On the Complexity of Information Spreading in Dynamic Networks

Chinmoy Dutta∗ Gopal Pandurangan† Rajmohan Rajaraman∗ Zhifeng Sun∗ Emanuele Viola∗

Abstract

We study how to spread \( k \) tokens of information to every node on an \( n \)-node dynamic network, the edges of which are changing at each round. This basic gossip problem can be completed in \( O(n + k) \) rounds in any static network, and determining its complexity in dynamic networks is central to understanding the algorithmic limits and capabilities of various dynamic network models. Our focus is on token-forwarding algorithms, which do not manipulate tokens in any way other than storing, copying and forwarding them.

We first consider the strongly adaptive adversary model where in each round, each node first chooses a token to broadcast to all its neighbors (without knowing who they are), and then an adversary chooses an arbitrary connected communication network for that round with the knowledge of the tokens chosen by each node. We show that \( \Omega(n + nk/\log n) \) rounds are needed for any randomized (centralized or distributed) token-forwarding algorithm, thus resolving an open problem raised in [KLO10]. The bound applies to a wide class of initial token distributions, including well-mixed ones in which each node has each token independently with a constant probability.

Our result for the strongly adaptive adversary model motivates us to study the weakly adaptive adversary model where in each round, the adversary is required to lay down the network first, and each node then sends a possibly distinct token to each of its neighbors. We propose a simple randomized distributed algorithm where in each round, along every edge \((u, v)\), a token sampled uniformly at random from the symmetric difference of the sets of tokens held by node \( u \) and node \( v \) is exchanged. We prove that starting from any well-mixed distribution of tokens where each node has each token independently with a constant probability, this algorithm solves the \( k \)-gossip problem in \( O((n + k) \log n \log k) \) rounds with high probability. We then show how the above uniform sampling problem can be solved using \( O(\log^{1/2} n) \) bits of communication, making the overall algorithm communication-efficient.

We then give an offline algorithm that, given the entire sequence of graphs chosen by the adversary, solves \( k \)-gossip for every starting distribution in \( O((n + k) \log^2 n) \) rounds. Finally, we present an \( O(n \min\{k, \sqrt{k \log n}\}) \)-round offline algorithm in which each node can only broadcast a single token to all of its neighbors in each round.

Keywords: Dynamic networks, Information Spreading, Gossip, Distributed Computation, Communication Complexity

∗College of Computer and Information Science, Northeastern University, Boston, 02115, USA. Email: \{chinmoy,rraj,austin,viola\}@ccs.neu.edu. Chinmoy Dutta is supported in part by NSF grant CCF-0845003 and a Microsoft grant to Ravi Sundaram; Rajmohan Rajaraman and Zhifeng Sun are supported in part by NSF grant CNS-0915985; Emanuele Viola is supported by NSF grant CCF-0845003.

†Division of Mathematical Sciences, Nanyang Technological University, Singapore 637371 and Department of Computer Science, Brown University, Providence, RI 02912, USA. Email: gopalpandurangan@gmail.com. Supported in part by the following research grants: Nanyang Technological University grant M58110000, Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 2 grant MOE2010-T2-2-082, and a grant from the US-Israel Binational Science Foundation (BSF).
1 Introduction

In a dynamic network, nodes (processors/end hosts) and communication links can appear and disappear over time. Modern networking technologies such as ad hoc wireless, sensor, and mobile networks, overlay and peer-to-peer (P2P) networks are inherently dynamic, bandwidth-constrained, and unreliable. This necessitates the development of a solid theoretical foundation to design efficient, robust, and scalable distributed algorithms and understand the power and limitations of distributed computation on such networks. Such a foundation is critical to realize the full potential of these large-scale dynamic networks.

In this paper, we study a fundamental problem of information spreading, called $k$-gossip, on dynamic networks. This problem was analyzed for static networks by Topkis [Top85], and was first studied on dynamic networks by Kuhn, Lynch, and Oshman [KLO10]. In $k$-gossip (also referred to as $k$-token dissemination), there are $k$ pieces of information (tokens) that are initially present in some nodes and the problem is to disseminate the $k$ tokens to all the $n$ nodes in the network, under the bandwidth constraint that one token can go through an edge per round. This problem is a fundamental primitive for distributed computing; indeed, solving $n$-gossip, where each node starts with exactly one token, allows any function of the initial states of the nodes to be computed, assuming the nodes know $n$ [KLO10].

The dynamic network models that we consider in this paper allow an adversary to choose any communication links among the nodes for each round, with the only constraint being that the resulting communication graph be connected in each round. Our adversarial models are either the same as or closely related to those adopted in recent studies [AKL08, KLO10, OW05, CFQS10].

The focus of this paper is on the power of token-forwarding algorithms, which do not manipulate tokens in any way other than storing, copying, and forwarding them. Token-forwarding algorithms are simple and easy to implement, typically incur low overhead, and have been widely studied (e.g., see [Lei91b, Pel00]). In any $n$-node static network, a simple token-forwarding algorithm that pipelines token transmissions up a rooted spanning tree, and then broadcasts them down the tree completes $k$-gossip in $O(n + k)$ rounds [Top85, Pel00], which is tight since $\Omega(n + k)$ rounds is a straightforward lower bound due to bandwidth constraints. The central question motivating our study is whether a linear or near-linear bound is achievable for $k$-gossip on dynamic networks.

1.1 Our results

Our first result, in Section 2, is a lower bound for $k$-gossip under a worst-case model due to [KLO10], which we call the strongly adaptive adversary model. We now define the model and then state the theorem.

**Definition 1 (Strongly adaptive adversary).** In each round of the strongly adaptive adversary model, each node first chooses a token to broadcast to all its neighbors (without knowing who they are), and then the adversary chooses an arbitrary connected communication network for that round with the knowledge of the tokens chosen by each node.

We note that the choice made by each node may depend arbitrarily on the tokens held by that and other nodes. Hence this model includes centralized algorithms.

**Theorem 1.** (a) Any token-forwarding algorithm for $k$-gossip needs $\Omega(nk / \log n) + n$ rounds in the strongly adaptive adversary model starting from any initial token distribution in which each of $k \leq n$ tokens is held by exactly one node. (b) In addition, the same bound holds with high probability over an initial token distribution where each of the $n$ nodes receives each of $k \leq n$ tokens independently with probability $3/4$.

This result resolves an open problem raised in [KLO10], improving their lower bound of $\Omega(n \log n)$ for $k = \omega(\log n \log \log n)$, and matching their upper bound to within a logarithmic factor. Our lower bound also enables a better comparison of token-forwarding with an alternative approach based on network coding.
due to [Hae11, HK11]. Assuming the size of each message is bounded by the size of a token, network coding completes $k$-gossip in $O(nk/\log n + n)$ rounds for $O(\log n)$-bit tokens, and $O(n + k)$ rounds for $\Omega(n \log n)$ bit tokens. Thus, for large token sizes, our result establishes a factor $\Omega(\min\{n, k\}/\log n)$ gap between token-forwarding and network coding, a significant new bound on the network coding advantage for information dissemination.\footnote{The strongly adaptive adversary model allows each node to broadcast one token in each round, and thus our bounds hold regardless of the token size.} Furthermore, for small token and message sizes (e.g., $O(\text{polylog}(n))$ bits), we do not know of any algorithm (network coding, or otherwise) that completes $k$-gossip against a strongly adaptive adversary in $o(nk/\text{polylog}(n))$ rounds.

Our lower bound for the strongly adaptive adversary model motivates us to study models which restrict the power of the adversary and/or strengthen the capabilities of the algorithm. We would like to restrict the adversary power as little as possible and yet design fast algorithms.

**Definition 2** (Weakly adaptive adversary). *In each round of the weakly adaptive adversary model, the adversary is required to lay down the communication network first, before the nodes can communicate. Hence nodes get to know their neighbors and thus each node can send a possibly distinct token to each of its neighbors. Note that the adversary still has full control of the topology in each round.*

We propose a simple protocol which we call the symmetric difference (SYM-DIFF) protocol.

**Definition 3** (SYM-DIFF protocol). *The protocol SYM-DIFF works as follows: in each round, independently along every edge $(u,v)$, sample a token $t$ uniformly at random from the symmetric difference (i.e., XOR) of the sets of tokens held by node $u$ and node $v$ at the start of the round. Then the node that holds $t$ sends it to the other node.*

Our second main result, in Section 3.1, shows that in the weakly adaptive model, the SYM-DIFF protocol beats the lower bound for mixed starting distribution of Theorem 1.

**Theorem 2.** *Starting from any well-mixed distribution of tokens where each of the $n$ nodes has each of the $k$ tokens independently with a positive constant probability, the SYM-DIFF protocol completes $k$-gossip in $O((n + k) \log n \log k)$ rounds with high probability. The probability is both over the initial assignment of tokens and the randomness of the protocol.*

A communication-efficient implementation of SYM-DIFF hinges on the communication complexity of sampling a uniform element from the symmetric difference of two sets. As another technical contribution, we give an explicit, communication-efficient protocol for this task in Section 3.2.

**Theorem 3.** *Let Alice and Bob have two subsets $A \subseteq [k]$ and $B \subseteq [k]$ respectively. There is an explicit, private-coin protocol to sample a random element from the symmetric difference of the two sets, $A \oplus B := (A \setminus B) \cup (B \setminus A)$, such that the sampled distribution is statistically $\epsilon$-close to the uniform distribution on $A \oplus B$ and the protocol uses $O(\log^{3/2}(k/\epsilon))$ bits of communication.*

As discussed in Section 3, an improvement on pseudorandom generators for combinatorial rectangles would improve the communication in Theorem 3 to $O(\lg n/\epsilon)$. We also note that for SYM-DIFF to be communication-efficient it is important that we work with symmetric difference as opposed to set difference, which might have looked a natural choice. This is because Theorem 3 becomes false if we replace symmetric difference $A \oplus B$ with set difference $A \setminus B$. For the latter, communication $\Omega(k)$ is required, due to the lower bounds for disjointness [KS92, Raz92].

Although we have only been able to establish the efficiency of the SYM-DIFF protocol starting from well-mixed distributions as in Theorem 2, we conjecture that in fact SYM-DIFF is efficient starting from
any token distribution. A priori, however, it is unclear if there is any token-forwarding algorithm that solves $k$-gossip in $O(n + k)$ rounds even in an offline setting, in which the network can change arbitrarily each round, but the entire evolution is known to the algorithm in advance. Our next result, in Section 4.1, resolves this problem.

**Definition 4 (Offline algorithm).** An offline algorithm for $k$-gossip takes as input an initial token distribution and a sequence of $nk$ graphs $G_1, \ldots, G_{nk}$, where $G_t$ represents the communication network in round $t$. The output of the algorithm is a schedule that specifies, for each $t$, each edge $e$ of $G_t$, a token (if any) sent along $e$ in each direction. The length of the schedule is the largest $t$ for which a token is sent on any edge in round $t$.

**Theorem 4.** There is a polynomial-time randomized offline algorithm that returns, for every $k$-gossip instance, a schedule of length $O((n + k) \log^2 n)$ with high probability.

Like SYM-DIFF, the schedule returned by the above offline algorithm allows each node to send a possibly distinct token to each of its neighbors in each round. However, in some applications, e.g., wireless networks, the preferred mode of communication is broadcast. Hence, we also consider offline broadcast schedules where each node can only broadcast a single token to all of its neighbors in each round and show the following result in Section 4.2.

**Theorem 5.** There is a polynomial-time randomized offline algorithm that returns, for every $k$-gossip instance, a broadcast schedule of length $O(n \min\{k, \sqrt{k \log n}\})$, with high probability.

### 1.2 Related work

Information spreading (or dissemination) in networks is a fundamental problem in distributed computing and has a rich literature. The problem is generally well-understood on static networks, both for interconnection networks [Lei91a] as well as general networks [Lyn96, Pel00, AW04]. In particular, the $k$-gossip problem can be solved in $O(n + k)$ rounds on any $n$-node static network [Top85]. There also have been several papers on broadcasting, multicasting, and related problems in static heterogeneous and wireless networks (e.g., see [ABNLP91, BYGI87, BNGNS00, CMS01]).

Dynamic networks have been studied extensively over the past three decades. Early studies focused on dynamics that arise when edges or nodes fail. A number of fault models, varying according to extent and nature (e.g., probabilistic vs. worst-case) of faults allowed, and the resulting dynamic networks have been analyzed (e.g., see [AW04, Lyn96]). There have been several studies that constrain the rate at which changes occur, or assume that the network eventually stabilizes (e.g., see [AAG87, DoI00, GB81]).

There also has been considerable work on general dynamic networks. Early studies in this area include [AGR92, APSPS92], which introduce building blocks for communication protocols on dynamic networks. Another notable work is the local balancing approach of [AL94] for solving routing and multi-commodity flow problems on dynamic networks, which has also been applied to multicast, anycast, and broadcast problems on mobile ad hoc networks [ABBS01, ABS03, JRS03]. To address highly unpredictable network dynamics, stronger adversarial models have been studied by [AKL08, OW05, KLO10] and others; see the recent survey of [CFQS10] and the references therein. Unlike prior models on dynamic networks, these models and ours do not assume that the network eventually stops changing; the algorithms are required to work correctly and terminate even in networks that change continually over time. The recent work of [CST12], studies the flooding time of Markovian evolving dynamic graphs, a special class of evolving graphs. The survey of [KO11] summarizes recent work on dynamic networks. We also note that our model and the ones we have discussed thus far only allow edge changes from round to round; the recent work of [APRU12] studies a dynamic network model where both nodes and edges can change in each round.
Recent work of [Hae11, HK11] presents information spreading algorithms based on network coding [ACLY00]. As mentioned earlier, one of their important results is that the \( k \)-gossip problem on the adversarial model of [KLO10] can be solved using network coding in \( O(n + k) \) rounds assuming the token sizes are sufficiently large (\( \Omega(n \log n) \) bits). For further references to using network coding for gossip and related problems, we refer to [Hac11, HK11, ABCHL11, BAL10, DMC06, MAS06] and the references therein.

As we show in Section 4.2, the problem of finding an optimal broadcast schedule in the offline setting reduces to the Steiner tree packing problem for directed graphs [CS06]. This problem is closely related to the directed Steiner tree problem (a major open problem in approximation algorithms) [CCC+98, ZK02] and the gap between network coding and flow-based solutions for multicast in arbitrary directed networks [AC04, SET03].

Finally, we note that a number of recent studies solve \( k \)-gossip and related problems using gossip-based processes, in which each node exchanges information with a small number of randomly chosen neighbors in each round, e.g., see [BCEG10, DGH+87, KK02, CP12, KSSV00, MAS06, BGPS06] and the references therein. All these studies assume a static communication network, and do not apply directly to the models considered in this paper.

## 2 Lower bound for the strongly adaptive adversary model

In this section, we prove Theorem 1. We first define the adversary used in the proof of Theorem 1.

**Adversary:** The strategy of the adversary is simple. We use the notion of *free edge* introduced in [KLO10]. In a given round \( r \), we call an edge \((u,v)\) *free* if at the start of the round, \( u \) has the token that \( v \) broadcasts in the round and \( v \) has the token that \( u \) broadcasts in the round; an edge that is not free is called *non-free.* Thus, if \((u,v)\) is a free edge in a particular round, neither \( u \) nor \( v \) can gain any new token through this edge in the round. Since we are considering a strong adversary model, at the start of each round, the adversary knows for each node \( v \), the token that \( v \) will broadcast in that round. In round \( r \), the adversary constructs the communication graph \( G_r \) as follows. First, the adversary adds all the free edges to \( G_r \). Let \( C_1, C_2, \ldots, C_l \) denote the connected components thus formed. The adversary then guarantees the connectivity of the graph by selecting an arbitrary node in each connected component and connecting them in a line. Figure 1 illustrates the construction.

The network \( G_r \) thus constructed has exactly \( l - 1 \) non-free edges, where \( l \) is the number of connected components formed by the free edges of \( G_r \). If \((u,v)\) is a non-free edge in \( G_r \), then \( u, v \) will gain at most one new token each through \((u,v)\). We refer to this exchange on a non-free edge as a *useful token exchange.*

Our proof proceeds as follows. First, we show that with high probability over the initial assignment of tokens, in every round there are at most \( O(\lg n) \) useful token exchanges. Then we note that, again with high probability over the initial assignment of tokens, overall \( \Omega(nk) \) useful token exchanges must occur for the protocol to complete.

**Definition 5.** We say that a sequence of nodes \( v_1, v_2, \ldots, v_k \) is half-empty in round \( r \) with respect to a sequence of tokens \( t_1, t_2, \ldots, t_k \) if the following condition holds at the start of round \( r \): for all \( 1 \leq i, j \leq k \), \( i \neq j \), either \( v_i \) is missing \( t_j \) or \( v_j \) is missing \( t_i \). We then say that \( \langle v_i \rangle \) is half-empty with respect to \( \langle t_i \rangle \) and refer to the pair \( (\langle v_i \rangle, \langle t_i \rangle) \) as a half-empty configuration of size \( k \).

**Lemma 6.** If \( m \) useful token exchanges occur in round \( r \), then there exists a half-empty configuration of size at least \( m/2 + 1 \) at the start of round \( r \).

**Proof.** Consider the network \( G_r \) in round \( r \). Each non-free edge can contribute at most 2 useful token exchanges. Thus, there are at least \( m/2 \) non-free edges. Based on the adversary we consider, no useful
In the above inequality, \(E_l\) denotes an event of size \(l\) half-empty configurations. Let \(E_l\) denote the event that \(n_l\) nodes are in half-empty configurations of size \(l\). For a pair of nodes \(i\) and \(j\), let \(t_i\) and \(t_j\) be the tokens broadcast by \(i\) and \(j\) in round \(r\). For \(i \neq j\), since \(v_i\) and \(v_j\) are in different connected components, \((v_i, v_j)\) is a non-free edge in round \(r\); hence, at the start of round \(r\), either \(v_i\) is missing \(t_j\) or \(v_j\) is missing \(t_i\). Thus, the sequence \((v_i)\) of nodes of size at least \(m/2 + 1\) is half-empty with respect to the sequence \((t_i)\) at the start of round \(r\).

An important point to note about the definition of a half-empty configuration is that, in a given round, it only depends on the tokens held by the nodes; it is independent of the tokens that the nodes broadcast. This allows us to prove the following easy lemma that shows a monotonicity property of half-empty configurations.

**Lemma 7 (Monotonicity Property).** If a sequence \((v_i)\) of nodes is half-empty with respect to \((t_i)\) at the start of round \(r\), then \((v_i)\) is half-empty with respect to \((t_i)\) at the start of round \(r'\) for any \(r' \leq r\). Hence, the size of the largest half-empty configuration cannot increase with the increase in the number of rounds.

**Proof.** The lemma follows by noting that if a node \(v_i\) is missing a token \(t_j\) at the start of round \(r\), then \(v_i\) is missing token \(t_j\) at the start of every round \(r' < r\).

Lemmas 6 and 7 suggest that if we can identify a token distribution in which all half-empty configurations are small, we can guarantee small progress in each round. We now show that a well-mixed distribution satisfies the desired property, establishing part (b) of the theorem.

**Proof of Theorem 1(b).** We first note that if the number of tokens \(k\) is less than \(100 \log n\), then the \(\Omega(n + nk/\log n)\) lower bound is trivially true because even to disseminate one token on a line it takes \(\Omega(n)\) rounds. Thus, in the following proof, we focus on the case where \(k \geq 100 \log n\).

Let \(E_l\) denote the event that there exists a half-empty configuration of size \(l\) at the start of the first round. For \(E_l\) to hold, we need \(l\) nodes \(v_1, v_2, \ldots, v_l\) and \(l\) tokens \(t_1, t_2, \ldots, t_l\) such that for all \(i \neq j\) either \(v_i\) is missing \(t_j\) or \(v_j\) is missing \(t_i\). For a pair of nodes \(u\) and \(v\), by union bound, the probability that \(u\) is missing \(t_v\) or \(v\) is missing \(t_u\) is at most \(1/4 + 1/4 = 1/2\). Thus, the probability of \(E_l\) can be bounded as follows.

\[
\Pr[E_l] \leq \binom{n}{l} \cdot \frac{k!}{(k-l)!} \cdot \left(\frac{1}{2}\right)^\binom{l}{2} \leq n^l \cdot k^l \cdot \frac{1}{2^{l(l-1)/2}} \leq 2^{2l} \log n \cdot \frac{1}{2^{l(l-1)/2}}.
\]

In the above inequality, \(\binom{n}{l}\) is the number of ways of choosing the \(l\) nodes that form the half-empty configuration, \(k!/(k-l)!\) is the number of ways of assigning \(l\) distinct tokens, and \(1/2\binom{l}{2}\) is the upper bound on the probability for each pair \(i \neq j\) that either \(v_i\) is missing \(t_j\) or \(v_j\) is missing \(t_i\). For \(l \geq 5 \log n\),
Pr \[ E_i \] \leq 1/n^2. Thus, the largest half-empty configuration at the start of the first round, and hence at the start of any round (by Lemma 7), is of size at most \( 5 \log n \) with probability at least \( 1 - 1/n^2 \). By Lemma 6, we thus obtain that the number of useful token exchanges in each round is at most \( 10 \log n \), with probability at least \( 1 - 1/n^2 \).

Let \( M_i \) be the number of tokens missing at node \( i \) in the initial distribution. Then \( M_i \) is a binomial random variable with \( E[M_i] = k/4 \). By a Chernoff bound, the probability that node \( i \) misses at most \( k/8 \) tokens is

\[
Pr \left[ M_i \leq \frac{k}{8} \right] = Pr \left[ M_i \leq \left( 1 - \frac{1}{2} \right) \cdot E[M_i] \right] \leq e^{-\frac{E[M_i](\frac{1}{2})^2}{2}} = e^{-\frac{k}{32}}.
\]

Thus, the total number of tokens missing in the initial distribution is at least \( n \cdot \frac{k}{8} = \Omega(kn) \) with probability at least \( 1 - n/e^{\frac{k}{32}} \geq 1 - 1/n^2 \) \( (k \geq 100 \log n) \). Since the number of useful tokens exchanged in each round is at most \( 10 \log n \), the number of rounds needed to complete \( k \)-gossip is \( \Omega(kn/\log n) \) with high probability.

Part (b) of Theorem 1 does not apply to some natural initial distributions, such as one in which each token resides at exactly one node. When starting from a distribution in this class, though there are far fewer tokens distributed initially, the argument above does not rule out the possibility that an algorithm avoids the problematic configurations that arise in the proof. Part (a) of Theorem 1 extends the lower bound to this class of distributions. The main idea of the proof is showing that a reduction exists (via the probabilistic method) to an initial well-mixed distribution of Theorem 1.

**Lemma 8.** From any distribution in which each token starts at exactly one node and no node has more than one token, any online token-forwarding algorithm for \( k \)-gossip needs \( \Omega(kn/\log n) \) rounds against a strong adversary.

**Proof.** We consider an initial distribution \( C \) where each token is at exactly one node, and no node has more than one token. Let \( C^* \) be an initial token distribution in which each node has each token independently with probability \( 3/4 \). By Theorem 1, any online algorithm starting from distribution \( C^* \) needs \( \Omega(kn/\log n) \) rounds with high probability.

We construct a bipartite graph on two copies of \( V \), \( V_1 \) and \( V_2 \). A node \( v \in V_1 \) is connected to a node \( u \in V_2 \) if in \( C^* \) \( u \) has all the tokens that \( v \) has in \( C \). We first show, using Hall’s Theorem, that this bipartite graph has a perfect matching with very high probability. Consider a set of \( m \) nodes in \( V_2 \). We want to show their neighborhood in the bipartite graph is of size at least \( m \). We show this condition holds by the following 2 cases. If \( m < 3n/5 \), let \( X_i \) denote the neighborhood size of node \( i \). We know \( E[X_i] \geq 3n/4 \). Then by Chernoff bound

\[
Pr [X_i < m] \leq Pr [X_i < 3n/5] \leq e^{-\frac{(1/5)^2E[X_i]}{2}} = e^{-\frac{3nk}{200}}.
\]

By union bound with probability at least \( 1 - n \cdot e^{-3n/200} \), the neighborhood size of every node is at least \( m \). Therefore, the condition holds in the first case. If \( m \geq 3n/5 \), we argue the neighborhood size of any set of \( m \) nodes is \( V_1 \) with high probability. Consider a set of \( m \) nodes, the probability that a given token \( t \) is missing in all these \( m \) nodes is \( (1/4)^m \). Thus the probability that any token is missing in all these nodes is at most \( n(1/4)^m \leq n(1/4)^{3n/5} \). There are at most \( 2^m \) such sets. By union bound, with probability at least \( 1 - 2^n \cdot n(1/4)^{3n/5} = 1 - n/2^{n/5} \), the condition holds in the second case.

By applying the union bound, we obtain that with positive probability (in fact, high probability), \( C^* \) takes \( \Omega(nk/\log n) \) rounds and there is a perfect matching \( M \) in the above bipartite graph. By the probabilistic method, thus both \( C^* \) and \( M \) exist. Given such \( C^* \) and \( M \), we complete the proof as follows. For \( v \in V_2 \), let \( M(v) \) denote the node in \( V_1 \) that got matched to \( v \). If there is an algorithm \( A \) that runs in \( T \) rounds from starting state \( C \), then we can construct an algorithm \( A^* \) that runs in the same number of rounds from starting state \( C^* \) as follows. First every node \( v \) deletes all its tokens except for those which \( M(v) \) has in \( C \). Then
algorithm $A^*$ runs exactly as $A$. Thus, the lower bound of Theorem 1, which applies to $A^*$ and $C^*$, also applies to $A$ and $C$.

Proof of Theorem 1(a). We extend our proof in Lemma 8 to the initial distribution $C$ where each token starts at exactly one node, but nodes may have multiple tokens. We consider the following two cases.

The first case is when at least $n/2$ nodes start with some token. This implies that $k \geq n/2$. Let us focus on the $n/2$ nodes with tokens. Each of them has at least one unique token. By the same argument used in Lemma 8, disseminating these $n/2$ distinct tokens to $n$ nodes takes $\Omega(n^2/\log n)$ rounds. Thus, in this case the number of rounds needed is $\Omega(kn/\log n)$.

The second case is when less than $n/2$ nodes start with some token. In this case, the adversary can group these nodes together, and treat them as one super node. There is only one edge connecting this super node to the rest of the nodes. Thus, the number of useful token exchanges provided by this super node is at most one in each round. If there exists an algorithm that can disseminate $k$ tokens in $o(kn/\log n)$ rounds, then the contribution by the super node is $o(kn/\log n)$. And by the same argument used in Lemma 8 we know dissemination of $k$ tokens to $n/2$ nodes (those start with no tokens) takes $\Omega(kn/\log n)$ rounds. Thus, the theorem also holds in this case.

3 Upper bound in the weakly adaptive adversary model

In this section, we first analyze the SYM-DIFF protocol starting from a well-mixed distribution of tokens and prove Theorem 2 (presented in Section 3.1), and then show how to sample an element from the symmetric difference of two sets efficiently in the two-player communication complexity model (presented in Section 3.2). However, before doing that, we present the following lower bound that shows randomization is crucial for the SYM-DIFF protocol.

Theorem 9. Consider the protocol DET-SYM-DIFF for $k$-gossip in the weakly adaptive adversary model which is identical to the SYM-DIFF protocol except for, in each round, the token sent along each edge $(u,v)$ is chosen deterministically from the symmetric difference of the set of tokens held by node $u$ and the set of tokens held by node $v$. Starting from an initial token distribution where one node has all the $k$ tokens and others have none, a strongly adaptive adversary can force $\Omega(nk)$ rounds for the DET-SYM-DIFF protocol to disseminate the $k$ tokens to the $n$ nodes.

3.1 Analysis of SYM-DIFF starting from well-mixed distributions

For the proof of Theorem 2, we will assume that we start from the initial token distribution where each node has each token independently with probability $\frac{1}{2}$. It is easy to extend it to any positive constant probability. We need the following definition. We call a maximal set of nodes that holds the same set of tokens at the start of a round $r$ to be a group for round $r$.

Lemma 10. In a token distribution where each node has each token independently with probability $\frac{1}{2}$, the union of the set of tokens of any $\ell$ nodes misses at most $\frac{n+k}{2}$ tokens with high probability.

Since in any round, no token can be exchanged along an edge between two nodes of the same group, we will consider only the edges that connect two nodes from different groups. We call such edges inter-group edges for that round. In fact, we will prove the theorem in a stronger sense where we let the adversary orient the inter-group edges to determine the direction of token movement along all these edges, and the token sent along each of these edges is chosen uniformly at random from the symmetric difference conditioned on this orientation. (The adversary must respect the condition that there can be no token movement from a node $u$ to a node $v$ if the set of nodes held by node $u$ is a subset of that held by node $v$.) We define one unit of progress in a round as a node receiving a token in that round that it did not have at the start of the round.
Lemma 11. With high probability, the following holds for every node $v$ and every round $i$: If $v$ misses $m > \log n$ tokens at the start of round $i$ and it has $d > \log k$ incoming inter-group edges in that round, then node $v$ makes $\Omega(\min\{m,d\})$ units of progress in round $i$. Here, the probability is over the initial token distribution and the randomness used in the protocol.

Proof of Theorem 2. We color each of the rounds red, blue, green or black. If in a round, there is a node $v$ that misses less than $\log n$ tokens and makes at least one unit of progress in that round, we color the round red. If a round is not colored red, and there is a node that gets a constant fraction of its missing tokens in that round (the same fraction as in Lemma 11), we color it green. If a round is neither colored red nor colored green, we color the round blue.

It is immediate that there can be at most $n \log n$ red rounds since each of the $n$ nodes can be responsible for coloring at most $\log n$ rounds red. Similarly, there can be at most $O(n \log k)$ green rounds since each node can be responsible for coloring at most $O(\log k)$ rounds green. Fix a blue round and let there be $r$ groups in that round. Using Lemma 10, we infer that there are at most $(n + k)r$ tokens missing in total at the start of this round. We also note that there must be at least $r - 1$ inter-group edges in this round and combining this with Lemma 11 and the fact that this round was not colored red or green, we infer that we make $\Omega(\frac{r}{\log k})$ units of progress in this round.

We can label each blue round by the smallest number of groups in a blue round seen so far. The sequence of labels is non-increasing and let us say it starts from $s \leq n$. We divide the blue rounds in partitions where the $i$'th partition contain those with labels in $[s/2^{i-1}, s/2^i)$. There are at most $\log n$ partitions. From the above argument, we see that there can be at most $O((n + k) \log k)$ blue rounds in each partition, which implies a bound of $O((n + k) \log n \log k)$ for the total number of blue rounds. This completes the proof of the theorem. □

3.2 Uniform sampling from symmetric difference

We now restate and prove our result on a communication-efficient protocol to sample from the symmetric difference of two sets.

Theorem 3. Let Alice and Bob have two subsets $A \subseteq [k]$ and $B \subseteq [k]$ respectively. There is an explicit, private-coin protocol to sample a random element from the symmetric difference of the two sets, $A \oplus B := (A \setminus B) \cup (B \setminus A)$, such that the sampled distribution is statistically $\epsilon$-close to the uniform distribution on $A \oplus B$ and the protocol uses $O(\log^{3/2}(k/\epsilon))$ bits of communication.

We now explain how we obtain a communication-efficient protocol to sample from the symmetric difference $A \oplus B$ of two sets $A, B \subseteq [k]$, proving Theorem 3.

Out starting point is Nisan and Safra’s protocol [Nis93] to determine the least $i$ such that $i \in A \oplus B$. (In [Nis93] the protocol is phrased as deciding if $A > B$, when $A$ and $B$ are viewed as $k$-bit integers. It is easy to switch between the two.) For uniform sampling from $A \oplus B$, our idea is to first let the parties permute their sets according to a random permutation $\sigma$, then run Nisan and Safra’s protocol. This results in an explicit protocol for uniform generation from $A \oplus B$ with communication $O(\log k/\epsilon)$ that uses public coins. A standard transformation to private coins via [New91] results in a protocol that is not explicit.

To obtain an explicit, private-coin protocol we derandomize the space of random permutations $\sigma$. The key idea is that it is sufficient to have a distribution on permutations $\sigma$ such that, for any set $D = A \oplus B$, any element in $D$ has roughly the same probability of being the first element in $D$ to appear in the sequence $\sigma(1), \sigma(2), \sigma(3), \ldots$. We then construct such a space of permutations with seed length $O(\log^{3/2}(k/\epsilon))$ using Lu’s pseudorandom generator for combinatorial rectangles [Lu02] (cf. [Nis92, NZ96, INW94, EGL+98, ASWZ96, Lu02, Vio11b]). Plugging a better pseudorandom generator for combinatorial rectangles in our argument would result in a protocol for uniform sampling from $A \oplus B$ with communication $\tilde{O}(\log k/\epsilon)$ and error $\epsilon$. 

As a first step, we have the following simple derandomization of Nisan and Safra’s protocol [Nis93], essentially from [Vio11a].

**Lemma 12.** There is an explicit, private-coin protocol to determine the least \(i \in A \oplus B\), where \(A, B \subseteq [k]\), with error \(\alpha\) and communication \(O(\log(k/\alpha) \log \log k) = \tilde{O}(\log k/\alpha)\).

Specifically, for given \(k\) and \(\epsilon\) as in Theorem 3 we set \(d = k \log \left(\frac{3k}{\epsilon}\right)\) and \(\alpha := \epsilon/3kd\). Alice then picks a random seed of length \(s(k, d, \alpha)\) for a generator that fools every combinatorial rectangle with universe size \(k\) and \(d\) dimensions with error \(\alpha\). That is, if \(X\) is the output of the generator on a random seed, we have, for every set \(R := R_1 \times R_2 \times \cdots \times R_d \subseteq [k]^d\),

\[
|\Pr[X \in R] - |R|/k^d| \leq \alpha.
\]

Alice sends the seed to Bob.

Both Alice and Bob expand the seed into a sample \(X\) of the generator, and use \(X\) to generate a permutation \(\sigma\) as follows. Let the number of distinct elements of \([k]\) that appear in \(X\) be \(t\). The permutation \(\sigma\) is constructed by defining \(\sigma(i)\) to be the \(i\)'th distinct element of \([k]\) that appears in \(X\) as we scan it from the beginning, for \(i \leq t\). For every \(i > t\), \(\sigma(i)\) is defined to be a distinct element not appearing in \(X\) in an arbitrary but deterministic way that is fixed before the start of the protocol and both Alice and Bob are aware of it. (For concreteness, it can simply be to assign the elements not appearing in \(X\) by order).

To show the correctness of our protocol we need the following lemma.

**Lemma 13.** Let \(X \in [k]^d\) be the output of a combinatorial rectangle generator with error \(\alpha = \epsilon/3kd\), over a uniform seed. Let \(D\) be any set, and let \(j\) be any element in \(D\). The probability that \(j\) appears in a coordinate of \(X\) before any other element of \(D\) is \(\geq \frac{1}{|D|} - \frac{2\epsilon}{3k}\).

Now we can complete the proof of Theorem 3.

**Proof of Theorem 3.** For given \(k, \epsilon\), we set \(d = k \log \left(\frac{3k}{\epsilon}\right)\) and \(\alpha := \epsilon/3kd\). Alice then picks a random seed of length \(s(k, d, \alpha)\).

If \(\sigma\) is chosen such that every element \(j \in D\) has probability \(\frac{1}{|D|}\) of preceding all other elements of \(D\), then \(\sigma(i^*)\) is a uniform random element of \(D\), where \(i^*\) is the first position where the permuted \(A\) and \(B\) differ. Using Lemma 13, we immediately see that if \(\sigma\) is chosen as in the first step of the protocol, then the distribution of \(\sigma(i^*)\) is at most \((\frac{3k}{\epsilon})|D| \leq \frac{2\epsilon}{3}\)-far from the uniform distribution on \(D\).

For the second part of the protocol we use Lemma 12 with \(\alpha := \epsilon/3\).

Overall, the sampled distribution has distance \(\leq 2\epsilon/3 + \epsilon/3 = \epsilon\) from the uniform distribution on \(D\).

Using the generator in [Lu02] we have \(s(k, d, \alpha) = O(\log n + \log d + \log^2 1/\alpha) = O(\log^{3/2} n/\epsilon)\). So overall the communication is \(O(\log^{3/2} n/\epsilon)\). \(\square\)

### 4 Offline token-forwarding algorithms

We present two offline algorithms for \(k\)-gossip. The first computes an \(O((n + k) \log^2 n)\)-round schedule assuming that each node can send at most one token to each neighbor in each round (Section 4.1); the second computes an \(O(\min\{n \sqrt{k \log n}, nk\})\)-round broadcast schedule assuming that each node can broadcast at most one token to its neighbors in each round (Section 4.2).
4.1 $O((n + k) \log^2 n)$-round offline schedule

In this section, we present an algorithm for computing an $O((n + k) \log^2 n)$ round offline schedule. Our bound is tight to within an $O(\log^2 n)$ factor since the dissemination of any $k$ tokens to even a single node of the network requires $\Omega(n + k)$ rounds in the worst case. We begin by defining the notion of an evolution graph that facilitates the design of the offline algorithms.

**Evolution graph:** Let $V$ be the set of nodes. Consider a dynamic network of $l$ rounds numbered 1 through $l$ and let $G_i$ be the communication graph for round $i$. The evolution graph $\hat{G}[l]$ for this network is a directed capacitated graph $G$ with $l + 1$ levels constructed as follows. We create $l + 1$ copies of $V$ and call them $V_0, V_1, V_2, \ldots, V_l$. $V_i$ is the set of nodes at level $i$ and for each node $v$ in $V$, we call its copy in $V_i$ as $v_i$. For $i = 1, \ldots, l$, level $i - 1$ corresponds to the beginning of round $i$ and level $i$ corresponds to the end of round $i$. Level 0 corresponds to the network at the start. There are two kinds of edges in the graph. First, for every node $v$ in $V$ and every round $i$, we place an edge with infinite capacity from $v_{i-1}$ to $v_i$. We call these edges buffer edges as they ensure tokens can be stored at a node from the end of one round to the end of the next. Second, for every round $i$ and every edge $(u, v) \in G_i$, we place two directed edges with unit capacity each, one from $u_{2i-1}$ to $v_{2i}$ and another from $v_{2i-1}$ to $u_{2i}$. We call these edges transmit edges as they correspond to every node transmitting a message to a neighbor in round $i$; the unit capacity ensures that in a given round a node can transmit at most one token to each neighbor. Figure 2 illustrates our construction.

**Figure 2:** An example of how to construct the evolution graph from a sequence of communication graphs.

---

**Algorithm 1** Computing an $O((n + k) \log^2 n)$-round schedule for $k$-gossip

**Require:** A sequence of communication graphs $G_i; i = 1, 2, \ldots$

**Ensure:** Schedule to disseminate $k$ tokens to all nodes

1: **Gather:** Send the $k$ tokens to a node $v_0$, chosen uniformly at random, in $n + k$ rounds.
2: for $i$ from $0$ to $\log n$ (Phase $i$) do
3: Choose a set $S_i$ of $2^i$ nodes uniformly at random from the collection of all $2^i$-size node sets.
4: **Flow:** Send the $k$ tokens to every node in $S_i$ using a maximum flow in an $O((n + k) \log n)$-round evolution graph from the set $\{v_0\} \cup \bigcup_{j < i} S_i$ of sources to the set $S_i$ of sinks.
4.2 An $O(\min\{n^{\sqrt{k\log n}}, nk\})$-round broadcast schedule

We extend the notion of the evolution graph to the broadcast model and show that finding a broadcast schedule for $k$-gossip can be reduced to packing Steiner trees in the evolution graph. We defer this section to the full paper.

5 Concluding remarks and open questions

We studied the fundamental $k$-gossip problem in dynamic networks and showed a lower bound of $\Omega(n + nk/\log n)$ rounds for any token forwarding algorithm against a strongly adaptive adversary, significantly improving over the previous best bound of $\Omega(n \log k)$ [KLO10] for sufficiently large $k$. Our lower bound matches the known upper bound of $O(nk)$ up to a logarithmic factor, and establishes a near-linear factor separation between token-forwarding and network-coding based algorithms. While our bound rules out significantly faster algorithms in the strongly adaptive adversary model, we complement our lower bound by presenting the SYM-DIFF protocol for a weakly adaptive adversary. We show that SYM-DIFF takes is near-optimal when the starting distribution is well-mixed. Intuitively, a well-mixed distribution captures the “hard” regime for information spreading in the adversarial setting, when most nodes have most of the tokens. Perhaps, the most interesting problem left open by our work is the analysis of SYM-DIFF in the weakly adaptive adversary model for an arbitrary starting distribution.

We also presented offline algorithms for $k$-gossip. An important intermediate model between the offline setting and the adaptive adversary models is the oblivious adversary model in which the adversary lays the dynamic network in advance (as in the offline setting), but the changing topology is revealed to the algorithm one round at a time. Finally, this paper has focused on models in which at most one token is sent per edge per round and the network can change every round. Subsequent to the announcement of our lower bound [DPRS11], the argument has been extended to the model where multiple tokens can be broadcast and the dynamic network is required to contain a stable subgraph for multiple rounds [HK].

References


