 1 if phase i is successful, and otherwise 0. For each phase i that occurs after the tokens are spread, $\hat{X}_i = X_i = 1$ by default.

Note that for each i, X_i stochastically dominates $\hat{X}_i$. It follows that if $\hat{Y}_T = \sum_{i=1}^{T} \hat{X}_i$ is greater than some x with some probability $\hat{p}$, then $Y_T = \sum_{i=1}^{T} X_i$ is greater than x with probability at least $\hat{p}$.

With this established, consider the first $T = (c/p)\ell$ phases, for some constant $c \geq 2$. Note that for this value of T, $E[\hat{Y}_T] = c\ell$. Because $\hat{Y}_T$ is the sum of independent random variables, we concentrate around this expectation. In particular, we once again apply the following form of a Chernoff Bound:

$$\Pr(Y \leq (1-\delta)\mu) \leq e^{-\frac{\delta^2\mu}{2}},$$

for $Y = \hat{Y}_T$, $\delta = 1/2$, and $\mu = c\ell$, to derive that the probability that $\hat{Y}_T \leq (c/2)\ell \geq \ell$, is upper bounded by $e^{-\frac{c\ell}{8}}$. The same bound therefore holds for the probability that $Y_T \leq (c/2)\ell$. Notice that this error bound is polynomially small in n with an exponent that grows with constant c. It follows, therefore, that with high probability in n, that token spreading succeeds in the first $T = (c/p)\ell = O((k/\alpha)\log^2 n \log \Delta)$ phases.

To achieve the final *round* complexity bound claimed by the theorem statement, we multiply this upper bound on phases by the length of $\log N$ rounds per phase. □

IV. Random Gossip in the Asynchronous Mobile Telephone Model

The mobile telephone model captures the basic dynamics of the peer-to-peer libraries included in standard smartphone

operating systems. This abstraction, however, makes simplifying assumptions—namely, the assumption of synchronized rounds. In this section we analyze the performance of simple random gossip processes in a more realistic version of the model that eliminates the synchronous round assumption. In particular, we first define the asynchronous mobile telephone model (aMTM), which describes an event-driven peer-to-peer abstraction in which an adversarial scheduler controls the timing of key events in the execution.

A key property of the asynchronous mobile telephone model is the abstraction of complex communication details from algorithmic design. The sophisticated communication patterns possible in the asynchronous setting are abstracted as a continuously-looping process, the execution of which can be altered by access to data members modified asynchronously by the model. The maximum duration of a single iteration of this loop is captured by the main model parameter δ_{max}, which is not exposed to the algorithm.

We then implement the random spread gossip algorithm for the asynchronous setting using our abstraction (a detailed specification of both the model and algorithm can be found in the full version of this paper [**?**]). We conclude with an in-depth analysis of the performance of this algorithm and provide two primary results, the first of which is a proof of worst-case convergence.

Theorem IV.1. *The asynchronous random gossip algorithm takes time $O(nk\delta_{max})$ to spread all k tokens to all n nodes of the network where δ_{max} is the maximum amount of time between iterations of the algorithm loop.*

Finally, we analyze the spread of a single token in the network to demonstrate that the performance of the asynchronous random spread gossip algorithm still improves with the vertex expansion α, thereby proving the following theorem.

Theorem IV.2. *The asynchronous random spread gossip algorithm takes time at most $O(\sqrt{n/\alpha}\log^2(n\alpha)\delta_{max})$, where n is the number of nodes in the network, α is the vertex expansion, and δ_{max} is the maximum time required for an iteration of the asynchronous mobile telephone model loop.*

REFERENCES

[1] M. Ghaffari and C. Newport, "How to discreetly spread a rumor in a crowd," in *Proceedings of the International Symposium on Distributed Computing (DISC)*, 2016.

[2] A. M. Frieze and G. R. Grimmett, "The shortest-path problem for graphs with random arc-lengths," *Discrete Applied Mathematics*, vol. 10, no. 1, pp. 57–77, 1985.

[3] ——, "The shortest-path problem for graphs with random arc-lengths," *Discrete Applied Mathematics*, vol. 10, no. 1, pp. 57–77, 1985.

[4] G. Giakkoupis, "Tight bounds for rumor spreading in graphs of a given conductance," in *Proceedings of the Symposium on Theoretical Aspects of Computer Science (STACS)*, 2011.

[5] G. Giakkoupis and T. Sauerwald, "Rumor spreading and vertex expansion," in *Proceedings of the ACM-SIAM symposium on Discrete Algorithms (SODA)*, 2012, pp. 1623–1641.

[6] G. Giakkoupis, "Tight bounds for rumor spreading in graphs of a given conductance," in *Proceedings of the Symposium on Theoretical Aspects of Computer Science (STACS)*, 2011.

[7] F. Chierichetti, S. Lattanzi, and A. Panconesi, "Rumour spreading and graph conductance." in *Proceedings of the ACM-SIAM symposium on Discrete Algorithms (SODA)*, 2010.

[8] G. Giakkoupis and T. Sauerwald, "Rumor spreading and vertex expansion," in *Proceedings of the ACM-SIAM symposium on Discrete Algorithms (SODA)*. SIAM, 2012, pp. 1623–1641.

[9] N. Fountoulakis and K. Panagiotou, "Rumor spreading on random regular graphs and expanders," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2010, pp. 560–573.

[10] G. Giakkoupis, "Tight bounds for rumor spreading with vertex expansion," in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2014.

[11] M. Dinitz, J. Fineman, S. Gilbert, and C. Newport, "Load balancing with bounded convergence in dynamic networks," in *IINFOCOM*, 2017, pp. 1–9.

[12] C. Newport, "Leader election in a smartphone peer-to-peer network," in *Proceedings of the IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2017, full version available online at: http://people.cs.georgetown.edu/ cnewport/pubs/le-IPDPS2017.pdf.

[13] ——, "Gossip in a smartphone peer-to-peer network," in *PODC*, 2017.

[14] N. Suzuki, J. L. F. Zamora, S. Kashihara, and S. Yamaguchi, "Soscast: Location estimation of immobilized persons through sos message propagation," in *Proceedings of the International Conference on Intelligent Networking and Collaborative Systems (INCoS)*. IEEE, 2012, pp. 428–435.

[15] G. Aloi, M. Di Felice, V. Loscrì, P. Pace, and G. Ruggeri, "Spontaneous smartphone networks as a user-centric solution for the future internet," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 26–33, 2014.

[16] D. Reina, M. Askalani, S. Toral, F. Barrero, E. Asimakopoulou, and N. Bessis, "A survey on multihop ad hoc networks for disaster response scenarios," *International Journal of Distributed Sensor Networks*, vol. 11, no. 10, p. 647037, 2015.

[17] Z. Lu, G. Cao, and T. La Porta, "Networking smartphones for disaster recovery," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2016, pp. 1–9.

[18] A. Holzer, S. Reber, J. Quarta, J. Mazuze, and D. Gillet, "Padoc: Enabling social networking in proximity," *Computer Networks*, vol. 111, pp. 82–92, 2016.

[19] O. Garden, "Firechat," *Internet: https://www.opengarden.com/ [Accessed: 07/20/2018]*, 2018.

[20] ——, "The open garden hotspot," *Internet: https://www.opengarden.com/ [Accessed: 07/20/2018]*, 2018.

[21] H. Nishiyama, M. Ito, and N. Kato, "Relay-by-smartphone: realizing multihop device-to-device communications," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 56–65, 2014.

[22] K. Doppler, M. Rinne, C. Wijting, C. B. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to lte-advanced networks," *IEEE Communications Magazine*, vol. 47, no. 12, 2009.

[23] F. Wang, C. Xu, L. Song, and Z. Han, "Energy-efficient resource allocation for device-to-device underlay communication," *IEEE Transactions on Wireless Communications*, vol. 14, no. 4, pp. 2082–2092, 2015.

[24] J. Liu, N. Kato, J. Ma, and N. Kadowaki, "Device-to-device communication in lte-advanced networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 1923–1940, 2015.

[25] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[26] R. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking, "Randomized rumor spreading," in *Proceedings of the Annual Symposium on Foundations of Computer Science*, 2000.

[27] D. Mosk-Aoyama and D. Shah, "Computing separable functions via gossip," in *Proceedings of the ACM Symposium on Principles of Distributed Computing*, 2006.

[28] G. Giakkoupis, Y. Nazari, and P. Woelfel, "How asynchrony affects rumor spreading time," in *Proceedings of the ACM Symposium on Principles of Distributed Computing*, 2016.