Fast gossiping with short unreliable messages

Bogdan S. Chlebus\textsuperscript{a,1}, Krzysztof Diks\textsuperscript{a,1}, Andrzej Pelc\textsuperscript{b,2,*}

\textsuperscript{*}Instytut Informatyki, Uniwersytet Warszawski, ul. Banacha 2, 02-097, Warszawa, Poland
\textsuperscript{1}Département d’Informatique, Université du Québec à Hull, C.P. 1250, succ. “B”, Hull, Qué. J8X 3X7, Canada

Received 25 June 1991; revised 28 May 1992

Abstract

Each of \( n \) nodes of a communication network has a piece of information (gossip) which should be made known to all other nodes. Gossiping is done by sending letters. In a unit of time each node can either send one letter to a neighbor or receive one such letter, containing one gossip currently known to the sender. Letters reach their destinations with constant probability \( 0 < q < 1 \), independently of one another. For a large class of networks, including rings, grids, hypercubes and complete graphs, we construct gossip schemes working in linear time and successfully performing gossiping with probability converging to 1, as the number of nodes grows.

1. Introduction

Each of \( n \) nodes of a communication network (modeled by a simple connected undirected graph) has a piece of information (gossip) unknown to others, which should be made known to all other nodes. This problem, known as gossiping, has received a lot of attention in the literature. An extensive bibliography can be found in [12].

Various ways of specifying the communication process yield different models of gossiping. We note three aspects which give rise to many such models. First, communication may be either half-duplex, i.e. information flows between neighbors in only one direction in a unit of time, or full-duplex, when neighbors can simultaneously exchange information. The half-duplex model corresponds to sending letters or...
telegrams while the full-duplex model corresponds to making phone calls. Second, a node may either communicate with all its neighbors in a unit of time (simultaneous communication corresponding to E-mail or conference calls) or communication may be restricted to only one neighbor at a unit of time (pairwise communication corresponding to ordinary phone calls). Each of the four models corresponding to these distinctions was studied, e.g. in [13, 14]. Finally, we may either assume that all information available to a node can be transmitted in a unit of time (an assumption commonly made in the literature) or that the length of transmission depends on the number of messages sent (cf. [6]).

One of the important parameters of a gossip scheme is the total time it uses. Gossiping in minimal time has been studied, e.g. in [8, 9, 13–15].

Recently a lot of attention has been devoted to gossiping (and closely related broadcasting) in the presence of faulty links [2, 4, 7–9, 11, 16, 17]. Two alternative assumptions about faults are usually made: either an upper bound $k$ on the total number of faults is supposed [9, 11] or it is assumed that links fail independently with fixed probability $p$ [2, 4, 7, 8]. If an upper bound is imposed and the worst case is considered, the maximum number of faults that can be tolerated cannot exceed the connectivity of the network. Thus, for large networks, the stochastic approach seems to be more realistic. A point which should be specified under the probabilistic fault model is this: link failures may either be permanent (the fault status of a link does not change during the execution of the scheme, cf. [4, 7]) or individual calls (letters) along each link may be subject to independent failures (cf. [8]). Individual transmission faults may occur in real-life situations when links are not defective but the fault is due to momentary random "noise" in the communication channel, e.g. to electromagnetic interference. It was proved in [7] that reliable gossip schemes are impossible for bounded maximum degree networks (such as rings or grids) under the permanent fault scenario, whence the interest in the other model.

In this paper we construct a fault-tolerant time-efficient gossip scheme under assumptions which make fast communication difficult. We work in the half-duplex pairwise model, i.e. in each unit of time every node can communicate with at most one neighbor and during such communication information can be either sent or received but not both. We assume that each transmission takes a unit of time but during such a transmission from node $u$ to node $v$, $u$ can send only one of the gossips it currently knows. Our schemes are fully synchronous, i.e. time units are measured by a global clock. We also assume that individual transmissions are subject to independent failures with constant probability $1 - q$, $0 < q < 1$, and no information arrives at the destination node during a faulty transmission. Thus, our communication model corresponds to sending short letters (each containing one gossip) which reach their destination with constant probability $q$, independently of one another. For the sake of simplicity, we assume that the sender knows if the letter arrived at its destination, i.e. we adopt the registered mail model. This assumption, however, can easily be removed by requiring that upon reception of a gossip the receiving node confirms it by returning the gossip back to the sender. Thus, the probability that both the letter and
its confirmation reach their destinations is now \( q_1 = q^2 \) and the number of time units doubles. Since our results concern only the order of magnitude of execution time and work for any failure probability strictly less than one, a gossip scheme developed for the registered mail model can be applied with parameter \( q_1 \) and the result holds true for parameter \( q \) without this extra assumption.

We say that a gossip scheme is \textit{successful} for a graph \( G \) if upon its completion every node gets all the gossips. A gossip scheme for \( G \) is \( \varepsilon \)-\textit{safe} for \( \varepsilon > 0 \) if it is successful with probability at least \( 1 - \varepsilon \). A gossip scheme working for a family \( \{G_n; n \geq 1\} \) of \( n \)-node graphs is \textit{almost safe} if it is \( \varepsilon_n \)-safe for the graph \( G_n \), where \( (\varepsilon_n; n \geq 1) \) is some sequence converging to 0. The \textit{execution time} of a gossip scheme is the number of time units it takes. A gossip scheme working for a family \( \{G_n; n \geq 1\} \) of \( n \)-node graphs is \textit{order-optimal} if its execution time for \( G_n \) is \( f(n) \), the shortest possible gossiping time for \( G_n \) is \( g(n) \), and \( f(n) \in O(g(n)) \). All operations other than sending letters are assumed to take negligible time.

The aim of this paper is the construction of fast almost safe gossip schemes. Let \( \{G_n; n \geq 1\} \) be a family of \( n \)-node graphs with spanning trees of bounded maximum degree (e.g., rings, grids, hypercubes, complete graphs). We construct an almost safe gossip scheme working for \( \{G_n; n \geq 1\} \) in time \( O(n) \) which is order-optimal.

In Section 2 our gossip scheme is described, while Section 3 is devoted to the analysis of its reliability. Section 4 contains conclusions.

2. Description of the gossip scheme

Our scheme is easiest to explain for Hamiltonian graphs. If \( H \) is a Hamiltonian cycle of the graph \( G \), the aim of the scheme is to make every gossip visit consecutive nodes of \( H \). It uses a procedure \textsc{Transmit}(v, u) (to be described later) whose aim is to transmit a single gossip from \( v \) to its successor \( u \) in the cycle. Since transmissions can be unsuccessful, the scheme works in \( cn \) phases, where \( n \) is the number of nodes and \( c \) is a constant parameter chosen in such a way as to guarantee the desired reliability of gossiping.

In the general case, instead of following the cycle \( H \), gossips make a tour of a spanning tree \( T \) of \( G \), visiting all its nodes in preorder. This implies visiting some nodes many times which, although not optimal, does not increase the order of magnitude of execution time and makes the analysis much simpler. The detailed description follows.

Let \( G \) be an \( n \)-node connected graph, \( n > 2 \), and \( T \) a spanning tree of \( G \). Denote by \( d(T) \) the maximum degree of \( T \). Fix any node \( r \) of degree 1 in \( T \) and consider it as the root of \( T \). For any node \( v \), the terms parent(v) and child(v) are meant with respect to this rooted tree. For any node \( v \), let \( N(v) \) denote the number of children of \( v \). Clearly, \( N(v) \leq d(T) - 1 \). If \( N(v) = 0 \) then \( v \) is called a leaf. Enumerate all children of every nonleaf \( v \) in any order and call them child(v, 1), \ldots, child(v, nc(v)). We assume that
every node \( u \neq r \) knows its position among the siblings, i.e. it knows the integer \( j \) such that \( \text{child}(\text{parent}(u), j) = u \).

Fix a positive integer \( c \) to be determined later. We describe the gossip scheme \( \text{GS}(T, c) \) whose aim is to make every gossip visit all nodes of the tree \( T \) in preorder. More precisely, every nonleaf \( v \) sends information to the subtree with root \( \text{child}(v, 1) \). After visiting all nodes of this subtree, information goes back to \( v \), then visits all nodes of the subtree with root \( \text{child}(v, 2) \), goes back to \( v \) and so on. After visiting the subtree with root \( \text{child}(v, N(v)) \) and coming back to \( v \), information is then sent to \( \text{parent}(v) \) (if \( u \neq r \)). Let \( C = (v_0, v_1, v_2, \ldots, v_{2n-3}, v_{2n-2}) \), with \( v_0 = v_{2n-2} = r \), be the cycle corresponding to this preorder traversal of \( T \), every node being listed each time it is visited. If all gossips have visited every node at least once, the scheme \( \text{GS}(T, c) \) is successful.

**Algorithm GS\((T, c)\)**

begin

\textbf{for} phase := 1 to \( c(2n - 2) \) \textbf{do} \\
\textbf{for} all \( v \) on even levels in \( T \) \textbf{in parallel do} \\
\textbf{for} \( i := 1 \) to \( d(T) - 1 \) \textbf{do} \\
\textbf{if} \( i \leq N(v) \) \textbf{then} \\
\text{TRANSMIT}(v, \text{child}(v, i)) \\
\textbf{fi} \\
\textbf{od}; \\
\textbf{for} \( i := 1 \) to \( d(T) - 1 \) \textbf{do} \\
\textbf{if} \( v \neq r \) \textbf{and} \( v = \text{child}(\text{parent}(v), i) \) \textbf{then} \\
\text{TRANSMIT}(v, \text{parent}(v)) \\
\textbf{fi} \\
\textbf{od}; \\
\textbf{od}; \\
\textbf{for} all \( v \) on odd levels in \( T \) \textbf{in parallel do} \\
\textbf{for} \( i := 1 \) to \( d(T) - 1 \) \textbf{do} \\
\textbf{if} \( i \leq N(v) \) \textbf{then} \\
\text{TRANSMIT}(v, \text{child}(v, i)) \\
\textbf{fi} \\
\textbf{od}; \\
\textbf{for} \( i := 1 \) to \( d(T) - 1 \) \textbf{do} \\
\textbf{if} \( v = \text{child}(\text{parent}(v), i) \) \textbf{then} \\
\text{TRANSMIT}(v, \text{parent}(v)) \\
\textbf{fi} \\
\textbf{od}; \\
\textbf{od}; \\
\textbf{od}; \\
end \{of the scheme\}.  

---
It remains to describe the procedure TRANSMIT(v, u), where u is a successor of v in the cycle C. For every node v we define one or more FIFO (first in first out) queues. For the root r, Q(r) is the queue which stores gossips that come from the unique child of r and will be sent back to this child. For any leaf u, Q(u) is the queue which stores gossips that come from parent(u) and will be sent back to parent(u). Finally, for a node u which is neither a leaf nor the root, we define N(u) + 1 queues Q(u, 1), Q(u, 2), ..., Q(u, N(u) + 1). The queue Q(u, 1) stores gossips that come from parent(u) and will be sent to child(u, 1). Each queue Q(u, i), 2 ≤ i ≤ N(u), stores gossips that come from child(u, i − 1) and will be sent to child(u, i). Finally, Q(u, N(u) + 1) stores gossips that come from child(u, N(u)), and will be sent to parent(u). All queues Q(u) and Q(u, i) are initialized by inserting the gossip originally held by u.

For every queue Q we define head(Q) to be the first element of Q (for Q ≠ 0), and describe two operations for a queue Q at vertex v.

insert(Q, e): put the gossip e at the end of Q;
delete(Q): if Q ≠ 0, delete head(Q) from Q.

Moreover, for any neighbors v, u, send(v, u, e) denotes the action of sending a letter containing gossip e from v to u. Since we work in the registered mail model, both v and u know if the letter was lost or not.

For any i < 2n − 2 consider nodes v_i, v_{i+1} in the cycle C (all indices are taken mod 2n − 2). The procedure TRANSMIT(v_i, v_{i+1}) identifies the queue P where node v_i stores information coming from v_{i-1} and the queue Q where v_{i+1} stores information coming from v_i. Then, if P ≠ 0, the action send(v_i, v_{i+1}, head(P)) is performed. If the letter arrives, v_i performs delete(P) and v_{i+1} performs insert(Q, head(P)). Thus, the procedure TRANSMIT(v_i, v_{i+1}) is performed in a unit of time and consequently the scheme GS(T, c) is executed in time c(2n − 2)d(T). (As usual, the word "executed" means only that appropriate messages have been sent and does not imply that they were actually received.)

3. Reliability of gossiping

Our main result shows that the scheme GS(T, c) can achieve any desired reliability strictly less than one for an appropriate constant c.

**Theorem 3.1.** Let G be an n-node graph and T its spanning tree with maximum degree d(T). For every ε > 0 there exists a constant c, independent of n and d(T), such that GS(T, c) is ε-safe.

In order to prove this theorem we will need some results from queuing theory (cf. [10]). A similar approach has been used in our paper [5] in a different context, that of sorting on a faulty mesh network.
We can visualize the travel of individual gossips in the network as a cycle of servers with customers waiting in queues. In the beginning there is exactly one customer at each server. If a customer has been served by a server then it is moved to the queue of the next server. In each unit of time a server takes the first customer from its queue, provided that it is nonempty, and attempts to serve it. Such an attempt is successful with probability \(q\).

We now modify the above process to facilitate the probabilistic analysis of its behavior. First, instead of a cycle of servers, consider a line of \(n\) servers with \(n\) customers waiting for service from the first server. It will be shown later how to reduce gossip circulation to this scenario.

Next, suppose that customers are not given in advance but are generated by the first server and then seek service from consecutive servers as before. They are generated with a geometric distribution on waiting time; more precisely, in each step a new customer is created with fixed probability \(p\) (we refer to \(p\) as the input probability). The probability that \(n\) customers thus generated are served by all servers in time at most \(T\) does not exceed the probability that \(n\) customers waiting for the first server in the beginning are served by all servers in time at most \(T\).

As a first step in the analysis we consider a one-server system. Suppose that at each step a new customer is created with input probability \(p\), and the first customer in the queue is served with probability \(q\). This is an example of a Markov chain. It is said to be in state \(i\) if there are exactly \(i\) customers in the queue. If the queue is nonempty, then the probability of its size being incremented in one step is \(a = (1 - q)p\), and the probability of its size being decremented is \(b = (1 - p)q\). The transition matrix \(P = (P_{ij})\) is obtained as follows:

\[
P_{ij} = \Pr\{\text{state } j \text{ is entered at time } t + 1 | \text{the state at time } t \}.
\]

Note that the above probability is independent of time \(t\).

The entries of \(P\) are given below:

1. \(P_{ij} = 0\) if \(|i - j| > 1\);
2. \(P_{00} = 1 - p, P_{10} = b\);
3. \(P_{01} = p, P_{11} = 1 - a - b, P_{21} = b\);
4. \(P_{k-1,k} = a, P_{kk} = 1 - a - b, P_{k+1,k} = b\), provided \(k > 1\).

If \(P^{(0)}\) is some initial distribution on the set of all the states then after \(k\) steps the chain is in a state determined by the probability distribution

\[
P^{(k)} = P^{(0)} \cdot P^k.
\]

A Markov chain is said to be ergodic if its probability distribution after \(k\) steps converges to the same probability distribution for every initial distribution, as \(k \to \infty\). The following lemma follows from Foster's theorem stating that a (aperiodic and irreducible) Markov chain is ergodic if there is a nonnull solution \(x = (x_{ij})\) of the equation \(x = xP\), with \(\sum x_{ii} < \infty\) (cf. [3, 5]).
Lemma 3.2. If $p < q$ then the single-server system is ergodic.

Ergodicity of the system implies the existence of a unique stationary distribution with the property that if it is the initial distribution $P^{(0)}$ then all the subsequent distributions

$$P^{(k)} = P^{(0)} \cdot p^k$$

are the same as $P^{(0)}$.

The following lemma was proved in [5].

Lemma 3.3. If the initial distribution is stationary, then the output of the single-server system has a geometric distribution with parameter $p$.

Now suppose that in the beginning there are already some customers waiting in queue at each server. They are introduced for technical reasons and are not the "true" customers who still need to be generated; we call them "dummy customers". Their number in each queue at the beginning of the process is given by the stationary distribution guaranteed by ergodicity (as long as $p < q$). It follows from Lemma 3.3 that a system of queues with such initial distribution will have input and output geometric distribution with parameter $p$, at each server. The probability of clearing the queues of dummy customers and serving $n$ true ones, in time at most $T$, does not exceed the probability of serving $n$ customers, initially waiting in queue at the first server, in time at most $T$.

The next lemma gives a bound on the probability that there is a specific number of dummy customers in the beginning of the process. This bound is used to prove Lemma 3.5. The assumption $p = q^2$ (implying $p < q$) is chosen for technical reasons.

Lemma 3.4. Let $s_k$ denote the probability that there are $k$ dummy customers in all the $n$ queues, if the number of customers in each queue is given by the stationary distribution. Take $p = q^2$. Then

$$s_k \leq \left(\frac{n + k - 1}{k}\right) \left(\frac{q}{1 + q}\right)^k.$$

Denote by $S$ the random variable equal to the time needed to serve $n$ true customers by the series of $n$ queues. The next lemma shows that for sufficiently large values the distribution of $S$ can be bounded by a geometric one. The proof is given in [5]; the lemma also follows from results of Berman and Simon [1].

Lemma 3.5. There are two constants $\delta$ and $c_0$, where $0 < \delta < 1$ and $c_0 > 0$, such that

$$\Pr(S = t) \leq (1 - \delta)^t \delta^t \quad \text{for} \quad t \geq c_0 n.$$
Proof of Theorem 3.1. If follows from Lemma 3.5 that for some constants $0 < \delta < 1$ and $c_0 > 0$ the probability that $m$ customers are served by $m$ servers in time larger than $dm$, for $d \geq c_0$, is at most

$$\Pr(S > dm) \leq (1 - \delta) \sum_{t = dm + 1}^{\infty} \delta^t = \delta^{dm + 1}.$$ 

Thus, for $c \geq 2 \max(c_0, \log \varepsilon/\log \delta)$, the probability that service time exceeds $cm/2$ is less than $\varepsilon$, for any $\varepsilon$ and $m$. Take $c$ as above to be the parameter of our scheme.

Consider the cycle $C$ of length $2n - 2$ defined in Section 2. In each phase of our gossip scheme $GS(T, c)$, an attempt of sending a gossip to the next node of the cycle is made and the probability of success is $q$. If all gossips visit all nodes of the cycle, the scheme is successful. Consider the line $L = (u_0, u_1, \ldots, u_{2n-3}, v_0, v_1, \ldots, v_{2n-3})$ of nodes, which makes twice the tour of the cycle $C$. Originally each gossip is situated at its node in the first half of the line. If all these gossips traverse all positions in the line $L$ to the right of their initial position, each gossip will make a full round of the cycle $C$. Putting all gossips at the beginning of the line, as well as increasing their number, can only make the task of traversing $L$ more difficult. Transmitting a gossip to the next node with probability $q$ can be interpreted as serving a customer by a server, with the same probability. Thus, the probability that all gossips make a full round of the cycle after time $T$, starting at their initial positions, is at least as large as the probability that $4n - 4$ customers are served by $4n - 4$ servers in time $T$. It follows that the probability that the scheme $GS(T, c)$ is not successful does not exceed the probability that $4n - 4$ customers are served by a line of $4n - 4$ servers in time exceeding $c(2n - 2)$. As noticed above, for $c \geq 2 \max(c_0, \log \varepsilon/\log \delta)$ this probability is less than $\varepsilon$ and consequently the scheme $GS(T, c)$ is $\varepsilon$-safe. \qed

Consider any family $\{G_n: n \geq 1\}$ of $n$-node graphs. In view of Lemma 3.5, for some constants $0 < \delta < 1$ and $c_0 > 0$ we have

$$\Pr(S > c_0 n) \leq \delta^{c_0 n + 1},$$

which is less than $1/n$ for sufficiently large $n$. It follows that the scheme $GS(T_n, 2c_0)$, where $T_n$ is a spanning tree of $G_n$, is $1/n$-safe for the graph $G_n$ and hence it is almost safe for the family $\{G_n: n \geq 1\}$.

Now suppose that the graphs $G_n$, $n \geq 1$, have spanning trees $T_n^*$ of bounded maximum degree. Many important classes of graphs, such as rings, grids, hexagonal meshes, hypercubes and complete graphs, satisfy this requirement and the trees $T_n^*$ are easy to construct in these cases. For such families of graphs the execution time of scheme $GS(T_n^*, c)$ is $c(2n - 2) d(T_n^*) \in O(n)$. This proves the following.

Corollary 3.6. Let $\{G_n: n \geq 1\}$ be a family of $n$-node graphs with spanning trees of bounded maximum degree. Then there exists an almost safe gossip scheme for this family, with execution time $O(n)$. 

If graphs have spanning trees of bounded maximum degree then the execution time of our scheme is order-optimal. Indeed, even when all letters reach their destinations, getting all gossips to a given node takes time at least \( n - 1 \). The scheme is also order-optimal for some other families of graphs, e.g. for trees with arbitrary maximum degree. Let \( \{ T_n; n \geq 1 \} \) be any family of \( n \)-node trees with maximum degree \( d(n) \). Our almost safe gossip scheme works for this family in time \( O(nd(n)) \). To show that this is order-optimal it is enough to prove that every gossip scheme in our model, even with all letters reaching their destinations \( (q = 1) \), must take \( \Omega(nd(n)) \) time units for trees. Indeed, let \( v \) be a node of degree \( d(n) \) in \( T_n \). Divide all neighbors of \( v \) into sets \( A \) and \( B \), each of size at least \( \lceil d(n)/2 \rceil \). This division partitions \( T_n \) into two subtrees: one on the same side of \( v \) as \( A \) and the other on the same side of \( v \) as \( B \). Without loss of generality, we may assume that the first subtree has at least \( n/3 \) nodes. All gossips originating in these nodes have to be transmitted via \( v \) to all nodes in \( B \). Since there are at least \( \lceil d(n)/2 \rceil \) such nodes, this requires at least \( (n/3) \lceil d(n)/2 \rceil \in \Omega(nd(n)) \) time units.

We do not know if our gossip scheme is order-optimal for every family \( \{ G_n; n \geq 1 \} \) of \( n \)-node graphs. A positive answer to this question would follow from the following conjecture which we are unable to prove.

Let \( G \) be an \( n \)-node graph, \( T \) its spanning tree and \( d(T) \) the maximum degree of \( T \). Define \( d(G) = \min \{ d(T); T \text{ is a spanning tree of } G \} \). Consider the half-duplex pairwise communication model with letters containing only one gossip, without failures.

**Conjecture 3.7.** There exists a constant \( c \) (independent of \( n \) and \( d(G) \)) such that every gossip scheme in \( G \) requires at least \( cnd(G) \) time units.

**Conclusions**

We presented a gossip scheme working under the assumption that every node of the communication network can either send or receive one letter containing one gossip in a unit of time and that letters reach their destinations with constant probability \( 0 < q < 1 \), independently of one another. Our scheme works with probability converging to 1 as the number of nodes grows and its execution time is of least possible order of magnitude for many important families of graphs, such as rings, grids, trees, hypercubes and complete graphs. It is an open problem if order-optimality of our scheme holds for every class \( \{ G_n; n \geq 1 \} \) of \( n \)-node graphs. As far as the total number of letters is concerned, our scheme uses \( O(n^2) \) of them, and \( \Omega(n^2) \) letters are needed, even without failures, for all graphs.

**References**